

MQC Based on Transport Hierarchy

The MQC Based on Transport Hierarchy(TPH) feature enables the use of TPH to apply policies according to a specific underlying protocol, instead of only according to the final classified protocol, for example, an email application over HTTP. A new MQC filter configured within a class-map matches all traffic which has this protocol in the hierarchy.

- Restrictions for MQC Based on Transport Hierarchy, on page 1
- Information About MQC Based on Transport Hierarchy, on page 1
- How to Configure MQC Based on Transport Hierarchy, on page 2
- Configuration Examples for MQC Based on Transport Hierarchy, on page 4
- Additional References, on page 5
- Feature Information for MQC Based on Transport Hierarchy, on page 5

Restrictions for MQC Based on Transport Hierarchy

- The MQC Based on Transport Hierarchy feature is supported only for DNS, HTTP, RTP, and SSL.
- Does not allow adding the match of the protocol and in-app-hierarchy to the same class-map.
- Match protocol http in-app-hierarchy and match protocol rtp in-app-hierarchy are not supported while match protocol attribute tunnel is configured, even on a different class-map.

Information About MQC Based on Transport Hierarchy

MQC Based on Transport Hierarchy Overview

The MQC based on transport hierarchy(TPH) feature enables NBAR to use TPH to apply policies according to a specific underlying protocol, instead of only according to the final classified protocol. The TPH of a particular application is the stack of protocols on which the application is delivered. For example, an application is being transported over HTTP and HTTP runs over TCP.

Prior to the configuartion of the MQC based on transport hierarchy(TPH) feature, it is only possible to apply a class-map filter on the final classified protocol using the **match protocol** *protocol-id* class-map filter. However, to apply QoS policies on all the traffic of HTTP, then include all the protocols which run over HTTP into the class-map makes the configuration of such use-cases considerably difficult. A solution for this problem

is an in-app-hierarchy class-map filter which uses TPH to apply policies according to a specific underlying protocol, instead of only according to the final classified protocol. For example, the rule **match protocol** *http* **in-app-hierarchy** matches if HTTP is present in the hierarchy.

How to Configure MQC Based on Transport Hierarchy

Configuring MQC Based on Transport Hierarchy

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-map-name*
4. **match protocol** *protocol-name* **in-app-hierarchy**
5. **end**
6. **configure terminal**
7. **policy-map** *policy-map-name*
8. **class** { *class-name* | **class-default** }
9. **end**
10. **configure terminal**
11. **interface** *type number*
12. **service-policy** { **input** | **output** } *policy-map-name*

DETAILED STEPS

Procedure

	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**] *class-map-name* **Example:** `Device(config)# class-map match-all C1`	Creates a class map to be used for matching packets to a specified class and enters QoS class-map mode. • Enter the name of the class map.

	Command or Action	Purpose
Step 4	**match protocol** *protocol-name* **in-app-hierarchy** **Example:** `Device(config-cmap)# match protocol http in-app-hierarchy`	Configures the match criterion for a class map on the basis of the specified protocol. The keyword **in-app-hierarchy** matches if the protocol is present in the transport hierarchy. Possible values for *protocol-name*: DNS, HTTP, RTP, SSL
Step 5	**end** **Example:** `Device(config-cmap)# end`	Exits class-map mode and returns to privileged EXEC mode.
Step 6	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 7	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map P1`	Specifies the name of the policy map and enters policy-map configuration mode.
Step 8	**class** { *class-name* | **class-default** } **Example:** `Device(config-pmap)# class C1`	Specifies the name of the class whose policy you want to create and enters policy-map class configuration mode.
Step 9	**end** **Example:** `Device(config-cmap)# end`	Exits class-map mode and returns to privileged EXEC mode.
Step 10	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/1`	Configures an interface type and enters interface configuration mode.
Step 12	**service-policy** { **input** | **output** } *policy-map-name* **Example:** `Device(config-if)# service-policy input P1`	Specifies the name of the policy map to be attached to the input or output direction of the interface.

Verifying MQC Based on Transport Hierarchy

To verify the MQC Based on Transport Hierarchy feature perform the following steps:

SUMMARY STEPS

1. **enable**

2. **show policy-map interface** *type number*
3. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map interface** *type number* **Example:** `Device# show policy-map interface GigabitEthernet0/0/1`	Displays the packet statistics of all classes that are configured for allservice policies either on the specified interface • Enter the interface type and the interface number.
Step 3	**exit** **Example:** `Device# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for MQC Based on Transport Hierarchy

Example: Configuring MQC Based on Transport Hierarchy

The following is an example of the configuring MQC based on Transport Hierarchy feature:

```
Device> enable
Device# configure terminal
Device(config)# class-map match-all C1
Device(config-cmap)# match protocol http in-app-hierarchy
Device(config-cmap)# match protocol youtube
Device(config-cmap)# end
Device# configure terminal
Device(config)# policy-map P1
Device(config-pmap)# class C1
Device(config-cmap)# end
Device# configure terminal
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# service-policy input P1
```

A traffic policy called P1 is configured. P1 contains a class called C1 for which QoS bandwidth limitation is configured as an example. All traffic that has final classification of Youtube with HTTP as a transport will be placed in the C1 class. Other possible transports for Youtube, such as DNS, SSL or RTSP, will not be matched by this class-map

Example: Verifying the MQC Based on Transport Hierarchy configuration

The following is a sample output from the **show policy-map interface** command:

```
Device#  show policy-map interface GigabitEthernet0/0/1

GigabitEthernet0/0/1
  Service-policy input: P1

Class-map: C1 (match-all)
      17 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: protocol http in-app-hierarchy
      Match: protocol youtube

    Class-map: class-default (match-any)
      3 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco IOS Quality of Service Solutions Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MQC Based on Transport Hierarchy

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

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

Table 1: Feature Information for MQC Based on Transport Hierarchy

Feature Name	Releases	Feature Information
MQC Based on Transport Hierarchy		The MQC Based on Transport Hierarchy feature enables the use of Transport Hierarchy to apply policies according to a specific underlying protocol, instead of only according to the final classified protocol. A new MQC filter is introduced which can be configured within a class-map. The following command was modified: **match protocol**
Transport Hierarchy support for DNS	Cisco IOS XE Denali 16.3	The **match protocol** CLI can match according to the following protocol types: DNS, HTTP, SSL, and RTP. Example: **match protocol dns in-app-hierarchy**