Policy Map - Class Map

bintec-Dm 795-I

Legal Notice

Warranty

This publication is subject to change.

bintec offers no warranty whatsoever for information contained in this manual.

bintec is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

I Related Documents

bintec-Dm 715-I BRS

bintec-Dm 752-I Access Control

bintec-Dm 754-I NSLA

bintec-Dm 772-I Configuration Interfaces

Chapter 1 Introduction

1.1 Service policies

As part of QoS, traffic policies provide us with a global classification level that allows us to filter received traffic (or the traffic that passes through the device).

Traffic policies can be created and connected to one or more interfaces. Said policies apply to specific actions in the corresponding traffic.

Traffic policies allow you to:

- Classify the traffic
- Label the traffic
- Limit the bandwidth (*traffic policing*)

Policy-maps define traffic policies. A policy-map is made up of one or several class-maps. The latter are particularly interesting since they are traffic classes that define the way traffic is classified. A class-map is defined through a series of entries that we want to apply to a group of input traffic.

Once done, the service policy is used to apply the traffic policy to the corresponding interface.

The service policy can be applied to the following types of data links:

- Frame Relay
- X.25 line
- PPP line
- HDLC line
- ATM subinterfaces
- Ethernet Interfaces
- Ethernet subinterfaces
- Wireless LAN Interfaces
- TNIP Interfaces
- BVI Interfaces

For further information on how service policies are applied to interfaces, see manual *bintec-Dm 772-I* "*Configuration Interfaces*".

The following sections detail the functions of traffic policies.

1.1.1 Traffic classification

Traffic policies classify incoming traffic and apply several policies of quality of service, limiting or labeling the traffic based on the configured criteria. This classification can be carried out through the following:

- Access lists
- Classification through specified criteria

Traffic that is not classified into either of these two categories is sent to the default class (should one exist).

1.1.1.1 Access lists for traffic classification

Traffic policies allow you to classify the traffic based on IPv4 access lists (both standard and extended). For further information on IPv4 access lists, please see manual *bintec-Dm 752-I* "*Access Control*".

```
Class-map Class_100>entry 1 match access-list 100
```

1.1.1.2 Classifying traffic through specific criteria

Traffic policies allow you to classify the traffic based on specific classification criteria, e.g., the value of the packet CoS or the packet source or destination MAC.

```
Class-map Class_100>entry 1 match cos 3
Class-map Class_100>entry 2 match source mac 02-01-02-03-04-05
```

1.1.2 Traffic labeling

Quite often, in order to distinguish them and deal with them in an appropriate fashion, it is useful to label the packets. Traffic policies allow you to:

- Label all the traffic belonging to the same category via the traffic class menu.

```
Pmap Policy_100-Cmap Class_100>set label 20
```

1.1.3 Limiting the bandwidth – Traffic policing

Traffic policies allow us to carry out *traffic policing* over all the interfaces that support this. *Traffic policing* allows you to limit an interface's maximum throughput. In this mode, *traffic policing* drops the traffic that exceeds the permitted limitation by default.

To limit the maximum throughput, *traffic policing* uses the **police** command (which is configured in the traffic policies classes menu).

```
Pmap Policy_100-Cmap Class_100>police <cir> [<bc> [<be>]]
Pmap Policy_100-Cmap Class_100>police link-layer [offset <value>]
Pmap Policy_100-Cmap Class_100>police network-layer
```

Given that the line rate is set and all packets must be integrally and continuously received, you can only act on the average throughput during a certain time interval. To do this, use the *police* command. It accepts up to three parameters: CIR, Bc and Be. These parameters define how the throughput is limited. The meaning of these parameters is as follows:

- CIR: Committed Information Rate. This is the throughput allowed, i.e., the sustained transfer rate that can be reached.
- Bc: Burst Committed. This is the maximum burst size allowed. Unless a value is specified, it is Bc = 3 x MTU by default, i.e., the burst allowed is three times the size of the interface's MTU. If the MTU interface varies, the Bc also changes its value.
- Be: Burst Excess. This is the burst excess allowed. This parameter serves an administrative purpose only. Unlike the Frame Relay DE bit and the ATM CLP bit, it doesn't execute a direct action on the congestion control bits. The value by default is 0.

These three parameters help give the final burst the maximum size. The whole calculation is as follows:

(1) If Bc is specified, jump to the next section. If Bc isn't specified, then Bc = 3 x MTU.

(2) Limit Bc to a minimum of 7.8 ms (CIR / 128) and a maximum of 1s (CIR).

(3) Get the Bf final burst size. If Be is specified, then Bf = Bc + Be. If Be isn't specified, then Bf = Bc.

(4) Use CIR and Bf to limit throughput. This results in an average interval Tf = Bf / CIR.

These calculations result in the following:

- In no time interval, Tf receives more data than the BF. I.e., the average reception speed in any Tf time interval is equal or less than the configured CIR.
- In normal configurations (only the CIR is indicated), bursts are allowed. These can be as big as three times the value of the MTU's interface. Thus, the maximum burst size depends on the size of the interface's MTU.
- In advanced configurations (the CIR and the Bc are both indicated), the user has control over the average interval and can limit the throughput (always within the 7.8 ms and 1 s limits, mentioned above).
- By configuring the Be parameters, you can exceed the 1 second limit for the average interval. Usually, the final burst size is the sum of Bc and Be.

Example:

Configuring the *police 100* for an Ethernet interface.

- CIR = 100 kbps.
- Bc = 3 x MTU = 3 x 1500 bytes = 4500 bytes = 36 kbits
- Measurement interval (Tf) = Bc / CIR = 36 kbit / 100 kbps = 0.36 s = 360 ms.
- The device limits the throughput so that no interval of 360 ms is greater than an average throughput of 100 kilobits per second.

1.2 Class maps

A traffic class is made up of the following:

- Class name.
- One or more entries that define the classification criteria.
- Instructions as to how the classification criteria are evaluated, when more than one of these criteria are specified (**match-any**, **match-all**).

```
feature class-map
; -- Class-Map Menu Configuration -
class-map "Class_100"
         entry 1 default
         entry 1 permit
         entry 1 match access-list 100
;
         match all
      exit
;
   exit
```

When a packet is received, it is evaluated to see if it matches a specified classification criteria. If it does, the packet is considered to belong to said class.

Packets that don't match any of the criteria are filed under the default traffic class (if one has been defined).

When a class has multiple classification criteria, the evaluation can be executed using the **match any / match all** criteria. If you have specified **match any** as an evaluation criteria, the traffic being evaluated must coincide with some of the specified criteria. If you have specified **match all**, the evaluated traffic must coincide with all the criteria specified in the class.

1.2.1 Class-map elements

As already discussed, a class-map is made up of one or more entries that define the classification criteria for said class.

In addition, you can define how to evaluate the classification criteria when more than one has been specified.

1.2.1.1 Class-map entries

Each entry should, as a minimum, contain a classification criterion that defines the criteria. This way, a packet can be considered as belonging (or not) to a class (match clause).

The entries are applied in numerical order, marked by the identifiers, until one that matches is found. Once the packet has matched an entry, a parameter tells you if you should carry on evaluating the next entries or if the packet has already been classified. If no entries match, it is as if the packet matched a deny entry.

```
Class-map class_name> entry <n> ?
  default        Sets default values to an existing or a new entry
  match          Configures the match criteria for a class map
  permit         Configures type of entry as permit
  description    Sets a description text for this entry
```

1.2.1.2 Classification criteria: Definition

The following classification criteria can be applied to an entry:

```
Class-map class_name> entry <n> match ?
  not            Negate a command
  access-list    Filter network traffic using a predefined access control list
  cos            IEEE 802.1Q/ISL class of service/user priority values
  source         Source package
  destination    Destination package
```

1.2.1.2.1 Access lists

One standard or extended access control list can be specified per entry. These lists are used to check that the packets match the class and, therefore, determine if the traffic is considered incoming or not.

```
Class-map class_name> entry 1 match access-list ?
  <1..1999>    Previously created access list identifier
```

1.2.1.2.2 CoS value (Class of Service)

You can specify a CoS value for each entry. These values range from 0 to 7 and indicate the packet's priority. A value of 7 indicates the highest priority. This value is located in the packet's VLAN header.

```
Class-map class_name> entry 1 match cos ?
  <0..7>    Value in the range
```

If a class only has one entry and the packet matches the criteria configured in this entry, the packet is considered as belonging to this class. Otherwise, depending on the relevant criteria, you need to follow the instructions under section *c) Match any, match all* if the class contains more than one entry.

1.2.1.2.3 Source or destination MAC address

You can specify a source/destination MAC for each entry. In this case, you can also indicate if you want the packet to match the entry or not.

```
Class-map class_name>entry 1 match {source|destination} ?
  mac    Source MAC address
Class-map class_name>entry 1 match {source|destination} mac ?
  <mac>    MAC format
```

Not option:

```
Class-map class_name>entry 1 match not ?
  source         Source package
  destination    Destination package
[…]
Class-map class_name>entry 1 match not{source|destination} mac ?
  <mac>    MAC format
```

1.2.1.3 Match any, match all

These instructions indicate how to evaluate the classification criteria when there is more than one entry in the class.

If **match any** is configured, a packet must meet the classification criteria of at least one class entry to belong to said class. If **match all** is configured, the packet must meet all the classification criteria of all entries to belong to said class.

If neither one is configured, then **match all** is taken as default.

```
Class-map class_name> match ?
  all    All matching statements under this classmap (default)
  any    Some matching statements under this classmap
```

1.3 Policy maps

A traffic policy contains the following components:

- Policy name.
- One or several class-maps (previously created, or the default class if this is configured) to associate them with the traffic policy.
- These class-maps can have QoS policies associated with them (*set, police*), applied over the packets they classify.

```
   feature policy-map
; -- Policy-Map Menu Configuration --
      policy-map "Policy_100"
         class Class_100
;
         class Clase_ext configuration
            set label 20
            police 2000 500
            police link-layer
         exit
;
         class default
;
      exit
```

Associating traffic classes with the traffic policy results in “Service Policy”. This is applied to an interface through the **service-policy** command (please see manual *bintec-Dm 772-I* "*Common Configuration Interfaces*").

```
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.213.241 255.255.0.0
;
;
      service-policy Policy_100 input
;
      load-interval 60
   exit
```

In this case, the service policy allows us to calculate and monitor the input traffic rates in each interface (based on the classification criteria defined in the service classes).

Chapter 2 Configuration

2.1 Configuring class-map

To access the class map feature configuration menu, use the *feature class-map* command found in the main menu of the configuration console.

```
Config>feature class-map
-- Class-Map Menu Configuration --
Class-map Config>
```

Once in the class map configuration menu, use the configuration commands (detailed in the following sections) to define the class maps used in the Policy Map feature.

2.2 Configuration commands

The class map configuration menu contains the following commands:

Command	Function
class-map *id-class*	Defines a class map.
list *id-class*	Displays the class map indicated in *id-class*
list all	Displays all the class maps.
no class-map *id-class*	Eliminates a class map.
exit	Returns to the general configuration menu.

These commands are further explained below.

2.2.1 Class-map

Use this command to enter the configuration mode for the specified class map.

Syntax:

```
Class-map Config>class-map <id-class>
```

idclass	Name of the class map to define.

Example:

```
Class-map Config>class-map Class_100
Class-map Class_100>
```

2.2.1.1 Class-map default

If, when creating the class, you indicate that its name is “default”, a default class will be created with its corresponding configuration already established:

Example:

```
Class-map Config>class-map default

Class-map default>show conf
; Showing Menu and Submenus Configuration for access-level 15 ...
; XXX Router
        entry 1 default
        entry 1 permit
;
        match any
```

This class is defined so that when a packet is not accepted as belonging to a class, it’s sent to the default class. Since the classification criteria of the latter is to accept any packet, the packet will be filed under the default class (as you can see in the configuration).

A class map is made up of one or more numerically ordered entries.

When a class-map is processed it checks the packet with each entry and, depending on the type of configured *match* instruction (*all*/*any*), acts one way or the other.

- If a packet *matches* an entry and **match any** is configured, the packet is considered as belonging to said class without having to check the rest of the entries. If **match all** is configured, you have to continue checking the rest of the entries.
- If a packet *doesn't match* an entry and **match any** is configured, the packet is considered as not belonging to said class without having to check the rest of the entries. If **match all** is configured, you have to continue checking the rest of the entries until at least one of the entries matches.

The configuration menu for a class map has the following available commands:

Command	Function
description	Adds a text description that helps you understand how to use the class.
entry *n* default	Sets the default configuration in the specified entry.
entry *n* permit	Identifies the entry as PERMIT. All traffic fulfilling the classification criteria is considered as belonging to that class.
entry *n* match access-list {*listaacceso*}	Rules which access lists can be used to find out whether a packet belongs to a class or not.
entry *n* match cos *{cos value}*	Determines the CoS value that the packets must contain to belong to the class.
entry n match [not] source mac <MAC>	Determines the source MAC value that the packet must contain to belong to the class. If this has a "no" option, then the packet cannot contain the source MAC in order to belong to the class.
entry n match [not] destination mac <MAC>	Determines the destination MAC value that the packet must contain to belong to the class. If this has a "no" option, then the packet cannot contain the destination MAC in order to belong to the class.
entry *n* no match { *criteria match* }	If you select the "no" option in front of the match criteria, these criteria are deleted (but not the whole of the entry).
entry *n* description	Adds a text description that helps you understand the purpose of the entry or how to use it.
match all	Shows how to evaluate a packet to decide whether it belongs to a class. This value (all) means that the packet must match all the class classification criteria in order to belong to it.
match any	Shows how to evaluate a packet to decide whether it belongs to a class. This value (any) means that the packet must fulfill the classification criteria of at least one of the class entries in order to belong to said class.
no	Disables functions or resets the default value for some parameters.

2.2.1.2 DESCRIPTION

Allows you to add a text description to better understand the purpose or use of the class.

Syntax:

```
Class-map id-class>description ?
  <1..64 chars>    Description text
```

Example:

```
Class-map Class_100>description "Class ACL100"
Class-map Class_100>
```

2.2.1.3 ENTRY <id> DEFAULT

Sets all the parameters for an entry to their default values.

- PERMIT

Syntax:

```
Class-map id-class> entry <id> default
```

Example:

```
Class-map Class_100>entry 1 default
```

```
Class-map Class_100>
```

2.2.1.4 ENTRY <id> PERMIT

Identifies the entry as PERMIT. This parameter indicates that all traffic matching the classification criteria belongs to the class.

Syntax:

```
Class-map id-class>entry <id> permit
```

Example:

```
Class-map Class_100>entry 1 permit
Class-map Class_100>
```

2.2.1.5 ENTRY <id> MATCH ACCESS-LIST

Establishes the access list that determines if a packet belongs to the class or not.

Syntax:

```
Class-map id-class>entry <id> match access-list <id-access-list>
```

Example:

```
Class-map Class_100>entry 1 match access-list 100
Class-map Class_100>
```

2.2.1.6 ENTRY <id> MATCH COS

Establishes the CoS (Class of Service) value that the packet must contain to belong to the class.

Syntax:

```
Class-map id-class>entry <id> match cos <0…7>
```

Example:

```
Class-map Clase_100>entry 1 match cos 3
Class-map Clase_100>
```

2.2.1.7 ENTRY <id> MATCH {SOURCE/DESTINATION} MAC

Establishes the source/destination MAC value that the packet must contain to belong to the class.

Syntax:

```
Class-map id-class>entry <id> match {source|destination} mac <MAC>
```

Example:

```
Class-map list_permitted>entry 1 match source mac 00016C3BADE3
Class-map list_permitted>sho conf
; Showing Menu and Submenus Configuration for access-level 15 ...
; Default Router 0 0 Version 10.8.13-Alfa

        entry 1 default
        entry 1 permit
        entry 1 match source mac 00-01-6c-3b-ad-e3
;
        match any
```

Note

When configuring a permitted entry in the class-map, the ‘match-any’ option is automatically configured. This indicates that the entries configured in the class-map are not exclusive, as the packet belongs to the class where one of the entries matches.

2.2.1.8 ENTRY <id> MATCH NOT {SOURCE/DESTINATION} MAC

Establishes the source/destination MAC value that the packet must not contain to belong to the class.

Syntax:

```
Class-map id-class>entry <id> match not {source|destination} mac <MAC>
```

Example:

```
Class-map list_denied>entry 1 match not source mac 00016C3BADE3
Class-map list_denied>sho conf
; Showing Menu and Submenus Configuration for access-level 15 ...
; Default Router 0 0 Version 10.8.13-Alfa
         entry 1 default
         entry 1 permit
         entry 1 match not source mac 00-01-6c-3b-ad-e3
;
         match all
```

Note

When configuring a denied entry in the class-map, the ‘match-all’ option is automatically configured. This indicates that the entries configured in the class-map are exclusive since, if one of the entries doesn’t match, the packet doesn’t belong to the class.

2.2.1.9 ENTRY <id> NO MATCH {match criteria}

Allows you to delete a *match* criteria without having to delete the whole entry.

Syntax:

```
Class-map id-class>entry <id> no match ?
  not            Negate a command
  access-list    Filter network traffic using a predefined access control list
  cos            IEEE 802.1Q/ISL class of service/user priority values
  source         Source package
  destination    Destination package
```

Example:

```
Class-map list>entry 1 no match source mac
Class-map list1>entry 1 no match access-list
Class-map list2>entry 1 no match cos
```

2.2.1.10 ENTRY <id> DESCRIPTION

Allows you to add a text description to better understand the purpose or use of the entry.

Syntax:

```
Class-map id-class>entry <id> description ?
  <1..64 chars>    Description text for this entry
```

Example:

```
Class-map Class_100>entry 1 description "Entrada 1"
Class-map Class_100>
```

2.2.1.11 MATCH

Sets how to evaluate a packet to decide whether it belongs to the class or not.

Syntax:

```
Class-map id-class>match ?
  all    All matching statements under this classmap (default)
  any    Some matching statements under this classmap
```

- On indicating that these criteria are **all**, we mean that, for the packet to belong to the class, it must fulfill all the

classification criteria for all the entries in the class.

Example:

```
Class-map Class_100>match all
Class-map Class_100>
```

- On indicating that these criteria are **any**, we mean that, for the packet to belong to the class, it must fulfill the classification criteria of at least one of the entries in the class.

Example:

```
Class-map Class_100>match any
Class-map Class_100>
```

Note

If no type of criteria is specified, then ALL is taken by default.

2.2.1.12 NO

This command is used to disable functions or to reset the default values in some parameters.

Syntax:

```
Class-map id-class>no ?
  description    Configure a description for this class-map
  entry          Configure an entry
  match          Configure type of matching statements
```

2.2.1.12.1 NO ENTRY

Deletes an entry from the class-map. You need to enter the identifier of the entry you wish to eliminate.

Syntax:

```
Class-map id-class>no entry <id>
```

Example:

```
Class-map Class_100>no entry 1
Class-map Class_100>
```

2.2.1.12.2 NO DESCRIPTION

Deletes the text description associated with the class-map

Syntax:

```
Class-map id-class>no description
```

Example:

```
Class-map Class_100>no description
Class-map Class_100>
```

2.2.1.12.3 NO MATCH ALL/ANY

Deletes the match all/any parameter.

Syntax:

```
Class-map id-class>no match ?
  all    All matching statements under this classmap (default)
  any    Some matching statements under this classmap
```

Example:

```
Class-map Class_100>no match all
Class-map Class_100>
```

Note

When you delete one of these parameters, the criteria is established as ALL (which is its default value).

2.2.2 List

This command has two options: to display all the configured classes or to display the information on the class that is selected.

Syntax:

```
Class-map Config>list ?
  <1..30 chars>    Class-map name
  all              Display all class map configuration
Class-map Config>list <id-class>
```

id-class	Name of the class-map you want to display.

Examples:

```
Class-map Config>list Class_100
-- Class Map "Class_100" --
Class Map Entries
-----------------
1     PERMIT  MATCH ACCESS-LIST = 100
Class-map Config>list all

-- Class Map "Class_100" --
Class Map Entries
-----------------
1     PERMIT  MATCH ACCESS-LIST = 100
-- Class Map "Class_200" --
Class Map Entries
-----------------
1     PERMIT  MATCH ACCESS-LIST = 200
-- Class Map "default" --
Class Map Entries
-----------------
1     PERMIT
```

2.2.3 No

This command deletes the definition of a class-map, provided that it isn't assigned to a policy-map, a port or to a *repeater*. If this happens, a message comes up indicating that you have to eliminate this assignment first.

Syntax:

```
Class-map Config>no class-map <id-class>
```

id-class	Name of the class-map to eliminate.

Examples:

```
Class-map Config>no class-map Clase_200
Class-map Config>
Class-map Config>no class-map Class_100
CLI Error: This class map is assigned to a Policy Map. First you must clear all
assignments
CLI Error: Command error
```

2.3 Policy-Map

To access the policy-map feature configuration menu, use the *feature policy-map* command found in the main menu of the configuration console.

```
Config>feature policy-map
-- Policy-Map Menu Configuration --
Policy-map Config>
```

Once in the policy-map configuration menu, use the configuration commands detailed in the following sections to define the class associated with the corresponding traffic policy.

2.4 Configuration commands

The policy-map configuration menu contains the following commands:

Command	Function
list	Displays the configured policy-maps.
no	Deletes a policy-map.
policy-map *id-policy*	Defines a policy-map.
exit	Returns to the general configuration menu.

The following sections explain these commands.

2.4.1 Policy-map

Use this command to enter the configuration mode for the specified policy-map.

Syntax:

```
Policy-map Config>policy-map <id-policy>
```

idpolicy	Name of the policy-map to define.

Example:

```
Policy-map Config>policy-map Policy_100
Policy-map Policy_100>
```

The configuration menu for a policy map contains the following commands:

Command	Function
class *id-class* [configuration]	Defines the classes associated with the policy. In cases where you select the 'configuration' option, you access the QoS (*Quality of Service)* policy menu.
description	Adds a text description that explains how to use the policy.
no	Allows you to delete both the class and the policy description.
shared	Allows you to share the same policy-map info when applying it on different interfaces through a **service-policy**.
exit	Returns to the previous menu.

2.4.1.1 Class <id-class> [configuration]

This command allows you to configure the classes you want to associate with the policy. Unless the class was previously defined, an error message will appear.

Examples:

```
Policy-map Policy_100>class ?
  <1..30 chars>    Class-map name
Policy-map Policy_100>class Clase_100
Policy-map Policy_100>sho conf
; Showing Menu and Submenus Configuration for access-level 15 ...
; XXX Router
         class Clase_100
Policy-map Policy_100>class Clase_200
CLI Error: Class-map not found
CLI Error: Command error
```

In addition, policies or actions can be added to the class. These are applied to the packets classified under said class.

To configure the actions, access the class submenu using the '*configuration*' option (entered after the name of the class).

The configuration menu for actions applied to the packets filed under said class is as follows:

Command	Function
no	Allows you to delete the actions associated with said class.
police	Allows you to set a maximum throughput limit.
report	Allows you to report statistics to a Generic-input NSLA filter.
set	Allows you to set the actions for the class.
timer-bps	Allows you to set the statistics refresh interval.
exit	Returns to the previous menu.

2.4.1.1.1 police

The *police* command allows you to limit the maximum throughput of an interface and indicate at what level you will carry out the throughput calculation.

The maximum throughput is specified in kilobits per second, while the *burst* and the *excess-burst* are specified in kilobits.

The quantity of bytes received are measured at the IP level or at the link level (*link-layer* and *network–layer* options), as the final throughput can be somewhat bigger depending on the interface headers.

Command	Function
police <cir> [<bc> [<be>]]	Limits the maximum average throughput.
police link-layer [offset <value>]	Indicates the limitation of the calculation carried out at layer 2 or the link layer.
police network-layer	Indicates the limitation of the calculation carried out at layer 3 or the network layer (default value).

police <cir> [<bc> [<be>]]

Allows you to limit the maximum average throughput for an interface.

Syntax:

```
Pmap Policy_100-Cmap Class_100>police <cir> [<bc> [<be>]]
```

`cir`	Specifies the maximum medium throughput.
`bc`	(Optional) Maximum burst size permitted.
`be`	(Optional) Maximum size of the permitted burst excess.

Example:

```
Pmap Policy_100-Cmap Class_100>police 1000 5
```

police link-layer [offset <value>]

Allows you to indicate the throughput limitation calculations carried out at layer 2 or the link layer.

Syntax:

```
Pmap Policy_100-Cmap Class_100>police link-layer [offset <value>]
```

`link-layer`	Indicates that the calculations are carried out at layer 2 or at the link layer.
`offset`	(Optional) Offset to bear in mind when carrying out the calculation (bytes)

Example:

```
Pmap Policy_100-Cmap Class_100>police link-layer offset 10
```

police network-layer

Allows you to indicate the limitation calculations for the throughput carried out at layer 3 or the network layer.

Syntax:

```
Pmap Policy_100-Cmap Class_100>police network-layer
```

network-layer	Indicates the calculation carried out at layer 3 or the network layer (default value)

Example:

```
Pmap Policy_100-Cmap Class_100>police network-layer
```

2.4.1.1.2 report

The *report* command allows you to send class statistics to the Generic-Input NSLA Filter selected once the refresh interval for statistics is defined. For further information about NSLA filters, see manual *bintec-Dm 754-I "NSLA"*.

For now, only statistic *RATE-KPBS* can be reported (the input throughput specified in kilobits per second).

Example:

```
Pmap Policy_100-Cmap Class_100>report ?
  rate-kbps    Report rate (Kbps)
Pmap Policy_100-Cmap Class_100>report rate-kbps ?
  nsla-filter    NSLA Filter that gets information
Pmap Policy_100-Cmap Class_100>report rate-kbps nsla-filter ?
  <1..65535>    Filter identifier
Pmap Policy_100-Cmap Class_100>report rate-kbps nsla-filter 1
Pmap Policy_100-Cmap Class_100>
```

The same RATE-KBPS statistic can be reported to more than one NSLA filter, if desired. To do this, type each selected filter in successive *report* commands.

Example:

```
Pmap Policy_100-Cmap Class_100>report rate-kbps nsla-filter 1
Pmap Policy_100-Cmap Class_100>report rate-kbps nsla-filter 2
```

To delete the report, use the negated form of this command.

Note

When applying the policy-map on different interfaces through a **service-policy** (see manual *bintec-Dm 772-I "Common Configuration Interfaces"*), they are seen as independent and the statistics gathered correspond to one interface only. However, if the *report* command is configured to notify statistics to some Generic-Input NSLA Filter, these filters are shared between the interfaces that are configured *with the same policy-map*, making the reported value unpredictable.

Therefore, please make sure you use different Generic-Input NSLA Filters belonging to different policy-maps for different interfaces.

Command history:

Release	Modification
11.00.05	This command was introduced as of version 11.00.05.
11.01.00	This command was introduced as of version 11.01.00.

2.4.1.1.3 set

The *set* command accesses the menu for the different actions that can be applied to the classified traffic in the class.

At the moment, you can only configure a label in order to tag the packet:

Command	Function
set label <value>	Label to tag the packet.

set label <value>

Allows you to label the traffic classified in the class with a tag.

Syntax:

```
Pmap Policy_100-Cmap Class_100>set label <value>
```

value	Label value [0-99]

Example:

```
Pmap Policy_100-Cmap Class_100>set label 10
```

2.4.1.1.4 timer-bps

The *timer-bps* command sets the refresh interval of the statistics. By default, this interval is 10 seconds.

Example:

```
Pmap Policy_100-Cmap Class_100>report ?
  rate-kbps    Report rate (Kbps)
Pmap Policy_100-Cmap Class_100>timer-bps ?
  <1..3600>    Time in seconds
Pmap Policy_100-Cmap Class_100>timer-bps 5
Pmap Policy_100-Cmap Class_100>
```

To re-establish the default interval use the negated form of this command.

Command history:

Release	Modification
11.00.05	This command was introduced as of version 11.00.05.
11.01.00	This command was introduced as of version 11.01.00.

2.4.1.2 Shared

This command allows you to share the same policy-map info when applying the policy-map on different interfaces through a **service-policy** (see manual *bintec-Dm 772-I* "*Common Configuration Interfaces*").

This function is disabled by default. When the policy-map is applied on several interfaces, they are seen as independent and the statistics gathered correspond to one interface only. However, when this function is enabled, the statistics gathered when the policy-map is applied on several interfaces correspond to the sum of these.

Examples:

```
Policy-map Policy_100>shared
Policy-map Policy_100>
```

Command history:

Release	Modification
11.00.05	This command was introduced as of version 11.00.05.
11.01.00	This command was introduced as of version 11.01.00.

2.4.2 List

This command has two options: to display all the configured policies or to display the information on the indicated policy.

Syntax:

```
Policy-map Config>list ?
  <1..15 chars>    Policy-map name
  all       Display all policy map configuration
Policy-map Config>list <id-policy>
Policy-map Config>list all
```

`idpolicy`	Name of the policy-map you want to display.
`all`	Displays all the configured policy map policies.

Example:

```
Policy-map Config>list Policy_100
-- Policy Map "Policy_100" --
Policy-Map Classes
------------------
Policy Map can be SHARED
   Class-Map Name = "Class_100"
     QoS Policies
```

```
      Police
        CIR = 100 (kbps)
        Layer = link-layer
      Set
        Label = 30
    Refresh Timer-bps 5 seconds
    Report rate-kbps to NSLA filter 1
  Class-Map Default = "default"
Policy-map Config>list all
-- Policy Map "Policy_100" --
Policy-Map Classes
------------------
Policy Map can be SHARED
  Class-Map Name = "Class_100"
    QoS Policies
      Police
        CIR = 100 (kbps)
        Layer = link-layer
      Set
        Label = 30
    Refresh Timer-bps 5 seconds
    Report rate-kbps to NSLA filter 1
  Class-Map Default = "default"
-- Policy Map "Policy_200" --
Policy-Map Classes
------------------
  Class-Map Name = "Clase_200"
  Class-Map Default = "default"
```

Command history:

Release	Modification
11.00.05	The output of this command was modified in version 11.00.05 to show info related to the *shared* command configured in the policy-map, and the *report* and *timer-bps* commands configured under class.
11.01.00	The output of this command was modified in version 11.01.00 to show info related to the *shared* command configured in the policy-map, and the *report* and *timer-bps* commands configured under class.

2.4.3 No

This command deletes the definition of a policy map, provided that it isn't assigned to a service-map. If this happens, a message comes up indicating that you have to eliminate this assignment first.

Syntax:

```
Policy-map Config>no policy-map <id-policy>
```

`id-policy`	Name of the policy-map to eliminate

Examples:

```
Policy-map Config>no policy-map Policy_100
Policy-map Config>no policy-map Policy_100
CLI Error: This policy map is assigned to a Service Policy. First you must clear all
its assignments before deleting
CLI Error: Command error
```

Chapter 3 Monitoring

3.1 Monitoring commands

This section details the commands we can use for the Policy-Map feature monitoring tools. To enter these commands, you need to access the Policy-Map feature monitoring prompt.

To access the Policy-Map feature monitoring area, enter the **FEATURE POLICY-MAP** command at the general monitoring prompt (+).

Example:

```
+feature policy-map
-- Policy Map user console --
Policy-Map+
```

The following commands are available in the Policy-Map feature monitoring environment:

Command	Function
clear	This sets the statistics to zero.
list	Displays the policy-maps configuration.
exit	Returns to the general monitoring menu.

3.1.1 LIST

The **list** command has several options. Depending on which one you choose, you'll get different types of information regarding the policy-maps.

Syntax:

```
Policy-Map+list policy-map ?
  interface    Display statistics by interface
  name         Display a policy map
  <cr>         Display all policies
```

3.1.1.1 list policy-map

This command displays all the configured policy-maps with their associated class-maps.

Syntax:

```
Policy-Map+list policy-map
```

Example:

```
Policy-Map+list policy-map
-- Policy Map "Policy_100" --
Policy-Map Classes
------------------
   Class-Map Name = "Class_100"
     QoS Policies
       Police
         CIR = 100 (kbps)
         Layer = link-layer
       Set
         Label = 30
     Refresh Timer-bps 5 seconds
     Report rate-kbps to NSLA filter 1
   Class-Map Default = "default"
-- Policy Map "Policy_200" --
Policy-Map Classes
------------------
   Class-Map Name = "Clase_200"
   Class-Map Default = "default"
```

Command history:

Release	Modification
11.00.05	The output of this command was modified in version 11.00.05 to show info related to the *report* and *timer-bps* commands configured under class.
11.01.00	The output of this command was modified in version 11.01.00 to show info related to the *report* and *timer-bps* commands configured under class.

3.1.1.2 list policy-map name <id-policy>

This command displays the class-maps and their actions, associated with the policy-map under*id-policy*.

Syntax:

```
Policy-Map+list policy-map name <id-policy>
```

Example:

```
Policy-Map+list policy-map name Policy_100
-- Policy Map "Policy_100" --
Policy-Map Classes
------------------
   Class-Map Name = "Class_100"
     QoS Policies
       Police
         CIR = 100 (kbps)
         Layer = link-layer
       Set
         Label = 30
     Refresh Timer-bps 5 seconds
     Report rate-kbps to NSLA filter 1
   Class-Map Default = "default"
```

Command history:

Release	Modification
11.00.05	The output of this command was modified in version 11.00.05 to show info related to the *report* and *timer-bps* commands configured under class.
11.01.00	The output of this command was modified in version 11.01.00 to show info related to the *report* and *timer-bps* commands configured under class.

3.1.1.3 list policy-map name<id-policy> class-map <id-class>

This command displays information on certain class-map entries within a specific policy-map:

Syntax:

```
Policy-Map+list policy-map name <id-policy> class-map <class>
```

Example:

```
Policy-Map+list policy-map name Policy_100 class-map Clase_100
-- Policy Map "Policy_100" --
-- Class Map "Clase_100" --
Class Map Entries
-----------------
1     PERMIT  MATCH ACCESS-LIST = 200
1     PERMIT  MATCH COS = 3
```

3.1.1.4 list policy-map interface <interface> input

Once you have the traffic classes and the policy-maps defined, you can apply the policy-maps in an interface through a **service-policy** (see manual *bintec-Dm 772-I "Common Configuration Interfaces")*.

Through this command, you can see the statistics for the interfaces where we have applied the service-policies and see what input traffic is accepted in each class associated with the policy-maps, as well as the action statistics that are carried out over the classified traffic.

Right now, this only works for input traffic.

Syntax:

```
Policy-Map+list policy-map interface <interface> input
```

Example 1:

```
Policy-Map+list policy-map interface ethernet0/0.2 input
-- Interface ethernet0/0.2 --
 Service policy input: "Policy_100"
 Policy map is being SHARED with other interfaces
   Class map "Clase_100" (match: all)
     48 packets, 69600 bytes
     instant received rate 10440 bps, not match rate 0 bps
     5 minutes offered rate 1726 bps, drop rate 0 bps
     Match:
            access-list 200
            cos 3
   Class map "default" (match: any)
     16 packets, 21814 bytes
     instant received rate 10440 bps, not match rate 0 bps
     5 minutes offered rate 154 bps, drop rate 0 bps
```

Example 2:

```
Policy-Map+list policy-map interface ethernet0/0 input
-- Interface ethernet0/0 --
 Service policy input: "Policy_200"
   Class map "Class_200" (match: all)
     45 packets, 66690 bytes
     instant received rate 58174 bps, not match rate 4822 bps
     5 minutes offered rate 40489 bps, drop rate 14811 bps
     Match:
            access-list 100
     QoS Set:
            label 20
               packets marked 45
```

Example 3:

```
Policy-Map+list policy-map interface ethernet0/0 input
-- Interface ethernet0/0 --
 Service policy input: "Policy_300"
    Class map "Class_300" (match: all)
     316 packets, 392472 bytes
     instant received rate 6660 bps, not match rate 2686 bps
     5 minutes offered rate 10383 bps, drop rate 3471 bps
     Match:
            access-list 300
     Police:
            cir 100 kbps
            burst 37000 kbits
            operating on link-layer
            conformed 206 packets, 255852 bytes; action: receive
            exceeded  110 packets, 136620 bytes; action: drop
```

The following sections describe each of the service policy monitoring fields:

3.1.1.4.1 interface <name>

Name of the interface we are monitoring.

Example:

```
     -- Interface ethernet0/0 --
```

3.1.1.4.2 Policy map is being SHARED with other interfaces

If this appears, it indicates that this policy map was configured as *shared* and is applied to another interface too, so the statistics are the sum of all interfaces.

3.1.1.4.3 service policy input: <policy-map name>

Name of the service policy applied at the input of the previous interface.

Example:

```
Service policy input: "Policy_300"
```

3.1.1.4.4 class map <name> (match: <type>

Name of traffic class we are monitoring. If there is more than one class, one is shown below the other.

The type of match (*all*/*any*) indicates whether or not all the classification criteria must be evaluated to classify the traffic under a certain class.

Example:

```
Class map "Class_300" (match: all)
```

3.1.1.4.5 # packets, #bytes

Number of packets and bytes that match the classification criteria defined in the class. These counter increase depending on whether there is congestion or not.

Example:

```
316 packets, 392472 bytes
```

3.1.1.4.6 instant received rate # bps, not match rate # bps

Shows the instant rate that updates every 10 seconds.

Command	Function
received rate # bps	This is the instant rate in bits per second for the received traffic.
not match rate # bps	This is the instant rate in bits per second for the received traffic that is not classified under the class.

Example:

```
instant received rate 6660 bps, not match rate 2686 bps
```

3.1.1.4.7 # {seconds/minutes} offered rate # bps, drop rate # bps

This shows the balance ratio that depends on the transfer rate interval (**load-interval**)

Command	Function
`# {seconds|minutes}`	This is the value configured as the transfer task interval in each interface: *load-interval*. For more information, please see manual *bintec-Dm 772-I* "*Common Configuration Interfaces*".
`offered rate # bps`	This is the rate in bits per second of all the traffic offered to the class, calculated on the basis of the exponential growth and decrease (depending on the interface's incoming traffic).
`drop rate # bps`	This is the rate in bits per second of the traffic that the class has dropped. It is calculated by the same means as the above statistics. This only make sense if *traffic policing* is enabled.

Exponential ratio calculus:

```
new_average = ( (previous_average - interval) *exp(-t/C) ) + interval
```

- As you can see, the calculation is carried out based on the previous calculations (**previous_average**).
- **exp(-t/C):** this is the “decay factor” and is the part that indicates to what extent the oldest samples are taken into account. The bigger ‘C’is (**C** is **load-interval**, [30, 600] seconds), the more significant the older samples will be in the calculation.
- Samples are taken every 5 seconds, which is the time that corresponds to the ‘t’ parameter.
- The new samples taken during these 5 seconds are found in: **interval**.

Example:

```
5 minutes offered rate 10383 bps, drop rate 3471 bps
```

3.1.1.4.8 Match

Shows the classification criteria used to classify traffic.

Example:

```
    Match:
           access-list 200
           cos 3
```

3.1.1.4.9 QoS set

This displays the group of actions applied to the traffic classified in the class.

Field	Description
`label #`	This is the label used to mark the packets classified in the class.
`packets marked #`	Number of packets marked with this label.

Example:

```
    QoS Set:
           label 20
              packets marked 45
```

3.1.1.4.10 Police

This shows that *traffic policing* is enabled. It shows the value of the CIR, Bc and Be. This also shows the statistics of the packets and bytes that have been received and those that have been dropped because the configured limit has been exceeded.

Field	Description
`cir # kbps`	Value of *committed information rate* in kilobits per second.
`bc # kbits`	Value of *burst committed* in kilobits.
`be # kbits`	Value of *burst excess* in kilobits.
`operating on <layer>`	Shows at which layer the calculation of the limitation (l*ink-layer* | *network-layer)* is being carried out.
`conformed # packets, # bytes; action: receive`	Number of packets and bytes that do not exceed the configured throughput limit. Action to carry out with these packets (by default, these are received).
`exceded # packets, # bytes; action: drop`	Number of packets and bytes that exceed the configured throughput limit. Action to carry out with these packets (by default, these are dropped).

Example:

```
    Police:
           cir 100 kbps
           burst 37000 bits
           operating on link-layer
           conformed 206 packets, 255852 bytes; action: receive
           exceeded  110 packets, 136620 bytes; action: drop
```

Command history:

Release	Modification
11.00.05	The output of this command was modified in version 11.00.05 to show if the policy map was being shared between interfaces.
11.01.00	The output of this command was modified in version 11.01.00 to show if the policy map was being shared between interfaces.

3.1.2 Clear

This command sets all statistics back to zero. You can delete all the statistics belonging to a single interface, or those of a given policy-map.

Syntax:

```
Policy-Map+clear <interface> ?
  <word>    Name of the policy-map to reset
  <cr>      Reset interface hits
```

3.1.2.1 CLEAR <interface> <id-policy>

If you indicate the name of a policy-map that is applied to the interface, this will only reset the statistics of this policy-map.

Syntax:

```
Policy-Map+clear <interface> <id-policy>
```

Example:

```
Policy-Map+clear ethernet0/0 Policy_100
```

Here we can see the results:

```
Policy-Map+list policy-map interface ethernet0/0.2 input
-- Interface ethernet0/0.2 --
 Service policy input: "Policy_100"
   Class map "Clase_100" (match: all)
     0 packets, 0 bytes
     instant received rate 0 bps, not match rate 0 bps
     5 minutes offered rate 0 bps, drop rate 0 bps
     Match:
            access-list 200
            cos 3
[…]
```

3.1.2.2 CLEAR <interface>

In this case, all the statistics for the classes that are associated with the policy-maps in the indicated interface are deleted.

Syntax:

```
Policy-Map+clear <interface>
```

Example:

```
Policy-Map+clear ethernet0/0
```

We can see the results here:

```
Policy-Map+list policy-map interface ethernet0/0.2 input
-- Interface ethernet0/0.2 --
 Service policy input: "Policy_100"
   Class map "Clase_100" (match: all)
     0 packets, 0 bytes
     instant received rate 0 bps, not match rate 0 bps
     5 minutes offered rate 0 bps, drop rate 0 bps
     Match:
            access-list 200
            cos 3
   Class map "default" (match: any)
     0 packets, 0 bytes
     instant received rate 0 bps, not match rate 0 bps
     5 minutes offered rate 0 bps, drop rate 0 bps
```

Chapter 4 Examples

4.1 Classification example with access-list

```
; Showing Menu and Submenus Configuration for access-level 15 ...
; XXX Router
   log-command-errors
   no configuration
   set inactivity-timer disabled
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.0 255.255.254.0
;
      exit
;
   exit
;
   feature class-map
; -- Class-Map Menu Configuration --
      class-map "Clase_100"
         entry 1 default
         entry 1 permit
         entry 1 match access-list 100
;
      exit
;
      class-map "default"
         entry 1 default
         entry 1 permit
;
         match any
      exit
;
   exit
;
   feature policy-map
; -- Policy-Map Menu Configuration --
      policy-map "Policy_100"
         class Clase_100
;
         class default
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.213.241 255.255.254.0
;
;
      service-policy Policy_100 input
;
      load-interval 30
   exit
;
;
;
   dump-command-errors
   end
```

You can see here what the statistics for this configuration will be:

```
Policy-Map+list policy-map interface ethernet0/0 input
-- Interface ethernet0/0 --
 Service policy input: "Policy_100"
   Class map "Clase_100" (match: all)
     103 packets, 124476 bytes
     instant received rate 13408 bps, not match rate 0 bps
     30 seconds offered rate 11360 bps, drop rate 0 bps
     Match:
            access-list 100

   Class map "default" (match: any)
     50 packets, 3734 bytes
     instant received rate 342 bps, not match rate 0 bps
     30 seconds offered rate 254 bps, drop rate 0 bps
```

4.2 Classification example with COS labeling

We have a scenario where we want to classify the traffic in a device that has a bridge configured and additionally receives traffic prioritized through CoS.

We want to sort incoming traffic labeled with value 2 CoS into the *Oro* class, traffic labeled with value 5 CoS into the *Multimedia* class and allocate the rest to the default class.

Since the device has a bridge configured, the packet's labeling information is lost when the traffic enters the device.

This example shows how to avoid losing this information (i.e., how to transfer the CoS value from one side of the bridge to the other).

To do this, we need to configure a service policy at the bridge entrance that allows us to classify the traffic based on its CoS value and, additionally, tag this traffic with a label.

4.2.1 Configuring the service policy

We need to configure the classes that allow us to classify the traffic depending on the CoS and the policies that apply to the labeling of the classified traffic.

Class-maps:

```
Config>feature class-map
-- Class-Map Menu Configuration --
Class-map Config>sho conf
; Showing Menu and Submenus Configuration for access-level 15 ...
; XXX Router
      class-map "cos2"
         entry 1 default
         entry 1 permit
         entry 1 match cos 2
;
      exit
;
      class-map "cos5"
         entry 1 default
         entry 1 permit
         entry 1 match cos 5
;
      exit
;
```

Policy-maps:

```
Config>feature policy-map
-- Policy-Map Menu Configuration --
```

```
Policy-map Config>sho conf
; Showing Menu and Submenus Configuration for access-level 15 ...
; XXX Router
      policy-map "poli_cos"
         class cos2 configuration
            set label 2
         exit
;
         class cos5 configuration
            set label 5
         exit
;
      exit
;
```

As we can see, the classes used to classify the incoming traffic have been defined and so has the label used to tag this traffic.

We now need to apply the policy in the subinterfaces:

```
Config>network ethernet0/0.11
ethernet0/0.11 config>sho conf
; Showing Menu and Submenus Configuration for access-level 15 ...
; XXX Router
;
;
      service-policy poli_cos input
;
      encapsulation dot1q 11
;
;
ethernet0/1.12 config>sho conf
; Showing Menu and Submenus Configuration for access-level 15 ...
; XXX Router
;
;
      service-policy poli_cos input
;
      encapsulation dot1q 12
;
;
```

Tagging traffic with a label is used to classify the traffic at the tunnel exit. BRS criteria allow you to classify the packet based on the value of its label.

4.2.2 Configuring classification in BRS

In this case, we need to configure the classes where the traffic is going to be finally classified.

- Class Oro for traffic labeled with a CoS value equal to 2.
- Class Multimedia for traffic labeled with a CoS value equal to 5.
- Class default for the remaining traffic.

To sort the traffic into these classes, classification is carried out based on the packet label value. To do this, we use the values applied in the pre-configured class policies.

The traffic with a CoS value equal to 2 has been tagged with a label with a value of 2 and the traffic with a CoS value equal to 5 has been tagged with a label with a value of 5.

Therefore, packets with a "2" label are sorted into Class Oro and packets with a "5" label are sorted into the Multimedia class.

Configuring the BRS classes:

```
BRS [i #] Config>sho conf
; Showing Menu and Submenus Configuration for access-level 15 ...
; XXX Router
         enable
         class local 10
```

```
;
        class default 20
;
        class oro 80
;
        class mm 100 real-time
        class mm rate-limit 100
```

Configuring the classification criteria for BRS:

```
BRS [i #] Config>sho conf
; Showing Menu and Submenus Configuration for access-level 15 ...
; XXX Router
        enable
        class local 10
;
        class default 20
;
        class oro 80
;
        class mm 100 real-time
        class mm rate-limit 100
;
;
        match label 2 class oro normal set cos 2
        match label 5 class mm urgent set cos 5
        match label 0 class default normal set cos 0
;
```

Packets classified under the default class are labeled with a CoS value equal to 0.

The resulting configuration is as follows:

```
; Showing Menu and Submenus Configuration for access-level 15 ...
; XXX Router
   log-command-errors
   no configuration
   set inactivity-timer disabled
   add device eth-subinterface ethernet0/0 11
   add device eth-subinterface ethernet0/1 12
   add device bvi 0
   feature class-map
; -- Class-Map Menu Configuration --
      class-map "cos2"
         entry 1 default
         entry 1 permit
         entry 1 match cos 2
;
      exit
;
      class-map "cos5"
         entry 1 default
         entry 1 permit
         entry 1 match cos 5
;
      exit
;
   exit
;
   feature policy-map
; -- Policy-Map Menu Configuration --
      policy-map "poli_cos"
         class cos2 configuration
            set label 2
         exit
;
         class cos5 configuration
            set label 5
         exit
```

```
;
      exit
;
   exit
;
;
;
   network ethernet0/0.11
; -- Ethernet Subinterface Configuration --
      service-policy poli_cos input
;
      encapsulation dot1q 11
;
;
;
   exit
;
;
;
   network ethernet0/1.12
; -- Ethernet Subinterface Configuration --
      service-policy poli_cos input
;
      encapsulation dot1q 12
;
;
   exit
;

   protocol asrt
; -- ASRT Bridge user configuration --
      bridge
      port ethernet0/0.11 1
      port ethernet0/1.12 2
   exit
;
;
;
   feature bandwidth-reservation
; -- Bandwidth Reservation user configuration --
      network ethernet0/0.11
         enable
         class local 10
;
         class default 20
;
         class oro 70
;
         class mm 100 real-time
         class mm rate-limit 1000
;
;
         match label 2 class oro normal set cos 2
         match label 5 class mm normal set cos 5
         match label 0 class default normal set cos 0
;
      exit
;
      network ethernet0/1.12
         enable
         class local 10
;
         class default 20
;
         class oro 70
;
         class mm 100 real-time
```

```
        class mm rate-limit 1000
;
;
        match label 2 class oro normal set cos 2
        match label 5 class mm normal set cos 5
        match label 0 class default normal set cos 0
;
     exit
;
   exit
;
   dump-command-errors
   end
```