Configuring Classification

- About Classification, on page 1
- Prerequisites for Classification, on page 2
- Guidelines and Limitations for Classification, on page 2
- Configuring Traffic Classes, on page 5
- Verifying the Classification Configuration, on page 18
- Configuration Examples for Classification, on page 18

About Classification

Classification is the separation of packets into traffic classes. You configure the device to take a specific action on the specified classified traffic, such as policing or marking down, or other actions.

You can create class maps to represent each traffic class by matching packet characteristics with the classification criteria in the following table:

Table 1: Classification Criteria

Classification Criteria	Description
CoS	Class of service (CoS) field in the IEEE 802.1Q header.
IP precedence	Precedence value within the type of service (ToS) byte of the IP header.
Differentiated Services Code Point (DSCP)	DSCP value within the DiffServ field of the IP header.
ACL	IP, IPv6, or MAC ACL name.
Packet length	Size range of Layer 3 packet lengths.
IP RTP	Identify applications using Real-time Transport Protocol (RTP) by UDP port number range.

You can specify multiple match criteria, you can choose to not match on a particular criterion, or you can determine the traffic class by matching any or all criteria.

Note However, if you match on an ACL, no other match criteria, except the packet length, can be specified in a match-all class. In a match-any class, you can match on ACLs and any other match criteria.

Traffic that fails to match any class in a QoS policy map is assigned to a default class of traffic called class-default. The class-default can be referenced in a QoS policy map to select this unmatched traffic.

You can reuse class maps when defining the QoS policies for different interfaces that process the same types of traffic.

Prerequisites for Classification

Classification has the following prerequisites:

- You must be familiar with using modular QoS CLI.
- You are logged on to the device.

Guidelines and Limitations for Classification

Classification has the following configuration guidelines and limitations:

- QoS policy will not be effective for fragmented packets. Fragmented packets will be forwarded to the default queue.
- The **show** commands with the **internal** keyword are not supported.
- PVLANs do not provide support for PVLAN QoS.
- When the **destination interface sup-eth0** CLI command is configured, the following system log message is displayed: `Enabling span destination to SUP will affect ingress QoS classification.`
- For VXLAN, the following Cisco Nexus platforms support QoS policies for traffic in the network to host direction (decapsulation path) as egress policy on both the port and VLAN:
  - Cisco Nexus 9300 and 9500 platform switches.
  - Cisco Nexus 9200 and 9300-EX platform switches; Cisco Nexus 93180YC-EX and 93108TC-EX switches; and the Cisco Nexus 9732C-EX line card.
  - The preceding is not supported for the following hardware: Cisco Nexus 9230QC, 9272Q, 9232C, 9236C, and 92300YC switches; and Cisco Nexus 9160YC-X switches.
- For VXLAN, the following Cisco Nexus platforms do not support QoS policies for traffic from the network to access direction (decapsulation path) as ingress policy on the uplink interface:
  - Cisco Nexus 9300 and 9500 platform switches.
  - Cisco Nexus 9200 and 9300-EX platform switches; and Cisco Nexus 93180YC-EX and 93108TC-EX switches; and the Cisco Nexus 9732C-EX line card.

    - Cisco Nexus 9230QC, 9272Q, 9232C, 9236C, and 92300YC switches; and Cisco Nexus 9160YC-X switches.

- QoS classification is not supported on the FEX interfaces ingressing the VXLAN traffic. This limitation is applicable to all Cisco Nexus 9000 series switches.
- Matching the packets based on DSCP, CoS, or precedence in Cisco Nexus 9300-EX platform switches, the TCAM entries for both IPv4 (single-wide is one entry) and IPv6 (double-wide are two entries) are installed in the hardware. For example, if you match DSCP 4, three entries are installed in the hardware, one entry for IPv4 and two entries for IPv6.
- You can specify a maximum of 1024 match criteria in a class map.
- You can configure a maximum of 128 classes for use in a single policy map.
- When you match on an ACL, the only other match you can specify is the Layer 3 packet length in a match-all class.
- The **match-all** option in the **class-map type qos match-all** command is not supported. The match criteria of this command becomes the same as in the **class-map type qos match-any** command. The **class-map type qos match-all** command yields the same results as the **class-map type qos match-any** command.
- The **match-all** option is not supported in CoPP class-map and it always defaults to the **match-any** option.
- You can classify traffic on Layer 2 ports that are based on either the port policy or VLAN policy of the incoming packet but not both. If both are present, the device acts on the port policy and ignores the VLAN policy.
- When a Cisco Nexus Fabric Extender (FEX) is connected and in use, do not mark data traffic with a CoS value of 7. CoS 7 is reserved for control traffic transiting the Fabric Extender.
- Control traffic (control frames) from the switch to the FEX are marked with a CoS value of 7 and are limited to a jumbo MTU frame size of 2344 bytes.
- FEX QoS policy supports FEX host interfaces (HIF).
    - QoS TCAM carving is supported on ALE (Application Leaf Engine) enabled switches.
    - Only system level policies are supported.
    - Match on CoS is supported.
    - Match on QoS-group is supported.
- Jumbo ping (MTU of 2400 or greater) from a switch supervisor with a COS of 7, to a FEX host, fails because the control queue on a FEX supports an MTU limited to 2240.
- QoS classification policies are not supported under system QoS for Layer 2 switch ports. However, you can configure a QoS policy to classify the incoming traffic based on CoS/DSCP and map it to different queues. The QoS policy must be applied under all the interfaces that require the classification.
- A QoS policy with a MAC-based ACL as a match in the class map does not work for IPv6 traffic. For QoS, IPv6 traffic must be matched based on IPv6 addresses and not on MAC addresses.
- As a best practice, avoid having a voice VLAN configuration where an access VLAN is same as the voice VLAN.

  The following are alternative approaches:

- If a separate dot1p tag (cos) value is not required for voice traffic, use the **switchport voice vlan untagged** command.

```
switch(config)# interface ethernet 1/1
switch(config-if)# switchport access vlan 20
switch(config-if)# switchport voice vlan untagged
```

- If a separate cos value is required for voice traffic, use the **switchport voice vlan dot1p** command.

```
switch(config)# interface ethernet 1/1
switch(config-if)# switchport access vlan 20
switch(config-if)# switchport voice vlan dot1p
```

- Cisco Nexus 9504 and Cisco Nexus 9508 switches with the following line cards do not support QoS match acl with fragments:
  - Cisco Nexus 96136YC-R
  - Cisco Nexus 9636C-RX
  - Cisco Nexus 9636Q-R
  - Cisco Nexus 9636C-R
- MPLS packets with a NULL label on transit nodes, receive an MPLS classification that is based on its NULL label EXP.
- Ingress DROP_ACL_DROP is seen with Cisco Nexus 9272Q, 9236C, and 92160YC-X switches on an ASIC during congestion. However, these drops do not impact the performance of the switch.
- A QoS policy that references an ACL that contains a match for ICMP type or code is not supported.
- A QoS Policy that references an ACL that contains a match for TCP flags is only supported on the following Cisco Nexus 9000 series switches:
  - Cisco Nexus 9200 platform switches
  - Cisco Nexus 9300-EX platform switches
  - Cisco Nexus 9300-FX platform switches
  - Cisco Nexus 9300-GX platform switches
  - Cisco Nexus 9500 platform switches with Cisco Nexus 97xx-EX and 97xx-FX line cards
  - Beginning with Cisco NX-OS Release 10.2(1q)F, QoS Classification is supported on the N9K-C9332D-GX2B platform switches.

Configuring Traffic Classes

Configuring ACL Classification

You can classify traffic by matching packets based on an existing access control list (ACL). Traffic is classified by the criteria defined in the ACL. The permit and deny ACL keywords are ignored in the matching; even though a match criteria in the access-list has a deny action, it is still used for matching for this class.

Note Use the **class-map class_acl** command to display the ACL class-map configuration.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match access-group name** *acl-name*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**class-map** [**type qos**] [**match-any**	**match-all**] *class-name* **Example:** `switch(config)# class-map class_acl`	Creates or accesses the class map named class-name and enters class-map mode. The class map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters. (**match-any** is the default when no option is selected and multiple match statements are entered.)
Step 3	**match access-group name** *acl-name* **Example:** `switch(config-cmap-qos)# match access-group name my_acl`	Configures the traffic class by matching packets based on the *acl-name*. The **permit** and **deny** ACL keywords are ignored in the matching.

Examples: Configuring ACL Classification

To prevent packets from being matched by the QoS class-map, you must explicitly specify the packets you want to match with permit statements. The *implicit* default deny statement at the end of the ACL will filter out the remainder. Any *explicit* deny statements configured inside the access list of a QoS class map will be ignored in the matching and treated as an explicit permit statement as shown in the examples below.

The following examples, A1, B1, and C1, all produce the same QoS matching results:

- A1

```
ip access-list extended A1
  permit ip 10.1.0.0 0.0.255.255 any
  permit ip 172.16.128.0 0.0.1.255 any
  permit ip 192.168.17.0 0.0.0.255 any
```

- B1

```
ip access-list extended B1
  permit ip 10.1.0.0 0.0.255.255 any
  deny ip 172.16.128.0 0.0.1.255 any  /* deny is interpreted as a permit */
  permit ip 192.168.17.0 0.0.0.255 any
```

- C1

```
ip access-list extended C1
  deny ip 10.1.0.0 0.0.255.255 any   /* deny is interpreted as a permit */
 deny ip 172.16.128.0 0.0.1.255 any  /* deny is interpreted as a permit */
 deny ip 192.168.17.0 0.0.0.255 any  /* deny is interpreted as a permit */
```

Adding an explicit DENY ALL at the end of a QoS matching ACL causes the QoS ACL to permit all traffic.

The following examples, D1 and E1, produce the same QoS matching results:

- D1

```
ip access-list extended D1
  permit ip 10.1.0.0 0.0.255.255 any
  permit ip 172.16.128.0 0.0.1.255 any
  permit ip 192.168.17.0 0.0.0.255 any
 deny ip 0.0.0.0 255.255.255.255 any  /* deny is interpreted as a permit */
```

Note The last line in the example effectively becomes a PERMIT ALL statement and results in the QoS ACL to permit all packets.

- E1

```
ip access-list extended E1
    permit ip 0.0.0.0 255.255.255.255 any
```

Configuring a DSCP Wildcard Mask

Use the DSCP wildcard mask feature to classify multiple DSCP values from a set of IP flows recognized by an ACL and the DSCP value. Classification of IP information and DSCP values occurs in a more granular way by using multiple parameters. With this granularity, you can treat these flows by policing them to protest the rest of the traffic, or assign them to a qos-group for further QoS operations.

Note Only Cisco Nexus 9300-EX/FX/FX2/FX3 platform switches support the DSCP wildcard mask feature.

SUMMARY STEPS

1. **configure terminal**
2. **ip access-list** *acl-name*
3. [*sequence-number*] { **permit** | **deny** } *protocol* { *source-ip-prefix* | *source-ip-mask* } { *destination-ip-prefix* | *destination-ip-mask* } [**dscp** *dscp-value* **dscp-mask** *0-63*]
4. [*sequence-number*] { **permit** | **deny** } *protocol* { *source-ip-prefix* | *source-ip-mask* } { *destination-ip-prefix* | *destination-ip-mask* } [**dscp** *dscp-value* [*dscp-mask*]]
5. **exit**
6. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
7. **match access-list** *acl-name*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**ip access-list** *acl-name*
	Example:
	`switch(config)# ip access-list acl-01`
	`switch(config-acl)`
		Enters the ACL configuration mode and creates an ACL with the entered name.

Step 3	[*sequence-number*] { **permit** | **deny** } *protocol* { *source-ip-prefix* | *source-ip-mask* } { *destination-ip-prefix* | *destination-ip-mask* } [**dscp** *dscp-value* **dscp-mask** *0-63*] **Example:** `switch(config-acl)# 10 permit ip 10.1.1.1/24 20.1.1.2/24 dscp 33 dscp-mask 33`	Creates an ACL entry that matches or filters traffic that is based on a DSCP wildcard bit mask. The *sequence-number* argument can be a whole number from 1 through 4294967295. **dscp** *dscp-value*: Match packets with a specific DSCP value. **dscp-mask** *dscp-mask-value*: Configures the DSCP wildcard mask which matches on any bit in the DSCP value to filter traffic. Range is from 0 to 0x3F.
Step 4	[*sequence-number*] { **permit** | **deny** } *protocol* { *source-ip-prefix* | *source-ip-mask* } { *destination-ip-prefix* | *destination-ip-mask* } [**dscp** *dscp-value* [*dscp-mask*]] **Example:** `switch(config-acl)# 10 permit ip 10.1.1.1/24 20.1.1.2/24 dscp 33 30`	Creates an ACL entry that matches or filters traffic that is based on a DSCP wildcard bit mask. The *sequence-number* argument can be a whole number from 1 through 4294967295. **dscp**: Match packets with a specific DSCP value. *dscp-mask*: Configures the DSCP wildcard mask which matches on any bit in the DSCP value to filter traffic. Range is from 0 to 0x3F.

Step 5	**exit**
	Example:
	`switch(config-acl)# exit`
	`switch(config)#`
		Exits ACL configuration mode and enters global configuration mode.

	Command or Action	Purpose

Step 6	**class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
	Example:
	`switch(config)# class-map type qos match-any class_dscp_mask`
	`switch(config-cmap-qos)#`
		Creates or accesses the class map that is named by the *class-name* variable and enters the class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.

Step 7	**match access-list** *acl-name*
	Example:
	`switch(config-cmap-qos)# match access-list acl-01`
	`switch(config-cmap-qos)#`
		Configures the traffic class by matching packets that are based on the IP access list.

Example

In the following example, an ACL looks at traffic that is sent from subnet 10.1.1.0 to subnet 20.1.1.0. The ACL also checks for traffic with DSCP 33, and any subsequent DSCP values from 33 through 63, with a mask value of 30. The ACL is set to a class map that is matching this ACL for further QoS operations.

```
switch# configure terminal
switch(config)# ip access-list acl-01
switch(config-acl)# 10 permit ip 10.1.1.1/24 20.1.1.2/24 dscp 33 dscp-mask 30
switch(config-acl)# exit
switch(config)# class-map type qos match-any class_dscp_mask
switch(config-cmap-qos)# match access-list acl-01
```

Configuring DSCP Classification

You can classify traffic based on the DSCP value in the DiffServ field of the IP header. The standard DSCP values are listed in the following table:

Table 2: Standard DSCP Values

Value	List of DSCP Values
af11	AF11 dscp (001010)—decimal value 10
af12	AF12 dscp (001100)—decimal value 12
af13	AF13 dscp (001110)—decimal value 14
af21	AF21 dscp (010010)—decimal value 18
af22	AF22 dscp (010100)—decimal value 20
af23	AF23 dscp (010110)—decimal value 22
af31	AF31 dscp (011010)—decimal value 26
af32	AF40 dscp (011100)—decimal value 28

Value	List of DSCP Values
af33	AF33 dscp (011110)—decimal value 30
af41	AF41 dscp (100010)—decimal value 34
af42	AF42 dscp (100100)—decimal value 36
af43	AF43 dscp (100110)—decimal value 38
cs1	CS1 (precedence 1) dscp (001000)—decimal value 8
cs2	CS2 (precedence 2) dscp (010000)—decimal value 16
cs3	CS3 (precedence 3) dscp (011000)—decimal value 24
cs4	CS4 (precedence 4) dscp (100000)—decimal value 32
cs5	CS5 (precedence 5) dscp (101000)—decimal value 40
cs6	CS6 (precedence 6) dscp (110000)—decimal value 48
cs7	CS7 (precedence 7) dscp (111000)—decimal value 56
default	Default dscp (000000)—decimal value 0
ef	EF dscp (101110)—decimal value 46

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **dscp** *dscp-values*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**class-map** [**type qos**] [**match-any**	**match-all**] *class-name* **Example:** `switch(config)# class-map class_dscp`	Creates or accesses the class map named class-name and enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **dscp** *dscp-values* **Example:** `switch(config-cmap-qos)# match dscp af21, af32`	Configures the traffic class by matching packets based on dscp-values. The standard DSCP values are shown in the following table.

	Command or Action	Purpose
		Use the **not** keyword to match on values that do not match the specified range.

Step 4	**exit**
	Example:
	`switch(config-cmap-qos)# exit`
	`switch(config)#`
		Exits global class-map queuing mode and enters global configuration mode.

Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the DSCP class-map configuration:

```
switch# show class-map class_dscp
```

Configuring IP Precedence Classification

You can classify traffic based on the precedence value in the type of service (ToS) byte field of the IP header. The precedence values are listed in the following:

Table 3: Precedence Values

Value	List of Precedence Values
0-7	IP precedence value
critical	Critical precedence (5)
flash	Flash precedence (3)
flash-override	Flash override precedence (4)
immediate	Immediate precedence (2)
internet	Internetwork control precedence (6)
network	Network control precedence (7)
priority	Priority precedence (1)
routine	Routine precedence (0)

SUMMARY STEPS

1. **configure terminal**

2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **precedence** *precedence-values*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**class-map** [**type qos**] [**match-any**	**match-all**] *class-name* **Example:** `switch(config)# class-map class_ip_precedence`	Creates or accesses the class map named class-name and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **precedence** *precedence-values* **Example:** `switch(config-cmap-qos)# match precedence 1-2, 5-7`	Configures the traffic class by matching packets based on *precedence-values*. Values are shown in the following table. Use the **not** keyword to match on values that do not match the specified range.

Step 4	**exit**
	Example:
	`switch(config-cmap-qos)# exit`
	`switch(config)#`
		Exits global class-map queuing mode and enters global configuration mode.

Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the IP precedence class-map configuration:

```
switch# show class-map class_ip_precedence
```

Configuring Protocol Classification

For Layer 3 protocol traffic, you can use the ACL classification match.

Table 4: match Command Protocol Arguments

Argument	Description
arp	Address Resolution Protocol (ARP)

Argument	Description
bridging	Bridging
cdp	Cisco Discovery Protocol (CDP)
dhcp	Dynamic Host Configuration (DHCP)
isis	Intermediate system to intermediate system (IS-IS)

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **protocol** {**arp** | **bridging** | **cdp** | **dhcp** | **isis**}
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**class-map** [**type qos**] [**match-any**	**match-all**] *class-name* **Example:** `switch(config)# class-map class_protocol`	Creates or accesses the class map named class-name and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **protocol** {**arp**	**bridging**	**cdp**	**dhcp**	**isis**} **Example:** `switch(config-cmap-qos)# match protocol isis`	Configures the traffic class by matching packets based on the specified protocol. Use the **not** keyword to match on protocols that do not match the protocol specified.

Step 4	**exit**
	Example:
	`switch(config-cmap-qos)# exit`
	`switch(config)#`
		Exits global class-map queuing mode and enters global configuration mode.

Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the protocol class-map configuration:

```
switch# show class-map class_protocol
```

Configuring Layer 3 Packet Length Classification

You can classify Layer 3 traffic based on various packet lengths.

Note This feature is designed for IP packets only.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **packet length** *packet-length-list*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**class-map** [**type qos**] [**match-any**	**match-all**] *class-name* **Example:** `switch(config)# class-map class_packet_length`	Creates or accesses the class map named class-name and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **packet length** *packet-length-list* **Example:** `switch(config-cmap-qos)# match packet length min 2000`	Configures the traffic class by matching packets based on various packet lengths (bytes). Values can range from 1 to 9198. Use the **not** keyword to match on values that do not match the specified range.

Step 4	**exit**
	Example:
	`switch(config-cmap-qos)# exit`
	`switch(config)#`
		Exits global class-map queuing mode and enters global configuration mode.

Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the packet length class-map configuration:

```
switch# show class-map class_packet_length
```

Configuring CoS Classification

You can classify traffic based on the class of service (CoS) in the IEEE 802.1Q header. This 3-bit field is defined in IEEE 802.1p to support QoS traffic classes. CoS is encoded in the high order 3 bits of the VLAN ID Tag field and is referred to as user_priority.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **cos** *cos-list*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-name* **Example:** `switch(config)# class-map class_cos`	Creates or accesses the class map named class-name and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **cos** *cos-list* **Example:** `switch(config-cmap-qos)# match cos 4,5-6`	Configures the traffic class by matching packets based on the list of CoS values. Values can range from 0 to 7. Use the **not** keyword to match on values that do not match the specified range. **Note** When a Cisco Nexus Fabric Extender (FEX) is connected and in use, data traffic should not be marked with a CoS value of 7. CoS 7 is reserved for control traffic transiting the Fabric Extender.

Step 4	**exit**
	Example:
	`switch(config-cmap-qos)# exit`
	`switch(config)#`
		Exits global class-map queuing mode and enters global configuration mode.

	Command or Action	Purpose
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the CoS class-map configuration:

```
switch# show class-map class_cos
```

Configuring CoS Classification for FEX

Note The CoS Classification for FEX feature is not supported on the Cisco Nexus 9508 switch (NX-OS 7.0(3)F3(3)).

You can classify traffic based on the class of service (CoS) for a FEX.

Before you begin

Before configuring the FEX, enable **feature-set fex**.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **cos** *cos-list*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**class-map** [**type qos**] [**match-any**	**match-all**] *class-name* **Example:** `switch(config)# class-map class_cos`	Creates or accesses the class map named class-name and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **cos** *cos-list* **Example:**	Configures the traffic class by matching packets based on the list of CoS values. Values can range from 0 to 7. Use

	Command or Action	Purpose
	`switch(config-cmap-qos)# match cos 4,5-6`	the **not** keyword to match on values that do not match the specified range. **Note** When a Cisco Nexus Fabric Extender (FEX) is connected and in use, data traffic should not be marked with a CoS value of 7. CoS 7 is reserved for control traffic transiting the Fabric Extender.

Step 4	**exit**
	Example:
	`switch(config-cmap-qos)# exit`
	`switch(config)#`
		Exits global class-map queuing mode and enters global configuration mode.

Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to configure the CoS class-map configuration:

```
switch# conf t
switch(config)# class-map type qos match-all cos6
switch(config-cmap-qos)# match cos 6
switch(config)# class-map type qos match-all cos1
switch(config-cmap-qos)# match cos 1
switch(config)# class-map type qos match-all cos2
switch(config-cmap-qos)# match cos 2
switch(config)# class-map type qos match-all cos3
switch(config-cmap-qos)# match cos 3
switch(config)# class-map type qos match-all cos0
switch(config-cmap-qos)# match cos 0
```

Configuring IP RTP Classification

The IP Real-Time Transport Protocol (RTP) is a transport protocol for real-time applications that transmit data such as audio or video (RFC 3550). Although RTP does not use a common TCP or UDP port, you typically configure RTP to use ports 16384 to 32767. UDP communications uses an even-numbered port and the next higher odd-numbered port is used for RTP Control Protocol (RTCP) communications.

Cisco Nexus 9000 Series switches support the transport of RDMA over Converged Ethernet (RoCE) v1 and v2 protocols. RoCE uses a UDP port.

When defining a match statement in a **type qos class-map**, to match with upper layer protocols and port ranges (UDP/TCP/RTP, among others), the system cannot differentiate, for example, between UDP traffic and RTP traffic in the same port range. The system classifies both traffic types the same. For better results, you must engineer the QoS configurations to match the traffic types present in the environment.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **ip rtp** *udp-port-value*
4. **match** [**not**] **ip roce** *udp-port-value*
5. **exit**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-name* **Example:** `switch(config)# class-map class_rtp`	Creates or accesses a class map and then enters the class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **ip rtp** *udp-port-value* **Example:** `switch(config-cmap-qos)# match ip rtp 2000-2100, 4000-4100`	Configures the traffic class by matching packets that are based on a range of lower and upper UDP port numbers, targeting applications using RTP. Values can range from 2000 to 65535. Use the **not** keyword to match on values that do not match the specified range.
Step 4	**match** [**not**] **ip roce** *udp-port-value* **Example:** `switch(config-cmap-qos)# match ip roce 3000-3100, 6000-6100`	Configures the traffic class by matching packets that are based on a range of lower and upper UDP port numbers, targeting applications using RoCE. Values can range from 2000 to 65535. Use the **not** keyword to match on values that do not match the specified range. **Note** If ip roce and ip rtp are configured to match with the same port number, only ip rtp is displayed when you use the **show policy-map interface** *interface-type***type qos** command.. When you use the help string for both the RTP and RoCE, the recommended range is displayed but you are allowed to specify the value outside the recommended range as well (based on your requirement).

Step 5	**exit**
	Example:
	`switch(config-cmap-qos)# exit`
	`switch(config)#`
		Exits global class-map queuing mode and enters global configuration mode.

	Command or Action	Purpose
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the RTP class-map configuration:

```
switch# show class-map class_rtp
```

Verifying the Classification Configuration

Use the **show class-map** command to verify the class-map configuration. This command displays all class maps.

Configuration Examples for Classification

The following example shows how to configure classification for two classes of traffic:

```
class-map class_dscp
match dscp af21, af32
exit
class-map class_cos
match cos 4, 5-6
exit
```