Configuring NBAR Using the MQC

You can configure Network-Based Application Recognition (NBAR) using the functionality of the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC). The MQC uses traffic classes and traffic policies (policy maps) to apply QoS features to classes of traffic and applications recognized by NBAR.

This module contains concepts and tasks for configuring NBAR using the MQC.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Configuring NBAR Using the MQC

Before configuring NBAR using the MQC, read the information in the "Classifying Network Traffic Using NBAR" module.

Information About NBAR Coarse-Grain Classification

NBAR and the MQC Functionality

To configure NBAR using the MQC, you must define a traffic class, configure a traffic policy (policy map), and then attach that traffic policy to the appropriate interface. These three tasks can be accomplished by using the MQC. The MQC is a command-line interface that allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach these traffic policies to interfaces.

In the MQC, the **class-map** command is used to define a traffic class (which is then associated with a traffic policy). The purpose of a traffic class is to classify traffic.

Using the MQC to configure NBAR consists of the following:

- Defining a traffic class with the **class-map** command.
- Creating a traffic policy by associating the traffic class with one or more QoS features (using the **policy-map** command).
- Attaching the traffic policy to the interface with the **service-policy** command.

A traffic class contains three major elements: a name, one or more **match** commands, and, if more than one **match** command exists in the traffic class, an instruction on how to evaluate these **match** commands (that is, **match-all** or **match-any**). The traffic class is named in the **class-map** command line; for example, if you enter the **class-map cisco** command while configuring the traffic class in the CLI, the traffic class would be named "cisco."

The **match** commands are used to specify various criteria for classifying packets. Packets are checked to determine whether they match the criteria specified in the **match** commands. If a packet matches the specified criteria, that packet is considered a member of the class and is forwarded according to the QoS specifications set in the traffic policy. Packets that fail to meet any of the matching criteria are classified as members of the default traffic class.

Note For NBAR, the **match protocol** commands are used to specify the match criteria, as described in the NBAR and the match protocol Commands, on page 2.

NBAR and the match protocol Commands

NBAR recognizes specific network protocols and network applications that are used in your network. Once a protocol or application is recognized by NBAR, you can use the MQC to group the packets associated with those protocols or applications into classes. These classes are grouped on the basis of whether the packets conform to certain criteria.

For NBAR, the criterion is whether the packet matches a specific protocol or application known to NBAR. Using the MQC, network traffic with one network protocol (citrix, for example) can be placed into one traffic class, while traffic that matches a different network protocol (gnutella, for example) can be placed into another traffic class. Later, the network traffic within each class can be given the appropriate QoS treatment by using a traffic policy (policy map).

You specify the criteria used to classify traffic by using a **match protocol**command. The table below lists some of the available **match protocol**commands and the corresponding protocol or traffic type recognized and supported by NBAR.

Note For a more complete list of the protocol types supported by NBAR, see the "Classifying Network Traffic Using NBAR" module.

Table 1: match protocol Commands and Corresponding Protocol or Traffic Type

match protocol Command[1]	Protocol Type
match protocol (NBAR)	Protocol type supported by NBAR
match protocol citrix	Citrix protocol
match protocol fasttrack	FastTrack peer-to-peer traffic
match protocol gnutella	Gnutella peer-to-peer traffic
match protocol http	Hypertext Transfer Protocol
match protocol rtp	Real-Time Transport Protocol traffic
match protocol unknown [final]	All unknown and/or unclassified traffic

[1] Cisco IOS match protocol commands can vary by release. For more information, see the command documentation for the Cisco IOS release that you are using.

How to Configure NBAR Using the MQC

Configuring DSCP-Based Layer 3 Custom Applications

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar custom** *name* **transport** {**tcp** | **udp** | **udp-tcp** }**id** *id*
4. **dscp** *dscp-value*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nbar custom** *name* **transport** {**tcp** | **udp** | **udp-tcp** }**id** *id* **Example:** `Device(config)# ip nbar custom mycustom transport tcp id 100`	Specifies the transport protocol to match as TCP, UDP, or both TCP and UDP, and enters custom configuration mode.
Step 4	**dscp** *dscp-value* **Example:** `Device(config-custom)# dscp ef`	Specifies the differentiated service code points (DSCP) value. **Note** In cases where two custom applications have the same filters, the priority is set according to the order of configuration.
Step 5	**exit** **Example:** `Device(config-custom)# exit`	Exits custom configuration mode.

Managing Unclassified and Unknown Traffic

Some protocols require the analysis of more than one packet for NBAR classification. So packets sent until such a classification occurs are considered **unknown**. **unknown final** excludes these temporarily classified packets, and includes only those packets that are determined as unknown after the NBAR classification process.

By default, all traffic not matched to the unknown, are matched to a default class, as is the case with MQC.

Before you begin

Ensure that NBAR is fully configured. If NBAR is configured to match only a partial set of protocols, then all inactivate protocols are considered as unclassified traffic and hence unknown.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] **unknown**
4. **match protocol unknown [final]**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map** [**match-all** | **match-any**] **unknown** **Example:** `Device(config)# class-map match-all my-unknown`	Creates a class map to be used for matching unknown traffic to a new class and enters class-map configuration mode.
Step 4	**match protocol unknown [final]** **Example:** `Device(config-cmap)# match protocol unknown final`	Configures NBAR to match unknown traffic. • The **unknown** keyword signifies any traffic that is unclassified • The **unknown final** signifies traffic that is determined by NBAR as unknown.
Step 5	**end** **Example:** `Device(config-cmap)# end`	(Optional) Returns to privileged EXEC mode.

You can now configure the following tasks

1. Configuring a Traffic Policy
2. Attaching a Traffic Policy to an Interface or sub-interface

Configuring a Traffic Policy

Traffic that matches a user-specified criterion can be organized into a specific class that can, in turn, receive specific user-defined QoS treatment when that class is included in a policy map.

To configure a traffic policy, perform the following steps.

Note The **bandwidth** command is shown in Step 5. The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use. As of Cisco IOS Release 12.2(18)ZY, CBWFQ is not supported on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA).

Note For Cisco IOS Release 12.2(18)ZY, an existing traffic policy (policy map) cannot be modified if the traffic policy is already attached to the interface. To remove the policy map from the interface, use the **no** form of the **service-policy** command.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **bandwidth** {*bandwidth-kbps*| **remaining percent** *percentage*| **percent** *percentage*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map policy1`	Creates or modifies a policy map that can be attached to one or more interfaces and enters policy-map configuration mode. • Enter the name of the policy map.
Step 4	**class** {*class-name*	**class-default**} **Example:** `Device(config-pmap)# class class1`	Specifies the name of the class whose policy you want to create or change and enters policy-map class configuration mode. • Enter the specific class name or enter the **class-default**keyword.

	Command or Action	Purpose
Step 5	**bandwidth** {*bandwidth-kbps*| **remaining percent** *percentage*| **percent** *percentage*} **Example:** `Device(config-pmap-c)# bandwidth percent 50` **Example:**	(Optional) Specifies or modifies the bandwidth allocated for a class belonging to a policy map. • Enter the amount of bandwidth as a number of kbps, a relative percentage of bandwidth, or an absolute amount of bandwidth. **Note** The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use. **Note** As of Cisco IOS Release 12.2(18)ZY, CBWFQ is not supported on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA.
Step 6	**end** **Example:** `Device(config-pmap-c)# end`	(Optional) Returns to privileged EXEC mode.

Attaching a Traffic Policy to an Interface or Subinterface

After a policy map is created, the next step is to attach the traffic policy (sometimes called a policy map) to an interface or subinterface. Traffic policies can be attached to either the input or output direction of the interface or subinterface.

Note Depending on the needs of your network, you may need to attach the traffic policy to an ATM PVC, a Frame Relay data-link connection identifier (DLCI), or other type of interface.

To attach a traffic policy (policy map) to an interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **pvc** [*name*] *vpi* / *vci* [**ilmi**| **qsaal**| **smds**| **l2transport**]
5. **exit**
6. **service-policy** {**input** | **output**} *policy-map-name*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface ethernet 2/4`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**pvc** [*name*] *vpi* / *vci* [**ilmi**| **qsaal**| **smds**| **l2transport**] **Example:** `Device(config-if)# pvc cisco 0/16`	(Optional) Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode. • Enter the PVC name, the ATM network virtual path identifier, and the network virtual channel identifier. **Note** This step is required only if you are attaching the policy map to an ATM PVC. If you are not attaching the policy map to an ATM PVC, advance to Step 6.
Step 5	**exit** **Example:** `Device(config-atm-vc)# exit`	(Optional) Returns to interface configuration mode. **Note** This step is required only if you are attaching the policy map to an ATM PVC and you completed Step 4. If you are not attaching the policy map to an ATM PVC, advance to Step 6.
Step 6	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Device(config-if)# service-policy input policy1`	Attaches a policy map (traffic policy) to an input or output interface. • Specify either the **input** or **output** keyword, and enter the policy map name.

	Command or Action	Purpose

		Note Policy maps can be configured on ingress or egress Devices. They can also be attached in the input or output direction of an interface. The direction (input or output) and the Device (ingress or egress) to which the policy map should be attached vary according your network configuration. When using the **service-policy** command to attach the policy map to an interface, be sure to choose the Device and the interface direction that are appropriate for your network configuration.
		
		Note After you use the **service-policy** command, you may see two messages similar to the following:
		
		`%PISA-6-NBAR_ENABLED: feature accelerated on input direction of: [interface name and type ]`
		`%PISA-6-NBAR_ENABLED: feature accelerated on output direction of: [interface name and type`
		
		While both of these messages appear, NBAR is enabled in the direction specified by the **input** or **output** keyword only.

Step 7	**end** **Example:** `Device(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Verifying NBAR Using the MCQ

After you create the traffic classes and traffic policies (policy maps), you may want to verify that the end result is the one you intended. That is, you may want to verify whether your traffic is being classified correctly and whether it is receiving the QoS treatment as intended. You may also want to verify that the protocol-to-port mappings are correct.

To verify the NBAR traffic classes, traffic policies, and protocol-to-port mappings, perform the following steps.

SUMMARY STEPS

1. **show class-map** [*class-map-name*]
2. **show policy-map** [*policy-map*]
3. **show policy-map interface** *type number*
4. **show ip nbar port-map** [*protocol-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show class-map** [*class-map-name*] **Example:** `Device# show class-map`	(Optional) Displays all class maps and their matching criteria. • (Optional) Enter the name of a specific class map.
Step 2	**show policy-map** [*policy-map*] **Example:** `Device# show policy-map`	(Optional) Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps. • (Optional) Enter the name of a specific policy map.
Step 3	**show policy-map interface** *type number* **Example:** `Device# show policy-map interface Fastethernet 6/0`	(Optional) Displays the packet and class statistics for all policy maps on the specified interface. • Enter the interface type and the interface number.
Step 4	**show ip nbar port-map** [*protocol-name*] **Example:** `Device# show ip nbar port-map`	(Optional) Displays the current protocol-to-port mappings in use by NBAR. • (Optional) Enter a specific protocol name.

Verifying Unknown and Unclassified Traffic Management

To verify the management of unknown and unclassified traffic, perform the following steps.

SUMMARY STEPS

1. **show ip nbar protocol-id unknown**
2. **show ip nbar link-age unknown**
3. **show ip nbar protocol-attribute unknown**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**show ip nbar protocol-id unknown**
	Example:
	`Device# show ip nbar protocol-id unknown`
	`Protocol Name id type`
	`----------`
	`unknown 1 L7 STANDARD`
		(Optional) Displays protocol classification ID for unknown and unclassified traffic.

Step 2	**show ip nbar link-age unknown** **Example:**	(Optional) Displays the protocol link age for unknown and unclassified traffic.

	Command or Action	Purpose

	`Device# show ip nbar link-age unknown`
	
	`Protocol Link Age (seconds)`
	`unknown 60`	

Step 3	**show ip nbar protocol-attribute unknown**
	Example:
	
	`Device# show ip nbar protocol-attribute unknown`
	
	`Protocol Name : unknown`
	`encrypted : encrypted-no`
	`tunnel : tunnel-no`
	`category : other`
	`sub-category : other`
	`application-group : other`
	`p2p-technology : p2p-tech-no`
		(Optional) Displays list of configured attributes for unknown and unclassified traffic.

Configuration Examples for Configuring DSCP-Based Layer 3 Custom Applications

Example Configuring a Traffic Class

In the following example, a class called cmap1 has been configured. All traffic that matches the citrix protocol will be placed in the cmap1 class.

```
Device> enable

Device# configure terminal

Device(config)# class-map cmap1

Device(config-cmap)# match protocol citrix

Device(config-cmap)# end
```

Example Configuring a Traffic Policy

In the following example, a traffic policy (policy map) called policy1 has been configured. Policy1 contains a class called class1, within which CBWFQ has been enabled.

```
Device> enable

Device# configure terminal
```

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

Device(config-pmap)# class class1

Device(config-pmap-c)# bandwidth percent 50

Device(config-pmap-c)# end
```

Note In the above example, the **bandwidth** command is used to enable Class-Based Weighted Fair Queuing (CBWFQ). CBWFQ is only an example of one QoS feature that can be applied in a policy map. Use the appropriate command for the QoS feature that you want to use. As of Cisco IOS Release 12.2(18)ZY, CBWFQ is not supported on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA.

Example Attaching a Traffic Policy to an Interface or Subinterface

In the following example, the traffic policy (policy map) called policy1 has been attached to Ethernet interface 2/4 in the input direction of the interface.

```
Device> enable

Device# configure terminal

Device(config)# interface ethernet 2/4

Device(config-if)# service-policy input policy1

Device(config-if)# end
```

Example Verifying the NBAR Protocol-to-Port Mappings

The following is sample output of the **show ip nbar port-map** command. This command displays the current protocol-to-port mappings in use by NBAR. Use the display to verify that these mappings are correct.

```
Device# show ip nbar port-map
port-map bgp      udp 179
port-map bgp      tcp 179
port-map cuseeme  udp 7648 7649
port-map cuseeme  tcp 7648 7649
port-map dhcp     udp 67 68
port-map dhcp     tcp 67 68
```

If the **ip nbar port-map** command has been used, the **show ip nbar port-map** command displays the ports assigned to the protocol.

If the **no ip nbar port-map**command has been used, the **show ip nbar port-map** command displays the default ports. To limit the display to a specific protocol, use the *protocol-name* argument of the **show ip nbar port-map** command.

Example: L3 Custom any IP Port

```
Device> enable
Device# configuration terminal
Device (config)# ip nbar custom mycustom transport udp-tcp
Device(config-custom)# dscp ef
Device (config-custom)# exit
```

Where to Go Next

To add application recognition modules (also known as Packet Description Language Modules or PDLMs) to your network, see the "Adding Application Recognition Modules" module.

To classify network traffic on the basis of a custom protocol, see the "Creating a Custom Protocol" module.

Additional References

The following sections provide references related to configuring NBAR using the MQC.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features and functionality on the Catalyst 6500 series switch	"Configuring PFC QoS" chapter of the *Catalyst Supervisor Engine 32 PISA Cisco IOS Software Configuration Guide* , Release 12.2ZY
MQC, traffic policies (policy maps), and traffic classes	"Applying QoS Features Using the MQC" module
CBWFQ	"Configuring Weighted Fair Queueing" module
Concepts and information about NBAR	"Classifying Network Traffic Using NBAR" module
Information about enabling Protocol Discovery	"Enabling Protocol Discovery" module
Information about adding application recognition modules (also known as PDLMs)	"Adding Application Recognition Modules" module
Creating a custom protocol	"Creating a Custom Protocol" module

Technical Assistance

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

Feature Information for Configuring NBAR Using the MQC

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

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

Table 2: Feature Information for Configuring NBAR Using the MQC

Feature Name	Releases	Feature Information
NBAR MQC Support for Pre-resolved and Unknown Applications	IOS Release 15.5(1)T IOS XE Release 3.14S	The NBAR MQC Support for Pre-resolved and Unknown Applications feature provides support for matching all unknown and unclassified traffic using MQC. The following commands were modified: **class-map**, **match protocol**
QoS: DirectConnect PDLM	12.4(4)T	Provides support for the DirectConnect protocol and Packet Description Language Module (PDLM). The DirectConnect protocol can now be recognized when using the MQC to classify traffic. The following sections provide information about the QoS: DirectConnect PDLM feature:
QoS: Skype Classification	12.4(4)T	Provides support for the Skype protocol. The Skype protocol can now be recognized when using the MQC to classify traffic. **Note** Cisco currently supports Skype Version 1 only. The following sections provide information about the QoS: Skype Classification feature:

Feature Name	Releases	Feature Information
NBAR--BitTorrent PDLM	12.4(2)T	Provides support for the BitTorrent PDLM and protocol. The BitTorrent protocol can now be recognized when using the MQC to classify traffic. The following sections provide information about the NBAR-BitTorrent PDLM feature:
NBAR--Citrix ICA Published Applications	12.4(2)T	Enables NBAR to classify traffic on the basis of the Citrix Independent Computing Architecture (ICA) published application name and tag number. The following sections provide information about the NBAR-Citrix ICA Published Applications feature:
NBAR--Multiple Matches Per Port	12.4(2)T	Provides the ability for NBAR to distinguish between values of an attribute within the traffic stream of a particular application on a TCP or UDP port. The following sections provide information about the NBAR-Multiple Matches Per Port feature:
NBAR Extended Inspection for HTTP Traffic	12.3(4)T	Allows NBAR to scan TCP ports that are not well known and identify HTTP traffic that traverses these ports. The following sections provide information about the NBAR Extended Inspection for HTTP Traffic feature:
NBAR Real-Time Transport Protocol Payload Classification	12.2(15)T	Enables stateful identification of real-time audio and video traffic. The following section provides information about the NBAR Real-Time Transport Protocol Payload Classification feature:
NBAR--Network-Based Application Recognition	12.2(18)ZYA	Integrates NBAR and Firewall Service Module (FWSM) functionality on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA). Additional protocols are now recognized by NBAR. The following sections provide information about the NBAR feature: The following command was modified: **match protocol (NBAR)**.
NBAR--Network-Based Application Recognition (Hardware Accelerated NBAR)	12.2(18)ZY	Enables NBAR functionality on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA). The following section provides information about the NBAR--Network-Based Application Recognition (Hardware Accelerated NBAR) feature: