Configuring Cisco Performance Monitor

This document contains information about and instructions for configuring Cisco Performance Monitor.

Information About Cisco Performance Monitor

Overview of Cisco Performance Monitor

Cisco Performance Monitor enables you to monitor the flow of packets in your network and become aware of any issues that might impact the flow before it starts to significantly impact the performance of the application in question. Performance monitoring is especially important for video traffic because high quality interactive video traffic is highly sensitive to network issues. Even minor issues that may not affect other applications can have dramatic effects on video quality.

Because Cisco Performance Monitor uses similar software components and commands as Cisco NetFlow and Cisco Flexible NetFlow, familiarity with these products will help you to understand how to configure Cisco Performance Monitor. These products provide statistics on packets flowing through a router and are the standard for acquiring IP operational data from IP networks. They provide data to support network and security monitoring, network planning, traffic analysis, and IP accounting. For more information about Cisco NetFlow and Cisco Flexible NetFlow, see the documents listed in the Additional References section.

For more information about the design, configuration, and troubleshooting of Performance Monitor and other Cisco Medianet products, including a Quick Start Guide and Deployment Guide, see the Cisco Medianet Knowledge Base Portal, located at http://www.cisco.com/web/solutions/medianet/knowledgebase/index.html.

Prerequisites for Configuring Cisco Performance Monitor

The following prerequisites must be met before you can configure Cisco Performance Monitor:

IPv4 Traffic

- The networking device must be configured for IPv4 routing.
- One of the following must be enabled on your router and on any interfaces on which you want to enable Cisco Performance Monitor: Cisco Express Forwarding or distributed Cisco Express Forwarding.

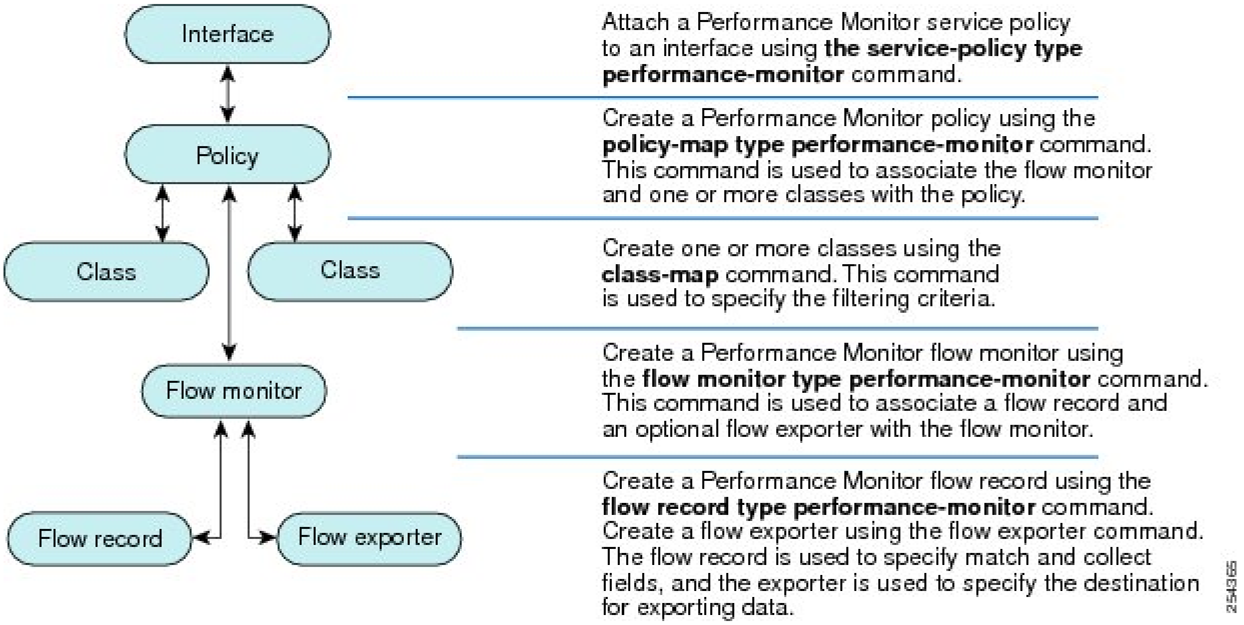
Configuration Components of Cisco Performance Monitor

To configure Cisco Performance Monitor, configure many of the same basic elements that you normally configure for Flexible NetFlow:

- Interface
- Policy
- Class
- Flow monitor
- Flow record
- Flow exporter

The figure below shows how these elements are related to each other. The elements at the bottom of the figure are configured first.

Figure 1: Cisco Performance Monitor Components



Attach a Performance Monitor service policy to an interface using **the service-policy type performance-monitor** command.

Create a Performance Monitor policy using the **policy-map type performance-monitor** command. This command is used to associate the flow monitor and one or more classes with the policy.

Create one or more classes using the **class-map** command. This command is used to specify the filtering criteria.

Create a Performance Monitor flow monitor using the **flow monitor type performance-monitor** command. This command is used to associate a flow record and an optional flow exporter with the flow monitor.

Create a Performance Monitor flow record using the **flow record type performance-monitor** command. Create a flow exporter using the flow exporter command. The flow record is used to specify match and collect fields, and the exporter is used to specify the destination for exporting data.

As shown above, a policy includes one or more classes. Each class has a flow monitor associated with it, and each flow monitor has a flow record and an optional flow exporter associated with it. These elements are configured in the following order:

1. Configure a flow record to specify the key and non-key fields that you want to monitor. This is configured using **match** and **collect** commands. You can also optimally configure a flow exporter to specify the

export destination. For Cisco Performance Monitor, you must configure a **performance-monitor** type flow record.

2. Configure a flow monitor that includes the flow record and flow exporter. For Cisco Performance Monitor, you must configure a **performance-monitor** type flow monitor.
3. Configure a class to specify the filtering criteria using the **class-map** command.
4. Configure a policy to include one or more classes and one or more **performance-monitor** type flow monitors using the **policy-map** command. For Cisco Performance Monitor, you must configure **performance-monitor** type policies.
5. Associate a **performance-monitor** type policy to the appropriate interface using the **service-policy type performance-monitor** command.

Data That You Can Monitor Using Cisco Performance Monitor

You can monitor the following information by configuring a flow record with **collect** or **match** commands for the corresponding non-key fields:

Tip For more information about these statistics, see the **show performance monitor status**command in the*Cisco Media Monitoring Command Reference.*

- IP Packet Count
- IP TTL
- IP TTL minimum
- IP TTL maximum
- Flow to Interface Mapping
- IP Flow destination address and port, source address and port, and protocol
- RTP Synchronization Source (SSRC)
- IP Octets Count
- Media Stream Packet Count
- Media Stream Octect Count
- Media Byte Rate
- Media Byte Count
- Media Packet Rate
- Media Packet Loss Count
- Media Packet Loss Rate
- Packets Expected Count
- Measured Rate

- Media Loss Event Count
- Round Trip Time (RTT)
- Interarrival Jitter (RFC3550) max
- Interarrival Jitter (RFC3550) min 2
- Interarrival Jitter (RFC3550) mean
- Media Rate Variation
- Monitor Event
- Media Error
- Media Stop
- IP Byte Count
- IP Byte Rate
- IP Source Mask
- IP Destination Mask
- Epoch of A Monitoring Interval
- Packet Forwarding Status
- Packet Drops
- DSCP and IPv6 Traffic Class
- TCP: Maximum Segment Size
- TCP: Window Size Maximum
- TCP: Window Size Maximum
- TCP: Window Size Average
- Out Of Order Bytes
- Out Of Order Packets

SNMP MIB Support for Cisco Performance Monitor

Cisco Performance Monitor provides support for the use of the industry-standard Simple Network Management Protocol (SNMP) to monitor media streams. This support is implemented with the addition of the following Cisco proprietary SNMP Management Information Base (MIB) modules:

- CISCO-FLOW-MONITOR-TC-MIB—Defines the textual conventions common to the following MIB modules.
- CISCO-FLOW-MONITOR-MIB—Defines the framework that describes the flow monitors supported by a system, the flows that it has learned, and the flow metrics collected for those flows.
- CISCO-RTP-METRICS-MIB—Defines objects that describe the quality metrics collected for RTP streams, similar to those described by an RTCP Receiver Report packet (RFC 3550).

- CISCO-IP-CBR-METRICS-MIB—Defines objects that describe the quality metrics collected for IP streams that have a Constant Bit Rate (CBR).

For detailed information about these MIBs, and to locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at http://www.cisco.com/go/mibs .

This feature also includes two new command-line interface (CLI) commands and one modified CLI command. The commands are as follows:

- **snmp-server host**—Enables the delivery of flow monitoring SNMP notifications to a recipient.
- **snmp-server enable traps flowmon**—Enables flow monitoring SNMP notifications. By default, flow monitoring SNMP notifications are disabled.
- **snmp mib flowmon alarm history**—Sets the maximum number of entries maintained by the flow monitor alarm history log.

Limitations for the Catalyst 6500 Platform

Cisco Performance Monitor has the following limitations on the Catalyst 6000 platform:

- There are some limitations on which types of interfaces can be monitored. The next two tables list which types of interfaces are supported for ingress and egress monitoring on the Catalyst 6500 platform.

Table 1: Support for Ingress Interfaces

Interface Type	Support
Layer 3 Routed Port	Yes
Layer 3 Sub-interface (a)	No
Layer 3 port channels	Yes
Layer 3 port-channel sub-interface (a)	No
Layer 3 SVI (b)	Partial (see the third bullet below)
L3 Tunnels	No
Layer 2 Physical (Switched) Ports	Yes
Layer 2 Port-channels	Yes
Layer 2 Vlans	Yes

Table 2: Support for Egress Interfaces

Interface Type	Support
Layer 3 Routed Port	Yes
Layer 3 Sub-interface (a)	Yes

Interface Type	Support
Layer 3 port channels	Yes
Layer 3 port-channel sub-interface (a)	Yes
Layer 3 SVI (b)	Yes
L3 Tunnels	No
Layer 2 Physical (Switched) Ports	No
Layer 2 Port-channels	No
Layer 2 Vlans	Yes

- Performance monitoring on VRFs is not supported.
- Performance Monitoring of multicast flows is not supported.
- Routed traffic from a trunk port on a VLAN interface cannot not be monitored because it is not possible to identify the source VLAN interface for the traffic. You will see the following syslog message: “Routed traffic from trunk ports will not be monitored by ingress policy on VLAN interface.”

  For a workaround, you can configure a performance monitoring policy on a trunk interface. This monitoring will result in additional CPU usage.
- You cannot use match all type Class maps. Only match any type of lookups are supported. If you configure performance monitoring to use match-all type class maps, it will result in the cloning of packet to the CPU. Packets will then again be classified in the CPU when match-all classes are properly applied and packet are dropped if required. This causes higher than expected CPU usage.
- Performance monitoring policy on the egress of a VLAN interface will not monitor traffic getting bridged within the VLAN. This is due to hardware limitation. Workaround is to apply the policy at the ingress of VLAN interface as well as egress. Policy on the ingress of the VLAN interface will monitor bridged packets.
- Cloned packets from Egress policies can only be software rate-limited. No hardware-based protection is available for these packets. Therefore, you might see high interrupt CPU usage during scenarios when many flows are being monitored.
- Egress performance monitoring makes use of a recirculation mechanism on the Catalyst 6500 platform. This introduces several microseconds of additional latency to the frame switching.
- Performance monitoring is not supported for the packets switched using the Fast (CEF) Path.
- Lawful intercept and performance monitoring makes use of the same mechanism for cloning the packets. The Lawful Intercept feature takes precedence over performance monitoring. Therefore, performance monitoring does not function when the Lawful Intercept feature is enabled. When this occurs, a syslog message is created.
- Performance monitoring makes use of same mechanism as other features, such as Optimized ACL logging, VACL Capture, IPv6 Copy, and so on. The feature that is enabled first takes precedence. The other features are blocked from being configured and a syslog message is created.

Limitations for IPv6 Support

Support for IPv6 with Performance Monitor has the following limitations:

- The following topologies are supported with IPv6: Non-MPLS, DMVPN (on most platforms), and dual stack.
- The following topologies are not supported with IPv6: MPLS/VRF (6PE and 6VPE), GETVPN and IPV6 over IPV4 tunnel.
- Mediatrace does not support IPv6.
- Exporting data to a IPv6 address is not supported on the ASR1K platform.
- Flexible NetFlow does not support IPv6 multicast.
- DMVPN is not supported with IPv6 on the ASR1K platform.

How to Configure Troubleshoot and Maintain Cisco Performance Monitor

Note Many of the Flexible NetFlow commands, keywords, and arguments used in used in these tasks are available in previous releases. For more information about these existing Flexible NetFlow commands, keywords, and arguments, refer to the *Cisco IOS Flexible NetFlow Command Reference*.

Configuring a Flow Exporter for Cisco Performance Monitor

Flow exporters are used to send the data that you collect with Cisco Performance Monitor to a remote system such as a NetFlow Collection Engine. Flow exporters use user datagram protocol (UDP) as the transport protocol and use the Version 9 export format.

To configure a flow exporter for the flow monitor, in order to export the data that is collected by Cisco Performance Monitor to a remote system for further analysis and storage, perform the following optional task. For Cisco Performance Monitor, flow exporters are configured the same way as they are configured for Cisco IOS Flexible NetFlow. For more information. see *Configuring Data Export for Cisco IOS Flexible NetFlow with Flow Exporters*.

Note You can export to a destination using either an IPv4 or IPv6 address.

Note Each flow exporter supports only one destination. If you want to export the data to multiple destinations, you must configure multiple flow exporters and assign them to the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **description** *description*
5. **destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*]
6. **export-protocol** {**netflow-v5** | **netflow-v9** | **ipfix** }
7. **dscp** *dscp*
8. **source** *interface-type interface-number*
9. **option** {**application-attributes** | **application table** | **exporter-stats** | **interface-table** | **metadata-table** | **sampler-table** | **vrf-table**} [**timeout** *seconds*]
10. **output-features**
11. **template data timeout** *seconds*
12. **transport udp** *udp-port*
13. **ttl** *seconds*
14. **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	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates the flow exporter and enters Flexible NetFlow flow exporter configuration mode. • This command also allows you to modify an existing flow exporter.
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description Exports to the datacenter`	(Optional) Configures a description to the exporter that will appear in the configuration and the display of the **show flow exporter** command.
Step 5	**destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the IP address or hostname of the system to which the exporter sends data. **Note** You can export to a destination using either an IPv4 or IPv6 address.

	Command or Action	Purpose

Step 6	**export-protocol** {**netflow-v5** | **netflow-v9** | **ipfix** }
	Example:
	`Device(config-flow-exporter)# export-protocol netflow-v9`
		Specifies the protocol used by the exporter.
		Note The export of extracted fields from NBAR is only supported over IPFIX.

Step 7	**dscp** *dscp* **Example:** `Device(config-flow-exporter)# dscp 63`	(Optional) Configures differentiated services code point (DSCP) parameters for datagrams sent by the exporter. • The range for the *dscp* argument is from 0 to 63. Default: 0.
Step 8	**source** *interface-type interface-number* **Example:** `Device(config-flow-exporter)# source ethernet 0/0`	(Optional) Specifies the local interface from which the exporter will use the IP address as the source IP address for exported datagrams.
Step 9	**option** {**application-attributes** | **application table** | **exporter-stats** | **interface-table** | **metadata-table** | **sampler-table** | **vrf-table**} [**timeout** *seconds*] **Example:** `Device(config-flow-exporter)# option exporter-stats timeout 120`	(Optional) Enables the use of option tables to decrease the amount of data exported. These tables allow the exporter to just export an ID that represents the complete value of the metadata and is mapped to the value by the option table. For example, the interface table maps the SNMP index to the interface name and the VRF table maps the VRF ID to the name. • You can enable the use of any combination of option tables concurrently. • The range for the *seconds* argument is 1 to 86,400. Default: 600.
Step 10	**output-features** **Example:** `Device(config-flow-exporter)# output-features`	(Optional) Enables sending export packets using quality of service (QoS) and encryption.
Step 11	**template data timeout** *seconds* **Example:** `Device(config-flow-exporter)# template data timeout 120`	(Optional) Configure the resending of templates based on a timeout. • The range for the *seconds* argument is 1 to 86400 (86400 seconds = 24 hours).
Step 12	**transport udp** *udp-port* **Example:** `Device(config-flow-exporter)# transport udp 650`	Configures UDP as the transport protocol and specifies the UDP port on which the destination system is listening for exported datagrams. • The range for the *udp-port* argument is from 1 to 65536.

	Command or Action	Purpose
Step 13	**ttl** *seconds* **Example:** `Device(config-flow-exporter)# ttl 15`	(Optional) Configures the time-to-live (TTL) value for datagrams sent by the exporter. • The range for the *seconds* argument is from 1 to 255.

Step 14	**end**
	Example:
	`Device(config-flow-exporter)# end`
		Exits flow exporter configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

To check the configuration and status of your flow exporter, use the **show flow exporter** command.

Configuring a Flow Record for Cisco Performance Monitor

The basic concepts and techniques for configuring a flow record for Cisco Performance Monitor are the same as flow records for Flexible NetFlow. The flow record specifies how the data collected data is aggregated and presented. The only significant difference is that, for Cisco Performance Monitor, the command includes **type performance-monitor**.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record type performance-monitor** *record-name*
4. **match application** {**name** [**account-on-resolution**] | **vendor** | **version**}
5. **match connection transaction-id**
6. **match flow** {**direction** | **sampler**}
7. **match interface** {**input** | **output**}
8. **match ipv4** {**destination**{**address** | **prefix** [**minimum-mask** *mask*]} | **protocol** | **source** {**address** | **prefix** [**minimum-mask** *mask*]}
9. **match ipv4 fragmentation** {**flags** |**offset**}
10. **match ipv4** {**section** {**header size** *header-size* | **payload size** *payload-size*}
11. **match ipv4 total-length**
12. **match ipv4 ttl**
13. **match ipv6** {**dscp** | **flow-label** | **next-header** | **payload-length** | **precedence** | **protocol** | **traffic-class** | **version**}
14. **match ipv6 destination** {**address** | {**mask** | **prefix**} [**minimum-mask** *mask*]}
15. **match ipv6 extension map**
16. **match ipv6 fragmentation** {**flags** | **id** | **offset**}
17. **match ipv6 hop-limit**
18. **match ipv6 length** {**header** | **payload** | **total**}
19. **match ipv6** {**section** {**header size** *header-size* | **payload size** *payload-size*}
20. **match ipv6 source** {**address** | {**mask** | **prefix**} [**minimum-mask** *mask*]}

21. **match metadata** {**global-session-id** | **multi-party-session-id**}
22. **match routing** {**destination** | **source**}
23. **match routing is-multicast**
24. **match routing multicast replication-factor**
25. **match transport** {**destination-port** | **igmp** | **rtp** [**ssrc**] | **source-port**}
26. **match transport icmp ipv4** {**code** | **type**}
27. **match transport icmp ipv6** {**code** | **type**}
28. **match transport tcp** {**acknowledgement-number** | **destination-port** | **flags** {[**ack**] | [**cwr**] | [**ece**] | [**fin**] | [**psh**] | [**syn**] | [**urg**]} | **header-length** | **maximum-segment-size** | **sequence-number** | **urgent-pointer** | **window-size** | **window-size-maximum** | **window-size-minimum** | **window-size-average**}
29. **match transport udp** {**destination-port** | **message-length** | **source-port**}
30. **collect application media** {**bytes**{**rate** | **counter**}| **packets** {**rate**|**counter**} | **events**}
31. **collect application** {**name** [**account-on-resolution**]| **description** | **http host** | **nntp group-name** | **pop3 server** | **rstp host-name** | **sip** {**destination** | **source**} | **smtp** {**sender** | **server**} | **vendor** | **version**}
32. **collect connection**
33. **collect counter** {**bytes** [**long** | **rate**] |**packets**[**dropped** [**long**] | **long**]}
34. **collect datalink mac source address** {**input** | **output**}
35. **collect flow direction**
36. **collect interface** {**input** | **output**}
37. **collect ipv4** {**destination mask** [**minimum-mask** *mask*]} | **dscp** | **source mask** [**minimum-mask** *mask*] | **ttl** [**minimum** | **maximum**]}
38. **collect ipv4 fragmentation** {**flags** | **offset**}
39. **collect ipv4** {**section** {**header size** *header-size* | **prefix**[**payload size** *payload-size*}
40. **collect ipv4 total-length** [**maximum** | **minimum**]
41. **collect ipv6** {**dscp** | **flow-label** | **next-header** | **payload-length** | **precedence** | **protocol** | **traffic-class** | **version**}
42. **collect ipv6 destination** {**address** {**mask** | **prefix**} [**minimum-mask** *mask*]}
43. **collect ipv6 extension-map**
44. **collect ipv6 fragmentation** {**flags** | **offset**}
45. **collect ipv6 hop-limit** [**maximum**] [**minimum**]
46. **collect ipv6 length**{**header** | **payload** | **total** [**maximum**] [**minimum**] }
47. **collect ipv6** {**section** {**header size** *header-size* | **prefix** [**payload size** *payload-size*}
48. **collect ipv6 source** {**address** {**mask** | **prefix**} [**minimum-mask** *mask*]}
49. **collect metadata** {**global-session-id** | **multi-party-session-id**}
50. **collect monitor event**
51. **collect routing forwarding-status** [**reason**]
52. **collect routing is-multicast**
53. **collect routing multicast replication-factor**
54. **collect timestamp internal**
55. **collect timestamp sys-uptime** {**first** | **last**}
56. **collect transport** {**destination-port** | **igmp type** | **source-port** | **event packet-loss counter** | **packets** {**expected counter** | **lost** {**counter** | **rate**} | **out-of-order**} | **round-trip-time** | **rtp jitter** {**minimum** | **mean** | **maximum**}}
57. **collect transport icmp ipv4**
58. **collect transport icmp ipv6**

59. **collect transport tcp** {**acknowledgement-number** | **destination-port** | **flags** {[**ack**] | [**cwr**] | [**ece**] | [**fin**] | [**psh**] | [**syn**] | [**urg**]} | **header-length** | **maximum-segment-size** | **sequence-number** | **urgent-pointer** | **window-size** | **window-size-maximum** | **window-size-minimum** | **window-size-average**}
60. **collect transport udp** {**destination-port** | **message-length** | **source-port**}
61. **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	**flow record type performance-monitor** *record-name* **Example:** `Device(config)# flow record type performance-monitor record-8`	Creates a flow record and enters flow record configuration mode. • This command also allows you to modify an existing flow record.

Step 4	**match application** {**name** [**account-on-resolution**] | **vendor** | **version**}
	Example:
	`Device(config-flow-record)# match application name`
		Specifies that the application name, vendor, or version will be used as a key field.

Step 5	**match connection transaction-id** **Example:** `Device(config-flow-record)# match connection transaction-id`	Specifies that the application name will be used as a key field.

Step 6	**match flow** {**direction** | **sampler**}
	Example:
	`Device(config-flow-record)# match flow direction`
		Specifies that the flow direction field will be used as a key field.

Step 7	**match interface** {**input** | **output**}
	Example:
	`Device(config-flow-record)# match flow direction`
		Specifies that the input interface field will be used as a key field.

	Command or Action	Purpose

Step 8	**match ipv4** {**destination**{**address** | **prefix** [**minimum-mask** *mask*]} | **protocol** | **source** {**address** | **prefix** [**minimum-mask** *mask*]}
	Example:
	`Device(config-flow-record)# match ipv4 destination address`
		Specifies that one or more of the IPv4 fields will be used as a key field.

Step 9	**match ipv4 fragmentation** {**flags** |**offset**}
	Example:
	`Device(config-flow-record)# match ipv4 fragmentation flags`
		Specifies that one or more of the IPv4 fields will be used as a key field.

Step 10	**match ipv4** {**section** {**header size** *header-size* | **payload size** *payload-size*}
	Example:
	`Device(config-flow-record)# match ipv4 section header size 8`
		Specifies that one or more of the IPv4 fields will be used as a key field.

Step 11	**match ipv4 total-length** **Example:** `Device(config-flow-record)# match ipv4 total-length`	Specifies that the IPv4 total length field will be used as a key field.
Step 12	**match ipv4 ttl** **Example:** `Device(config-flow-record)# match ipv4 ttl`	Specifies that the IPv4 ttl field will be used as a key field.

Step 13	**match ipv6** {**dscp** | **flow-label** | **next-header** | **payload-length** | **precedence** | **protocol** | **traffic-class** | **version**}
	Example:
	`Device(config-flow-record)# match ipv6 dscp`
		Specifies that the IPv6 DSCP field will be used as a key field.

Step 14	**match ipv6 destination** {**address** | {**mask** | **prefix**} [**minimum-mask** *mask*]}
	Example:
	`Device(config-flow-record)# match ipv4 destination address`
		Specifies that the IPv6 destination address field will be used as a key field.

Step 15	**match ipv6 extension map** **Example:**	Specifies that the IPv6 extension map field will be used as a key field.

	Command or Action	Purpose
	`Device(config-flow-record)# match ipv6 extension map`	

Step 16	**match ipv6 fragmentation** {**flags** | **id** | **offset**}
	Example:
	`Device(config-flow-record)# match ipv6 fragmentation flags`
		Specifies that the IPv6 fragmentation flags field will be used as a key field.

Step 17	**match ipv6 hop-limit** **Example:** `Device(config-flow-record)# match ipv6 hop-limit`	Specifies that the IPv6 hop limit field will be used as a key field.

Step 18	**match ipv6 length** {**header** | **payload** | **total**}
	Example:
	`Device(config-flow-record)# match ipv6 length total`
		Specifies that the IPv6 total length field will be used as a key field.

Step 19	**match ipv6** {**section** {**header size** *header-size* | **payload size** *payload-size*}
	Example:
	`Device(config-flow-record)# match ipv6 section header size 8`
		Specifies that the IPv6 section header size field will be used as a key field.

Step 20	**match ipv6 source** {**address** | {**mask** | **prefix**} [**minimum-mask** *mask*]}
	Example:
	`Device(config-flow-record)# match ipv6 source address`
		Specifies that the IPv6 source address field will be used as a key field.

Step 21	**match metadata** {**global-session-id** | **multi-party-session-id**}
	Example:
	`Device(config-flow-record)# match metadata global-session-id`
		Specifies that a metadata session ID field will be used as a key field.

Step 22	**match routing** {**destination** | **source**}
	Example:
	`Device(config-flow-record)# match routing source`
		Specifies that the routing source flag field will be used as a key field.

Step 23	**match routing is-multicast** **Example:**	Specifies that the routing is-multicast flag field will be used as a key field.

	Command or Action	Purpose
	`Device(config-flow-record)# match routing is-multicast`	
Step 24	**match routing multicast replication-factor** **Example:** `Device(config-flow-record)# match routing multicast replication-factor`	Specifies that the routing multicast replication-factor flag field will be used as a key field.

Step 25	**match transport** {**destination-port** | **igmp** | **rtp** [**ssrc**] | **source-port**}
	Example:
	`Device(config-flow-record)# match transport destination-port`
		Specifies that one or more of the transport layer fields will be used as a key field, including the Synchronization Source (SSRC) field in the Real-Time Transport Protocol (RTP) packet header.

Step 26	**match transport icmp ipv4** {**code** | **type**} **Example:** `Device(config-flow-record)# match transport icmp ipv4 code`	Specifies that the IPv4 ICMP transport code field will be used as a key field.
Step 27	**match transport icmp ipv6** {**code** | **type**} **Example:** `Device(config-flow-record)# match transport icmp ipv6 code`	Specifies that the IPv6 ICMP transport code field will be used as a key field.

Step 28	**match transport tcp** {**acknowledgement-number** | **destination-port** | **flags** {[**ack**] | [**cwr**] | [**ece**] | [**fin**] | [**psh**] | [**syn**] | [**urg**]} | **header-length** | **maximum-segment-size** | **sequence-number** | **urgent-pointer** | **window-size** | **window-size-maximum** | **window-size-minimum** | **window-size-average**}
	Example:
	`Device(config-flow-record)# match transport tcp destination-port`
		Specifies that the IPv6 TCP transport destination port field will be used as a key field.

Step 29	**match transport udp** {**destination-port** | **message-length** | **source-port**}
	Example:
	`Device(config-flow-record)# match transport udp destination-port`
		Specifies that the IPv6 UDP transport destination port field will be used as a key field.

Step 30	**collect application media** {**bytes**{**rate** | **counter**}| **packets** {**rate**|**counter**} | **events**} **Example:**	Specifies that the application media bytes, packets, or events will be used as a nonkey field. An application event occurs when either one of the thresholds specified by a

	Command or Action	Purpose
	`Device(config-flow-record)# collect application media events`	react statement for the flow was crossed at least once in the monitoring interval or no media packets were seen.

Step 31	**collect application** {**name** [**account-on-resolution**]| **description** | **http host** | **nntp group-name** | **pop3 server** | **rstp host-name** | **sip** {**destination** | **source**} | **smtp** {**sender** | **server**} | **vendor** | **version**}
	Example:
	`Device(config-flow-record)# collect application name`
		Specifies that the application name will be used as a nonkey field.

Step 32	**collect connection** **Example:** `Device(config-flow-record)# collect connection initiator`	Specifies that the connection initiator will be used as a nonkey field.

Step 33	**collect counter** {**bytes** [**long** | **rate**] |**packets**[**dropped** [**long**] | **long**]}
	Example:
	`Device(config-flow-record)# collect counter bytes long`
		Specifies the number of bytes or packets that will be used as a nonkey field.

Step 34	**collect datalink mac source address** {**input** | **output**} **Example:** `Device(config-flow-record)# collect flow direction`	Specifies that the flow direction field will be used as a nonkey field.
Step 35	**collect flow direction** **Example:** `Device(config-flow-record)# collect flow direction`	Specifies that the flow direction field will be used as a nonkey field.

Step 36	**collect interface** {**input** | **output**}
	Example:
	`Device(config-flow-record)# collect interface input`
		Specifies that the input or output interface will be used as a nonkey field.

Step 37	**collect ipv4** {**destination mask** [**minimum-mask** *mask*]} | **dscp** | **source mask** [**minimum-mask** *mask*] | **ttl** [**minimum** | **maximum**]}
	Example:
	`Device(config-flow-record)# collect ipv4 dscp`
		Specifies that the IPv4 DSCP field will be used as a nonkey field.

	Command or Action	Purpose

Step 38	**collect ipv4 fragmentation** {**flags** | **offset**}
	Example:
	`Device(config-flow-record)# collect ipv4 fragmentation flags`
		Specifies that the IPv4 fragmentation flags field will be used as a nonkey field.

Step 39	**collect ipv4** {**section** {**header size** *header-size* | **prefix**[**payload size** *payload-size*}
	Example:
	`Device(config-flow-record)# collect ipv4 section header size 8`
		Specifies that the IPv4 section header size field will be used as a nonkey field.

Step 40	**collect ipv4 total-length** [**maximum** | **minimum**] **Example:** `Device(config-flow-record)# collect ipv4 total-length`	Specifies that the IPv4 total-length field will be used as a nonkey field.

Step 41	**collect ipv6** {**dscp** | **flow-label** | **next-header** | **payload-length** | **precedence** | **protocol** | **traffic-class** | **version**}
	Example:
	`Device(config-flow-record)# collect ipv6 dscp`
		Specifies that the IPv6 DSCP field will be used as a nonkey field.

Step 42	**collect ipv6 destination** {**address** {**mask** | **prefix**} [**minimum-mask** *mask*]}
	Example:
	`Device(config-flow-record)# collect ipv6 destination mask`
		Specifies that the IPv6 destination mask field will be used as a nonkey field.

Step 43	**collect ipv6 extension-map** **Example:** `Device(config-flow-record)# collect ipv6 extension-map`	Specifies that the IPv6 extension-map field will be used as a nonkey field.

Step 44	**collect ipv6 fragmentation** {**flags** | **offset**}
	Example:
	`Device(config-flow-record)# collect ipv6 fragmentation flags`
		Specifies that the IPv6 fragmentation flags field will be used as a nonkey field.

Step 45	**collect ipv6 hop-limit** [**maximum**] [**minimum**] **Example:** `Device(config-flow-record)# collect ipv6 hop-limit`	Specifies that the IPv6 hop-limit field will be used as a nonkey field.

	Command or Action	Purpose

Step 46	**collect ipv6 length**{**header** | **payload** | **total** [**maximum**] [**minimum**] }
	Example:
	`Device(config-flow-record)# collect ipv6 length total`
		Specifies that the IPv6 total length field will be used as a nonkey field.

Step 47	**collect ipv6** {**section** {**header size** *header-size* | **prefix** [**payload size** *payload-size*}
	Example:
	`Device(config-flow-record)# collect ipv6 section header size 8`
		Specifies that the IPv6 section header size field will be used as a nonkey field.

Step 48	**collect ipv6 source** {**address** {**mask** | **prefix**} [**minimum-mask** *mask*]}
	Example:
	`Device(config-flow-record)# collect ipv6 source mask`
		Specifies that the IPv6 source mask field will be used as a nonkey field.

Step 49	**collect metadata** {**global-session-id** | **multi-party-session-id**}
	Example:
	`Device(config-flow-record)# collect meatdata global-session-id`
		Specifies that a metadata session ID field will be used as a nonkey field.

Step 50	**collect monitor event** **Example:** `Device(config-flow-record)# collect monitor event`	Specifies that the monitor event field will be used as a nonkey field. A monitor event occurs when no media application packets were seen
Step 51	**collect routing forwarding-status** [**reason**] **Example:** `Device(config-flow-record)# collect routing forwarding-status`	Specifies that the one or more of the routing attributes will be used as a nonkey field.
Step 52	**collect routing is-multicast** **Example:** `Device(config-flow-record)# collect routing is-multicast`	Specifies that the routing is-multicast field will be used as a nonkey field.
Step 53	**collect routing multicast replication-factor** **Example:**	Specifies that the routing multicast replication-factor field will be used as a nonkey field.

	Command or Action	Purpose
	`Device(config-flow-record)# collect routing multicast replication-factor`	
Step 54	**collect timestamp internal** **Example:** `Device(config-flow-record)# collect timestamp internal`	Specifies that the system timestamp of the first seen or last seen packet in a flow will be used as a nonkey field.

Step 55	**collect timestamp sys-uptime** {**first** | **last**}
	Example:
	`Device(config-flow-record)# collect timestamp sys-uptime`
		Specifies that the system timestamp of the sys-uptime will be used as a nonkey field.

Step 56	**collect transport** {**destination-port** | **igmp type** | **source-port** | **event packet-loss counter** | **packets** {**expected counter** | **lost** {**counter** | **rate**} | **out-of-order**} | **round-trip-time** | **rtp jitter** {**minimum** | **mean** | **maximum**}}
	Example:
	`Device(config-flow-record)# collect transport packets expected counter`
		Specifies that one or more of the transport layer fields will be used as a nonkey field. These fields include metrics for:
		• Packet-loss counter
		• Expected packets counter
		• Jitter

Step 57	**collect transport icmp ipv4** **Example:** `Device(config-flow-record)# collect transport icmp ipv4`	Specifies that the transport ICMP IPv4 field will be used as a nonkey field.
Step 58	**collect transport icmp ipv6** **Example:** `Device(config-flow-record)# collect transport icmp ipv6`	Specifies that the transport ICMP IPv6 field will be used as a nonkey field.

Step 59	**collect transport tcp** {**acknowledgement-number** | **destination-port** | **flags** {[**ack**] | [**cwr**] | [**ece**] | [**fin**] | [**psh**] | [**syn**] | [**urg**]} | **header-length** | **maximum-segment-size** | **sequence-number** | **urgent-pointer** | **window-size** | **window-size-maximum** | **window-size-minimum** | **window-size-average**}
	Example:
	`Device(config-flow-record)# collect transport tcp destination-port`	

	Command or Action	Purpose

Step 60	**collect transport udp** {**destination-port** | **message-length** | **source-port**}
	Example:
	`Device(config-flow-record)# collect transport udp destination-port`
		Specifies that the transport UDP destination port field will be used as a nonkey field.

Step 61	**end**
	Example:
	`Device(config-flow-record)# end`
		Exits flow record configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

To check the configuration and status of your flow record, use the **show flow record type performance-monitor** command.

Configuring a Usage Record for AVC Phase 2

To configure an input usage record, perform the following required task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. flow record flow-record-name
4. match interface input
5. match flow direction
6. match connection client {ipv4 | ipv6} address
7. match connection client transport port
8. match connection server {ipv4 | ipv6} address
9. match connection server transport port
10. match ipv4 {initiator | responder} address
11. match ipv6 {initiator | responder} address
12. match transport {initiator | responder} port
13. match routing vrf {input | output}
14. match datalink {destination-vlan-id | source-vlan-id}
15. match datalink vlan {input | output}
16. match datalink mac {destination | source} address {input | output}
17. match flow {class | qos-class}
18. match policy performance-monitor classification hierarchy
19. match services waas segment
20. collect interface output
21. collect flow direction
22. collect timestamp sys-uptime first

23. collect timestamp sys-uptime last
24. collect counter bytes long
25. collect counter packets
26. collect connection client {ipv4 | ipv6} address
27. collect connection client counter {bytes long | packets long | packets retransmitted}
28. collect connection client transport port
29. collect connection new-connections
30. collect connection sum-duration
31. collect routing vrf {input | output}
32. collect connection delay application {sum | min | max}
33. collect connection delay network {client-to-server | to-server [histogram { bucket1 | bucket2 | bucket3 | bucket4 | bucket5 | bucket6 | bucket7}] {sum | min | max}
34. collect connection delay response {client-to-server | to-client | to-server} {sum | min | max}
35. collect connection performance application-delay {sum | min | max}
36. collect connection performance initiator bytes long
37. collect connection performance initiator count re-transmitted-packets
38. collect connection performance initiator network-delay {sum | min | max}
39. collect connection performance initiator packets long
40. collect connection performance network-delay {sum | min | max}
41. collect connection performance new-transaction-time
42. collect connection performance total-transaction-time {sum | min | max}
43. collect connection performance total-transaction-time {sum | min | max}
44. collect connection performance responder bytes long
45. collect connection performance responder response-time {sum | min | max}
46. collect connection performance responder network-delay {sum | min | max}
47. collect connection performance responder count {histogram { bucket1 | bucket2 | bucket3 | bucket4 | bucket5 | bucket6 | bucket7} | late-responses | responses}
48. collect connection performance responder packets long
49. collect connection performance total-delay {sum | min | max}
50. collect connection performance total-transaction-time {sum | min | max}
51. collect connection server {ipv4 | ipv6} address
52. collect connection server counter {bytes long | packets long | packets retransmitted}
53. collect connection server transport port
54. collect connection transaction {counter complete | duration {sum | min | max}}
55. collect datalink {destination-vlan-id | source-vlan-id}
56. collect datalink mac {destination | source} address {input | output}
57. collect datalink vlan {input | output}
58. collect policy performance-monitor classification hierarchy
59. collect services waas {passthrough-reason | segment}
60. collect timestamp absolute {first | last}
61. collect transport tcp {option map | window-size {sum | minimum | maximum} | maximum-segment-size}
62. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	flow record flow-record-name **Example:** `Router(config)# flow record my-input-usage-monitor`	Creates a flow record and enters flow record configuration mode.
Step 4	match interface input **Example:** `Router(config-flow-record)# match interface input`	Configures the input interface for the packet as a key field for the flow record. input—Traffic arrives on the Cisco router's input interface.
Step 5	match flow direction **Example:** `Router(config-flow-record)# match flow direction`	Configures the direction of the flow record as a key field. The direction is either input or output.
Step 6	match connection client {ipv4 | ipv6} address **Example:** `Router(config-flow-record)# match connection client ipv6 address`	Configures the Ipv6 address of the client as a key field for a flow record.
Step 7	match connection client transport port **Example:** `Router(config-flow-record)# match connection client transport port`	Configures the connection port of the client as a key field for a flow record.
Step 8	match connection server {ipv4 | ipv6} address **Example:** `Router(config-flow-record)# match connection server ipv6 address`	Configures the Ipv6 address of the server as a key field for a flow record.
Step 9	match connection server transport port **Example:**	Configures the connection port of the server as a key field for a flow record.

	Command or Action	Purpose
	`Router(config-flow-record)# match connection server transport port`	
Step 10	match ipv4 {initiator | responder} address **Example:** `Router(config-flow-record)# match ipv4 initiator address`	(Optional) For IPv4 networks, configures the IPv4 address of the initiator or responder as a key field. The direction is either input or output.
Step 11	match ipv6 {initiator | responder} address **Example:** `Router(config-flow-record)# match ipv6 initiator address`	(Optional) For IPv6 networks, configures the IPv6 address of the initiator or responder as a key field. The direction is either input or output.
Step 12	match transport {initiator | responder} port **Example:** `Router(config-flow-record)# match transport initiator port`	(Optional) Configures the transport port of the initiator or responder as a key field.

Step 13	match routing vrf {input | output}
	Example:
	`Router(config-flow-record)# match routing vrf input`
		(Optional) Configures the virtual routing and forwarding (VRF) ID for incoming or outgoing packets as a key field.

Step 14	match datalink {destination-vlan-id | source-vlan-id}
	Example:
	`Router(config-flow-record)# match datalink destination-vlan-id`
		(Optional) Configures the destination VLAN ID as a key field.

Step 15	match datalink vlan {input | output}
	Example:
	`Router(config-flow-record)# match datalink vlan input`
		(Optional) Configures the VLAN ID for incoming or outgoing packets as a key field.

Step 16	match datalink mac {destination | source} address {input | output}
	Example:
	`Router(config-flow-record)# match datalink mac destination address output`
		(Optional) Configures the destination MAC address as a key field.

Step 17	match flow {class | qos-class} **Example:**	Configures the use of the class ID as a key field for a flow record.

	Command or Action	Purpose
	`Router(config-flow-record)# match flow class`	
Step 18	match policy performance-monitor classification hierarchy **Example:** `Router(config-flow-record)# match policy performance-monitor classification hierarchy`	Configures the use of the Performance Monitor policy classification hierarchy as a key field for a flow record.
Step 19	match services waas segment **Example:** `Router(config-flow-record)# match services waas segment`	Configures the use of the WAAS segment as a key field for a flow record.
Step 20	collect interface output **Example:** `Router(config-flow-record)# collect interface output`	Configures the output interface as a non-key field for a flow record and enables collecting the output interface fields from the flows for the flow record.
Step 21	collect flow direction **Example:** `Router(config-flow-record)# collect flow direction`	Configures the flow direction as a non-key field for a flow record.
Step 22	collect timestamp sys-uptime first **Example:** `Router(config-flow-record)# collect timestamp sys-uptime first`	Configures the system uptime of the first seen packet in a flow as a nonkey field for a flow record. • first—Configures the system uptime for the time the first packet was seen from the flows as a nonkey field and enables collecting time stamps based on the system uptime for the time the first packet was seen from the flows.
Step 23	collect timestamp sys-uptime last **Example:** `Router(config-flow-record)# collect timestamp sys-uptime last`	Configures the system uptime of the last seen packet in a flow as a nonkey field for a flow record. • last—Configures the system uptime for the time the last packet was seen from the flows as a nonkey field and enables collecting time stamps based on the system uptime for the time the most recent packet was seen from the flows.
Step 24	collect counter bytes long **Example:** `Router(config-flow-record)# collect counter bytes long`	Configures the number of bytes in a flow as a nonkey field for a flow record. • bytes—Configures the number of bytes seen in a flow as a nonkey field and enables collecting the total number of bytes from the flow.

	Command or Action	Purpose
		• long—Enables collecting the total number of bytes or packets from the flow by using a 64-bit counter rather than a 32-bit counter.
Step 25	collect counter packets **Example:** `Router(config-flow-record)# collect counter packets`	Configures the number of packets in a flow as a nonkey field for a flow record. • packets—Configures the number of packets seen in a flow as a nonkey field and enables collecting the total number of packets from the flow.
Step 26	collect connection client {ipv4 | ipv6} address **Example:** `Router(config-flow-record)# collect connection client ipv6 address`	Configures the Ipv6 address of the client as a nonkey field for a flow record.
Step 27	collect connection client counter {bytes long | packets long | packets retransmitted} **Example:** `Router(config-flow-record)# collect connection client counter packets retransmitted`	Configures the number of the client packets retransmitted as a nonkey field for a flow record.
Step 28	collect connection client transport port **Example:** `Router(config-flow-record)# collect connection client transport port`	Configures the client connection port as a nonkey field for a flow record.
Step 29	collect connection new-connections **Example:** `Router(config-flow-record)# collect connection new-connections`	Counts the number of TCP or UDP connections which were opened during the observation period. The observation period may be specified by the flow start and end timestamps.
Step 30	collect connection sum-duration **Example:** `Router(config-flow-record)# collect connection sum-duration`	Aggregates the total time, in seconds, for all the TCP or UDP connections, which were in use during the observation period. For example, if there are five concurrent connections each for 10 seconds, the value would be 50 seconds.

Step 31	collect routing vrf {input | output}
	Example:
	`Router(config-flow-record)# collect routing vrf output`
		Configures the virtual routing and forwarding (VRF) ID for incoming or outgoing packets output as a nonkey field for a flow record.

	Command or Action	Purpose
Step 32	collect connection delay application {sum | min | max} **Example:** `Router(config-flow-record)# collect connection delay application sum`	Configures the total amount of application delay as a nonkey field for a flow record.
Step 33	collect connection delay network {client-to-server | to-server [histogram { bucket1 | bucket2 | bucket3 | bucket4 | bucket5 | bucket6 | bucket7}] {sum | min | max} **Example:** `Router(config-flow-record)# collect connection delay network client-to-server sum`	Configures the total amount of network delay between the client and the server as a nonkey field for a flow record.
Step 34	collect connection delay response {client-to-server | to-client | to-server} {sum | min | max} **Example:** `Router(config-flow-record)# collect connection delay response client-to-server sum`	Configures the total amount of response delay between the client and the server as a nonkey field for a flow record.
Step 35	collect connection performance application-delay {sum | min | max} **Example:** `Router(config-flow-record)# collect connection performance application-delay sum`	Configures the total application delay as a nonkey field for a flow record.
Step 36	collect connection performance initiator bytes long **Example:** `Router(config-flow-record)# collect connection performance initiator bytes long`	Configures the number of long bytes for the Mediatrace initiator as a nonkey field for a flow record.
Step 37	collect connection performance initiator count re-transmitted-packets **Example:** `Router(config-flow-record)# collect connection performance initiator count re-transmitted-packets`	Configures the number of retrransmitted packets for the Mediatrace initiator as a nonkey field for a flow record.
Step 38	collect connection performance initiator network-delay {sum | min | max} **Example:** `Router(config-flow-record)# collect connection performance initiator network-delay sum`	Configures the total network delay for the Mediatrace initiator as a nonkey field for a flow record.

	Command or Action	Purpose
Step 39	collect connection performance initiator packets long **Example:** `Router(config-flow-record)# collect connection performance initiator packets long`	Configures the number of long packets for the Mediatrace initiator as a nonkey field for a flow record.
Step 40	collect connection performance network-delay {sum | min | max} **Example:** `Router(config-flow-record)# collect connection performance network-delay sum`	Configures the total network delay as a nonkey field for a flow record.
Step 41	collect connection performance new-transaction-time **Example:** `Router(config-flow-record)# collect connection performance new-transaction`	Configures the new transaction field as a nonkey field for a flow record.
Step 42	collect connection performance total-transaction-time {sum | min | max} **Example:** `Router(config-flow-record)# collect connection performance total-transaction-time sum`	Configures the total transaction time as a nonkey field for a flow record.
Step 43	collect connection performance total-transaction-time {sum | min | max} **Example:** `Router(config-flow-record)# collect connection performance total-transaction-time sum`	Configures the total transaction time as a nonkey field for a flow record.
Step 44	collect connection performance responder bytes long **Example:** `Router(config-flow-record)# collect connection performance responder bytes long`	Configures the number of long bytes for the Mediatrace responder as a nonkey field for a flow record.
Step 45	collect connection performance responder response-time {sum | min | max} **Example:** `Router(config-flow-record)# collect connection performance responder response-time sum`	Configures the total response time for the Mediatrace responder as a nonkey field for a flow record.
Step 46	collect connection performance responder network-delay {sum | min | max} **Example:**	Configures the total network delay for the Mediatrace responder as a nonkey field for a flow record.

	Command or Action	Purpose
	`Router(config-flow-record)# collect connection performance responder network-delay sum`	
Step 47	collect connection performance responder count {histogram { bucket1 | bucket2 | bucket3 | bucket4 | bucket5 | bucket6 | bucket7} | late-responses | responses} **Example:** `Router(config-flow-record)# collect connection performance responder count late-responses`	Configures the number of late responses for the Mediatrace responder as a nonkey field for a flow record.
Step 48	collect connection performance responder packets long **Example:** `Router(config-flow-record)# collect connection performance responder packets long`	Configures the number of long packets for the Mediatrace responder as a nonkey field for a flow record.
Step 49	collect connection performance total-delay {sum | min | max} **Example:** `Router(config-flow-record)# collect connection performance total-delay sum`	Configures the total connection delay as a nonkey field for a flow record.
Step 50	collect connection performance total-transaction-time {sum | min | max} **Example:** `Router(config-flow-record)# collect connection performance total-transaction-time sum`	Configures the total transaction time as a nonkey field for a flow record.
Step 51	collect connection server {ipv4 | ipv6} address **Example:** `Router(config-flow-record)# collect connection server ipv6 address`	Configures the IPv6 address of the server as a nonkey field for a flow record.
Step 52	collect connection server counter {bytes long | packets long | packets retransmitted} **Example:** `Router(config-flow-record)# collect connection server counter packets retransmitted`	Configures the number of the server packets retransmitted as a nonkey field for a flow record.
Step 53	collect connection server transport port **Example:** `Router(config-flow-record)# collect connection server transport port`	Configures the server connection port as a nonkey field for a flow record.

	Command or Action	Purpose

Step 54	collect connection transaction {counter complete | duration {sum | min | max}}
	Example:
	`Router(config-flow-record)# collect connection transaction duration sum`
		Configures the total duration of the transaction as a nonkey field for a flow record.

Step 55	collect datalink {destination-vlan-id | source-vlan-id}
	Example:
	`Router(config-flow-record)# collect datalink destination-vlan-id`
		(Optional) Configures the destination VLAN ID as a nonkey field.

Step 56	collect datalink mac {destination | source} address {input | output}
	Example:
	`Router(config-flow-record)# collect datalink mac destination address input`
		(Optional) Configures the destination MAC address as a nonkey field.

Step 57	collect datalink vlan {input | output}
	Example:
	`Router(config-flow-record)# collect datalink vlan input`
		(Optional) Configures the VLAN ID for incoming or outgoing packets as a nonkey field.

Step 58	collect policy performance-monitor classification hierarchy **Example:** `Router(config-flow-record)# collect  policy performance-monitor classification hierarchy`	Configures the use of the Performance Monitor policy classification hierarchy as a nonkey field for a flow record.

Step 59	collect services waas {passthrough-reason | segment}
	Example:
	`Router(config-flow-record)# collect services waas segment`
		Configures the use of the WAAS segment as a nonkey field for a flow record.

Step 60	collect timestamp absolute {first | last}
	Example:
	`Router(config-flow-record)# collect timestamp absolute first`
		Configures the use of the first timestamp as a nonkey field for a flow record.

Step 61	collect transport tcp {option map | window-size {sum | minimum | maximum} | maximum-segment-size} **Example:**	Configures the total network delay for the Mediatrace initiator as a nonkey field for a flow record.

	Command or Action	Purpose
	`Router(config-flow-record)# collect connection performance initiator network-delay sum`	

Step 62	end
	Example:
	`Router(config-flow-record)# end`
		Exits flow record configuration mode and returns to privileged EXEC mode.

Configuring a Flow Monitor for Cisco Performance Monitor

The basic concepts for configuring a flow monitor for Cisco Performance Monitor are the same as flow monitors for Flexible NetFlow. Each flow monitor has a separate cache assigned to it and requires a record to define the contents and layout of its cache entries.

When you configure a flow monitor, you must use either:

- An existing flow record that you configured
- One of the following default predefined records:
  - The default RTP record (**default-rtp**)
  - The default TCP record (**default-tcp**)
  - Flexible NetFlow's "NetFlow IPv4 original input"

Note To modify a flow record, you must remove it from all flow monitors it is associated with.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor type performance-monitor** *monitor-name*
4. **description** *description*
5. **cache** {**entries**| **timeout**| **type**}
6. **statistics** {**packet**}
7. **exporter** *exporter-name*
8. **record** {*record-name*| **default-rtp**| **default-tcp**|**netflow ipv4 original-input**}
9. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor type performance-monitor** *monitor-name* **Example:** `Device(config)# flow monitor type performance-monitor FLOW-MONITOR-2`	Creates a flow monitor and enters flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for monitoring IPv4 traffic`	(Optional) Creates a description for the flow monitor.

Step 5	**cache** {**entries**| **timeout**| **type**}
	Example:
	`Device(config-flow-monitor)# cache timeout 20`
		(Optional) Creates a cache for the flow monitor.

Step 6	**statistics** {**packet**}
	Example:
	`Device(config-flow-monitor)# statistics`
		(Optional) specifies whether statistics are collected for the flow monitor.

Step 7	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter export-4`	Specifies the flow exporter for the flow monitor.

Step 8	**record** {*record-name*| **default-rtp**| **default-tcp**|**netflow ipv4 original-input**}
	Example:
	`Device(config-flow-monitor)# record default-rtp`
		Specifies the flow record for the flow monitor.

Step 9	**end**
	Example:
	`Device(config-flow-monitor)# end`
		Exits flow monitor configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

To check the configuration and status of your flow monitor, use the **show flow monitor type performance-monitor** command and the **show running-config flow monitor** command.

Configuring a Flow Class for Cisco Performance Monitor

The basic concepts and techniques for configuring a class for Cisco Performance Monitor are the same as for any other type of class. The class specifies the filter that determines which flow traffic to monitor. The filter is configured using various match commands in class-map mode.

If you do not already have a flow monitor configured, you can either:

Note Nested class maps are not supported. In other words, you cannot use the **class-map** command while in class-map configuration mode (config-cmap).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-name*
4. **description** *description*
5. **match** {*access-group* {*access-group* | **name** *access-group-name*} | **any** | **class-map** *class-map-name* | **cos** *cos-value* | **destination-address mac** *address* | **discard-class** *class-number* | **dscp** *dscp-value* | **flow** {**direction** | **sampler**} | **fr-de** | **fr-dlci** *dlci-number* | **input-interface** *interface-name* | **ip** {**rtp** *starting-port-number port-range* | **precedence** | **dscp**} | **mpls experimental topmost** *number* | **not** *match-criterion*| **packet length** {**max** *maximum-length-value* [**min** *minimum-length-value*] | **min** *minimum-length-value* [**max** *maximum-length-value*]} | **precedence** {*precedence-criteria1* | *precedence-criteria2* | *precedence-criteria3* | *precedence-criteria4*} | **protocol** *protocol-name* | **qos-group** *qos-group-value* | **source-address** *mac address-destination*| **vlan** {*vlan-id* | *vlan-range* | *vlan-combination*}}
6. **rename** *class-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.

	Command or Action	Purpose

Step 3	**class-map** *class-name*
	Example:
	`Device(config)# class-map class-4`
		Specifies a class to include in the policy. Repeat this command for each class that you want to include in the policy.

Step 4	**description** *description* **Example:** `Device(config-cmap)# description match any packets`	(Optional) Creates a description for the flow class.

Step 5	**match** {*access-group* {*access-group* | **name** *access-group-name*} | **any** | **class-map** *class-map-name* | **cos** *cos-value* | **destination-address mac** *address* | **discard-class** *class-number* | **dscp** *dscp-value* | **flow** {**direction** | **sampler**} | **fr-de** | **fr-dlci** *dlci-number* | **input-interface** *interface-name* | **ip** {**rtp** *starting-port-number port-range* | **precedence** | **dscp**} | **mpls experimental topmost** *number* | **not** *match-criterion*| **packet length** {**max** *maximum-length-value* [**min** *minimum-length-value*] | **min** *minimum-length-value* [**max** *maximum-length-value*]} | **precedence** {*precedence-criteria1* | *precedence-criteria2* | *precedence-criteria3* | *precedence-criteria4*} | **protocol** *protocol-name* | **qos-group** *qos-group-value* | **source-address** *mac address-destination*| **vlan** {*vlan-id* | *vlan-range* | *vlan-combination*}}
	Example:
	`Device(config-cmap)# match any`
		Specifies the classification criteria.
		For more information and examples, see the *Cisco Media Monitoring Command Reference*.

Step 6	**rename** *class-name* **Example:** `Device(config-cmap)# rename class-4`	Specifies a new name for the flow class.

Step 7	**end**
	Example:
	`Device(config-cmap)# end`
		Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

To check the configuration and status of your flow class, use the **show policy-map type performance-monitor** or **show class-map** command.

Configuring a Flow Policy for Cisco Performance Monitor Using an Existing Flow Monitor

The basic concepts and techniques for configuring a class for Cisco Performance Monitor are the same as for any other type of class. The class specifies which flow monitor is included. The only significant difference is that, for Cisco Performance Monitor, the **policy-map** command includes **type performance-monitor**.

If you do not already have a flow monitor configured or do not want to use any of your existing flow monitors for a new class, you can configure it using the flow monitor inline option and specifying which flow record and flow exporter are included.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type performance-monitor** *policy-name*
4. **parameter-map type performance-monitor system-default-aor**
5. **class** {*class-name* | **class-default**}
6. **flow monitor** *monitor-name*
7. **monitor metric ip-cbr**
8. **rate layer3** {byte-*rate* {**bps** | **kbps** | **mbps** | **gbps**} | **packet**}
9. **exit**
10. **monitor metric rtp**
11. **clock-rate** {*type-number* | *type-name* | **default**} *rate*
12. **max-dropout** *number*
13. **max-reorder** *number*
14. **min-sequential** *number*
15. **ssrc maximum** *number*
16. exit
17. **monitor parameters**
18. **flows** *number*
19. **interval duration** *number*
20. **history** *number*
21. **timeout** *number*
22. **exit**
23. **react** *ID* {**media-stop** | **mrv** | **rtp-jitter-average** | **transport-packets-lost-rate**}
24. **action** {**snmp** | **syslog**}
25. **alarm severity** {**alert** | **critical** | **emergency** | **error** | **info**}
26. **alarm type** {**discrete** | **grouped {count** *number* | **percent** *number*}
27. **threshold value** {**ge** *number* | **gt** *number* | **le** *number* | **lt** *number* | **range** *rng-start rng-end*}
28. **description** *description*
29. **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 type performance-monitor** *policy-name* **Example:** `Device(config)# policy-map type performance-monitor FLOW-MONITOR-4`	Creates a policy and enters policy configuration mode. - This command also allows you to modify an existing policy.
Step 4	**parameter-map type performance-monitor system-default-aor** **Example:** `Device(config-pmap)# parameter-map type performance-monitor system-default-aor`	Creates a parameter map for Performance Monitor. The only map available is the system-default -aor map

Step 5	**class** {*class-name* | **class-default**}
	Example:
	`Device(config-pmap)# class class-4`
		Specifies a class to include in the policy. Repeat this command for each class that you want to include in the policy.

Step 6	**flow monitor** *monitor-name* **Example:** `Device(config-pmap-c)# flow monitor FLOW-MONITOR-4`	Enters flow monitor configuration mode. If you do not want to use an existing flow monitor, you can use the **inline** option to configure a new one, as described in the Applying a Cisco Performance Monitor Policy to an Interface Without Using an Existing Flow Policy, on page 45.
Step 7	**monitor metric ip-cbr** **Example:** `Device(config-pmap-c)# monitor metric ip-cbr`	(Optional) Enters IP-CBR monitor metric configuration mode.

Step 8	**rate layer3** {byte-*rate* {**bps** | **kbps** | **mbps** | **gbps**} | **packet**}
	Example:
	`Device(config-pmap-c-mipcbr)# rate layer3 248 mbps`
		(Optional) Specifies the rate for monitoring the metrics.
		- *byte-rate* --Data rate in Bps, kBps, mBps, or gBps. The range is 1 to 65535.
		- **packet** --Packet rate in packets per second.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-pmap-c-mipcbr)# exit`	Returns to policy class configuration mode.
Step 10	**monitor metric rtp** **Example:** `Device(config-pmap-c)# monitor metric rtp`	Enters RTP monitor metric configuration mode.
Step 11	**clock-rate** {*type-number* | *type-name* | **default**} *rate* **Example:** `Device(config-pmap-c-mrtp)# clock-rate 8 9600`	Specifies the clock rate used to sample RTP video-monitoring metrics. For more information about the clock-type numbers and names, see the *Cisco Media Monitoring Command Reference.* The range for *rate* is 1 kHz to 192 kHz.
Step 12	**max-dropout** *number* **Example:** `Device(config-pmap-c-mrtp)# max-dropout 2`	Specifies the maximum number of dropouts allowed when sampling RTP video-monitoring metrics.
Step 13	**max-reorder** *number* **Example:** `Device(config-pmap-c-mrtp)# max-reorder 4`	Specifies the maximum number of reorders allowed when sampling RTP video-monitoring metrics.
Step 14	**min-sequential** *number* **Example:** `Device(config-pmap-c-mrtp)# min-sequential 2`	Specifies the minimum number of sequential packets required to identify a stream as being an RTP flow.
Step 15	**ssrc maximum** *number* **Example:** `Device(config-pmap-c-mrtp)# ssrc maximum 20`	Specifies the maximum number of SSRCs that can be monitored within the same flow. A flow is defined by the protocol, source/destination address, and source/destination port).
Step 16	exit **Example:** `Device(config-pmap-c-mrtp)# exit`	Returns to policy class configuration mode.
Step 17	**monitor parameters** **Example:** `Device(config-pmap-c)# monitor parameters`	Enters monitor parameters configuration mode.

	Command or Action	Purpose
Step 18	**flows** *number* **Example:** `Device(config-pmap-c-mparam)# flows 40`	Specifies the maximum number of flows for each monitor cache.
Step 19	**interval duration** *number* **Example:** `Device(config-pmap-c-mparam)# interval duration 40`	Specifies the interval, in seconds, between samples taken of video-monitoring metrics.
Step 20	**history** *number* **Example:** `Device(config-pmap-c-mparam)# history 4`	Specifies the number of historical buckets of collected video-monitoring metrics.
Step 21	**timeout** *number* **Example:** `Device(config-pmap-c-mparam)# timeout 20`	Specifies the number of intervals before a stopped flow is removed from the database.
Step 22	**exit** **Example:** `Device(config-pmap-c-mparam)# exit`	Returns to policy class configuration mode.

Step 23	**react** *ID* {**media-stop** | **mrv** | **rtp-jitter-average** | **transport-packets-lost-rate**}
	Example:
	`Device(config-pmap-c)# react 41 rtp-jitter-average`
		Enters a mode where you can specify what reaction occurs when a threshold is violated for the following metrics:
		• *ID*-- ID for react configuration. Range is 1 to 65535.
		• **media-stop** --No traffic is found for the flow.
		• **mrv** --Ratio calculated by dividing the difference between the actual rate and the expected rate, by the expected rate.
		• **rtp-jitter-average** --Average jitter.
		• **transport-packets-lost-rate** --Ratio calculated by dividing the number of lost packets by the expected packet count.

Step 24	**action** {**snmp** | **syslog**}
	Example:
	`Device(config-pmap-c-react)# action syslog`
		Specifies how violations of the thresholds with be reported.

	Command or Action	Purpose

Step 25	**alarm severity** {**alert** | **critical** | **emergency** | **error** | **info**}
	Example:
	`Device(config-pmap-c-react)# alarm severity critical`
		Specifies which level of alarm will be reported. The default setting is **info**.

Step 26	**alarm type** {**discrete** | **grouped {count** *number* | **percent** *number*}
	Example:
	`Device(config-pmap-c-react)# alarm type discrete`
		Specifies which types of levels are considered alarms that require reporting. The default setting is **discrete**.

Step 27	**threshold value** {**ge** *number* | **gt** *number* | **le** *number* | **lt** *number* | **range** *rng-start rng-end*}
	Example:
	`Device(config-pmap-c-react)# threshold value ge 20`
		Specifies which types of threshold values are considered alarms that require reporting.
		If no value is set but the application name is configured as a key field, then the system uses the value for the threshold that it finds in the default map. If no value is set and the application name is not configured as a key field, then the default value is used for the threshold.
		If more than one react command is configured for the same policy and class but only one of the react configurations has threshold values set, then the values of the configured react take precedence and the rest of the threshold values are ignored.
		If more than one react command is configured for the same policy and none of them have the threshold value configured, then the default threshold value is applied for the configuration with the lowest react ID.

Step 28	**description** *description* **Example:** `Device(config-cmap-c-react)# description rtp-jitter-average above 40`	(Optional) Creates a description for the reaction.

Step 29	**end**
	Example:
	`Device(config-pmap-c-react)# end`
		Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

To check the configuration and status of your flow policy, use the **show policy-map type performance-monitor** command.

Configuring a Flow Policy for Cisco Performance Monitor Without Using an Existing Flow Monitor

The basic concepts and techniques for configuring a class for Cisco Performance Monitor are the same as for any other type of class. The class specifies which flow monitor is included. The only significant difference is that, for Cisco Performance Monitor, the **policy-map** command includes**type performance-monitor**.

If you do not already have a flow monitor configured or do not want to use any of your existing flow monitors for a new class, you can configure it under the class configuration mode, by specifying which flow record and flow exporter are included.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type performance-monitor** *policy-name* **class** class-name
4. **parameter-map type performance-monitor system-default-aor**
5. **class** {*class-name* | **class-default**}
6. **flow monitor inline**
7. **record** {*record-name* | **default-rtp** | **default-tcp**}
8. **exporter** *exporter-name*
9. **exit**
10. monitor metric ip-cbr
11. **rate layer3** {*byte-rate* {**bps** | **kbps** | **mbps** | **gbps**} | **packet**}
12. **exit**
13. **monitor metric rtp**
14. **clock-rate** {*type-number*| *type-name*} *rate*
15. **max-dropout** *number*
16. **max-reorder** *number*
17. **min-sequential** *number*
18. **ssrc maximum** *number*
19. exit
20. **monitor parameters**
21. **flows** *number*
22. **interval duration** *number*
23. **history** *number*
24. **timeout** *number*
25. **exit**
26. **react** *ID* {**media-stop** | **mrv** | **rtp-jitter-average** | **transport-packets-lost-rate**}
27. **action** {**snmp** | **syslog**}
28. **alarm severity** {**alert** | **critical** | **emergency** | **error** | **info**}
29. **alarm type** {**discrete** | **grouped {count** *number* | **percent** *number*}
30. **threshold value** {**ge** *number* | **gt** *number* | **le** *number* | **lt** *number* | **range** *rng-start rng-end*
31. **description** *description*
32. **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 type performance-monitor** *policy-name* **class** class-name **Example:** `Device(config)# policy-map type performance-monitor FLOW-MONITOR-4`	Creates a policy and enters policy configuration mode. - This command also allows you to modify an existing policy.
Step 4	**parameter-map type performance-monitor system-default-aor** **Example:** `Device(config-pmap)# parameter-map type performance-monitor system-default-aor`	Creates a parameter map for Performance Monitor. The only map available is the system-default -aor map

Step 5	**class** {*class-name* | **class-default**}
	Example:
	`Device(config-pmap)# class class-4`
		Specifies a class to include in the policy. Repeat this command for each class that you want to include in the policy.

Step 6	**flow monitor inline** **Example:** `Device(config-pmap-c)# flow monitor inline`	Enters inline mode and enables you to configure a new flow monitor.

Step 7	**record** {*record-name* | **default-rtp** | **default-tcp**}
	Example:
	`Device(config-pmap-c-flowmon)# record default-tcp`
		Specifies a flow record to associate with the flow monitor.

Step 8	**exporter** *exporter-name* **Example:** `Device(config-pmap-c-flowmon)# exporter exporter-4`	Specifies a flow record to associate with the flow exporter.
Step 9	**exit** **Example:**	Returns to policy class configuration mode.

	Command or Action	Purpose
	`Device(config-pmap-c-flowmon)# exit`	
Step 10	monitor metric ip-cbr **Example:** `Device(config-pmap-c)# monitor metric ip-cbr`	(Optional) Enters IP-CBR monitor metric configuration mode.

Step 11	**rate layer3** {*byte-rate* {**bps** | **kbps** | **mbps** | **gbps**} | **packet**}
	Example:
	`Device(config-pmap-c-mipcbr)# rate layer3 248 mbps`
		(Optional) Specifies the rate for monitoring the metrics.
		• *byte-rate*—Data rate in Bps, kBps, mBps, or gBps. The range is 1 to 65535.
		• **packet**—Packet rate in packets per second.

Step 12	**exit** **Example:** `Device(config-pmap-c-mipcbr)# exit`	Returns to policy class configuration mode.
Step 13	**monitor metric rtp** **Example:** `Device(config-pmap-c)# monitor metric rtp`	Enters RTP monitor metric configuration mode.
Step 14	**clock-rate** {*type-number*| *type-name*} *rate* **Example:** `Device(config-pmap-c-mrtp)# clock-rate 8 9600`	Specifies the clock rate used to sample RTP video-monitoring metrics. For more information about the clock-type numbers and names, see the *Cisco Media Monitoring Command Reference*. The range for *rate* is 1 kHz to 192 kHz.
Step 15	**max-dropout** *number* **Example:** `Device(config-pmap-c-mrtp)# max-dropout 2`	Specifies the maximum number of dropouts allowed when sampling RTP video-monitoring metrics.
Step 16	**max-reorder** *number* **Example:** `Device(config-pmap-c-mrtp)# max-reorder 4`	Specifies the maximum number of reorders allowed when sampling RTP video-monitoring metrics.
Step 17	**min-sequential** *number* **Example:** `Device(config-pmap-c-mrtp)# min-sequential 2`	Specifies the minimum number of sequential packets required to identify a stream as being an RTP flow.

	Command or Action	Purpose
Step 18	**ssrc maximum** *number* **Example:** `Device(config-pmap-c-mrtp)# ssrc maximum 20`	Specifies the maximum number of SSRCs that can be monitored within the same flow. A flow is defined by the protocol, source/destination address, and source/destination port).
Step 19	exit **Example:** `Device(config-pmap-c-mrtp)# exit`	Returns to policy class configuration mode.
Step 20	**monitor parameters** **Example:** `Device(config-pmap-c)# monitor parameters`	Enters monitor parameters configuration mode.
Step 21	**flows** *number* **Example:** `Device(config-pmap-c-mparam)# flows 40`	Specifies the maximum number of flows for each monitor cache.
Step 22	**interval duration** *number* **Example:** `Device(config-pmap-c-mparam)# interval duration 40`	Specifies the duration of the intervals, in seconds, for collecting monitoring metrics.
Step 23	**history** *number* **Example:** `Device(config-pmap-c-mparam)# history 4`	Specifies the number of historical intervals of collected monitoring metrics to display.
Step 24	**timeout** *number* **Example:** `Device(config-pmap-c-mparam)# timeout 20`	Specifies the number intervals before a stopped flow is removed from the database.
Step 25	**exit** **Example:** `Device(config-pmap-c-mparam)# exit`	Returns to policy class configuration mode.

Step 26	**react** *ID* {**media-stop** | **mrv** | **rtp-jitter-average** | **transport-packets-lost-rate**}
	Example:
	`Device(config-pmap-c)# react 41 rtp-jitter-average`
		Enters a mode where you can specify what reaction occurs when a threshold is violated for the following metrics:
		• *ID*—ID for react configuration. Range is 1 to 65535.
		• **media-stop**—No traffic is found for the flow.

	Command or Action	Purpose
		• **mrv**—Ratio calculated by dividing the difference between the actual rate and the expected rate, by the expected rate. • **rtp-jitter-average**—Average jitter. • **transport-packets-lost-rate**—Ratio calculated by dividing the number of lost packets by the expected packet count.

Step 27	**action** {**snmp** | **syslog**}
	Example:
	`Device(config-pmap-c-react)# action syslog`
		Specifies how violations of the thresholds with be reported.

Step 28	**alarm severity** {**alert** | **critical** | **emergency** | **error** | **info**}
	Example:
	`Device(config-pmap-c-react)# alarm severity critical`
		Specifies which level of alarm will be reported.The default setting is **info**.

Step 29	**alarm type** {**discrete** | **grouped {count** *number* | **percent** *number*}
	Example:
	`Device(config-pmap-c-react)# alarm severity critical`
		Specifies which types of levels are considered alarms that require reporting. The default setting is **discrete**.

Step 30	**threshold value** {**ge** *number* | **gt** *number* | **le** *number* | **lt** *number* | **range** *rng-start rng-end* **Example:** `Device(config-pmap-c-react)# threshold value ge 20`	Specifies which types of threshold values are considered alarms that require reporting. If no value is set but the application name is configured as a key field, then the system uses the value for the threshold that it finds in the default map. If no value is set and the application name is not configured as a key field, then the default value is used for the threshold. If more than one react command is configured for the same policy and class but only one of the react configurations has threshold values set, then the values of the configured react take precedence and the rest of the threshold values are ignored. If more than one react command is configured for the same policy and none of them have the threshold value configured, then the default threshold value is applied for the configuration with the lowest react ID.
Step 31	**description** *description* **Example:**	(Optional) Creates a description for the reaction.

	Command or Action	Purpose
	`Device(config-cmap-c-react)# description rtp-jitter-average above 40`	

Step 32	**end**
	Example:
	`Device(config-pmap-c-react)# end`
		Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

To check the configuration and status of your flow policy, use the **show policy-map type performance-monitor** command.

Applying a Cisco Performance Monitor Policy to an Interface Using an Existing Flow Policy

Before it can be activated, a Cisco Performance Monitor policy must be applied to at least one interface. To activate a Cisco Performance Monitor policy, perform the following required task.

Note You can apply a Cisco Performance Monitor policy to an IPv6 interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service-policy type performance-monitor** {**input** | **output**} *policy-name*
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.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/0`	Specifies an interface and enters interface configuration mode. You can specify an IPv6 interface.
Step 4	**service-policy type performance-monitor** {**input** | **output**} *policy-name* **Example:** **Example:** `Device(config-if)# service-policy type performance-monitor input mypolicy-map-4` **Example:**	Attaches a policy map to an input interface or virtual circuit (VC), or an output interface or VC, to be used as the service policy for that interface or VC. • **input**—Attaches the specified policy map to the input interface or input VC. • **output**—Attaches the specified policy map to the output interface or output VC. • *policy-name*—name of a service policy map (created by the **policy-map** command) to be attached. The name can be a maximum of 40 alphanumeric characters.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

To check the configuration and status of your service policy, use the following commands:

- **show performance monitor history**
- **show performance monitor status**
- **show policy-map ypre performance-monitor interface**

Applying a Cisco Performance Monitor Policy to an Interface Without Using an Existing Flow Policy

Before it can be activated, a Cisco Performance Monitor policy must be applied to at least one interface. To activate a Cisco Performance Monitor policy, perform the following required task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service-policy type performance-monitor inline** {**input** | **output**}
5. **match** {*access-group* {*access-group* | **name** *access-group-name*} | **any** | **class-map***class-map-name* | **cos** *cos-value* | **destination-address mac** address | **discard-class** *class-number* | **dscp** *dscp-value* |

flow {**direction** | **sampler**} | **fr-de** | **fr-dlci** *dlci-number* | **input-interface** *interface-name* | **ip** {**rtp** *starting-port-number port-range* | **precedence** | **dscp**} | **mpls experimental topmost** *number* | **not** *match-criterion*| **packet length** {**max** *maximum-length-value* [**min** *minimum-length-value*] | **min** *minimum-length-value* [**max** *maximum-length-value*]} | **precedence** {*precedence-criteria1* | *precedence-criteria2* | *precedence-criteria3* | *precedence-criteria4*} | **protocol** *protocol-name* | **qos-group** *qos-group-value* | **source-address** *mac address-destination*| **vlan** {*vlan-id* | *vlan-range* | *vlan-combination*}}

6. **flow monitor** {*monitor-name*| **inline**}
7. **record** {*record-name*| **default-rtp**| **default-tcp**}
8. **exporter** *exporter-name*
9. **exit**
10. **monitor metric ip-cbr**
11. **rate layer3** {*byte-rate* {**bps** | **kbps** | **mbps** | **gbps**} | **packet**}
12. **exit**
13. **monitor metric rtp**
14. **clock-rate** {*type-number*| *type-name*} *rate*
15. **max-dropout** *number*
16. **max-reorder** *number*
17. **min-sequential** *number*
18. **ssrc maximum** *number*
19. exit
20. **monitor parameters**
21. **flows** *number*
22. **interval duration** *number*
23. **history** *number*
24. **timeout** *number*
25. **exit**
26. **react** *ID* {**media-stop** | **mrv** | **rtp-jitter-average** | **transport-packets-lost-rate**}
27. **action** {**snmp** | **syslog**}
28. **alarm severity** {**alert** | **critical** | **emergency**| **error** | **info**}
29. **alarm type** {**discrete**| **grouped**{**count** *number* | **percent** *number*}}
30. **threshold value** {**ge** *number* | **gt** *number* | **le** *number* | **lt** *number* | **range** *rng-start rng-end*}
31. **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.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/0`	Specifies an interface and enters interface configuration mode. You can specify an IPv6 interface.

Step 4	**service-policy type performance-monitor inline** {**input** | **output**}
	Example:
	Example:
	`Device(config-if)# service-policy type performance-monitor inline input`
		Attaches a policy map to an input interface or virtual circuit (VC), or an output interface or VC, to be used as the service policy for that interface or VC.
		• **input**—Attaches the specified policy map to the input interface or input VC.
		• **output**—Attaches the specified policy map to the output interface or output VC.

Step 5	**match** {*access-group* {*access-group* | **name** *access-group-name*} | **any** | **class-map***class-map-name* | **cos** *cos-value* | **destination-address mac** address | **discard-class** *class-number* | **dscp** *dscp-value* | **flow** {**direction** | **sampler**} | **fr-de** | **fr-dlci** *dlci-number* | **input-interface** *interface-name* | **ip** {**rtp** *starting-port-number port-range* | **precedence** | **dscp**} | **mpls experimental topmost** *number* | **not** *match-criterion*| **packet length** {**max** *maximum-length-value* [**min** *minimum-length-value*] | **min** *minimum-length-value* [**max** *maximum-length-value*]} | **precedence** {*precedence-criteria1* | *precedence-criteria2* | *precedence-criteria3* | *precedence-criteria4*} | **protocol** *protocol-name* | **qos-group** *qos-group-value* | **source-address** *mac address-destination*| **vlan** {*vlan-id* | *vlan-range* | *vlan-combination*}}
	Example:
	`Device(config-if-spolicy-inline)# match any`
		Specifies the classification criteria.
		For more information and examples, see the *Cisco Media Monitoring Command Reference* .

Step 6	**flow monitor** {*monitor-name*| **inline**}
	Example:
	`Device(config-if-spolicy-inline)# flow monitor inline`
		Specifies an existing flow monitor to associate with a flow policy. If you do not want to use an existing flow monitor, you can use the **inline** option to configure a new one.
		If needed, you can also use the **inline** option to specify a flow record and flow exporter.

Step 7	**record** {*record-name*| **default-rtp**| **default-tcp**}
	Example:
	`Device(config-spolicy-inline-flowmon)# record default-tcp`
		(Optional) If you do not want to use an existing flow monitor, and instead used the **inline** option, use this command to configure a flow record.

	Command or Action	Purpose
Step 8	**exporter** *exporter-name* **Example:** `Device(config-spolicy-inline-flowmon)# exporter exporter-4`	(Optional) If you do not want to use an existing flow monitor, and instead used the **inline** option, use this command to configure a flow exporter.
Step 9	**exit** **Example:** `Device(config-spolicy-inline-flowmon)# exit`	Returns to service-policy inline configuration mode.
Step 10	**monitor metric ip-cbr** **Example:** `Device(config-if-spolicy-inline)# monitor metric ip-cbr`	Enters IP-CBR monitor metric configuration mode.

Step 11	**rate layer3** {*byte-rate* {**bps** | **kbps** | **mbps** | **gbps**} | **packet**}
	Example:
	`Device(config-spolicy-inline-mipcbr)# rate layer3 248 mbps`
		Specifies the rate for monitoring the metrics.
		• *byte-rate*—Data rate in Bps, kBps, mBps, or gBps. The range is 1 to 65535.
		• **packet**—Packet rate in packets per second.

Step 12	**exit** **Example:** `Device(config-spolicy-inline-mipcbr)# exit`	Returns to service-policy inline configuration mode.
Step 13	**monitor metric rtp** **Example:** `Device(config-if-spolicy-inline)# monitor metric rtp`	Enters RTP monitor metric configuration mode.
Step 14	**clock-rate** {*type-number*| *type-name*} *rate* **Example:** `Device(config-spolicy-inline-mrtp)# clock-rate 8 9600`	Specifies the clock rate used to sample RTP video-monitoring metrics. For more information about the clock-type numbers and names, see the *Cisco Media Monitoring Command Reference.* The range for *rate* is 1 kHz to 192 kHz.
Step 15	**max-dropout** *number* **Example:** `Device(config-spolicy-inline-mrtp)# max-dropout 2`	Specifies the maximum number of dropouts allowed when sampling RTP video-monitoring metrics.

	Command or Action	Purpose
Step 16	**max-reorder** *number* **Example:** `Device(config-spolicy-inline-mrtp)# max-reorder 4`	Specifies the maximum number of reorders allowed when sampling RTP video-monitoring metrics.
Step 17	**min-sequential** *number* **Example:** `Device(config-spolicy-inline-mrtp)# min-sequential 2`	Specifies the minimum number of sequential packets required to identify a stream as being an RTP flow.
Step 18	**ssrc maximum** *number* **Example:** `Device(config-spolicy-inline-mrtp)# ssrc maximum 20`	Specifies the maximum number of SSRCs that can be monitored within the same flow. A flow is defined by the protocol, source/destination address, and source/destination port).
Step 19	exit **Example:** `Device(config-spolicy-inline-mrtp)# exit`	Returns to service-policy inline configuration mode.
Step 20	**monitor parameters** **Example:** `Device(config-if-spolicy-inline)# monitor parameters`	Enters monitor parameters configuration mode.
Step 21	**flows** *number* **Example:** `Device(config-spolicy-inline-mparam)# flows 40`	Specifies the maximum number of flows for each monitor cache.
Step 22	**interval duration** *number* **Example:** `Device(config-spolicy-inline-mparam)# interval duration 40`	Specifies the duration of the intervals, in seconds, for collecting monitoring metrics.
Step 23	**history** *number* **Example:** `Device(config-spolicy-inline-mparam)# history 4`	Specifies the number of historical intervals of collected monitoring metrics to display.
Step 24	**timeout** *number* **Example:**	Specifies the number of intervals before a stopped flow is removed from the database.

	Command or Action	Purpose
	`Device(config-spolicy-inline-mparam)# timeout 20`	
Step 25	**exit** **Example:** `Device(config-spolicy-inline-mparam)# exit`	Returns to service-policy inline configuration mode.

Step 26	**react** *ID* {**media-stop** | **mrv** | **rtp-jitter-average** | **transport-packets-lost-rate**}
	Example:
	`Device(config-if-spolicy-inline)# react 6 rtp-jitter-average`
		Enters a mode where you can specify what reaction occurs when a threshold is violated for the following metrics:
		• *ID*— ID for react configuration. Range is 1 to 65535.
		• **media-stop** —No traffic is found for the flow.
		• **mrv** —Ratio calculated by dividing the difference between the actual rate and the expected rate, by the expected rate.
		• **rtp-jitter-average** —Average jitter.
		• **transport-packets-lost-rate** —Ratio calculated by dividing the number of lost packets by the expected packet count.

Step 27	**action** {**snmp** | **syslog**}
	Example:
	`Device(config-spolicy-inline-react)# action syslog`
		Specifies how violations of the thresholds with be reported.

Step 28	**alarm severity** {**alert** | **critical** | **emergency**| **error** | **info**}
	Example:
	`Device(config-spolicy-inline-react)# alarm severity critical`
		Specifies which level of alarm will be reported.

Step 29	**alarm type** {**discrete**| **grouped**{**count** *number* | **percent** *number*}}
	Example:
	`Device(config-pspolicy-inline-react)# alarm severity critical`
		Specifies which types of levels are considered alarms that require reporting.

Step 30	**threshold value** {**ge** *number* | **gt** *number* | **le** *number* | **lt** *number* | **range** *rng-start rng-end*}
	Example:
	`Device(config-spolicy-inline-react)# threshold value ge 20`
		Specifies which types of threshold values are considered alarms that require reporting.
		If no value is set but the application name is configured as a key field, then the system uses the value for the threshold that it finds in the default map. If no value is set and the application name is not configured as a key field, then the default value is used for the threshold.

	Command or Action	Purpose
		If more than one react command is configured for the same policy and class but only one of the react configurations has threshold values set, then the values of the configured react take precedence and the rest of the threshold values are ignored. If more than one react command is configured for the same policy and none of them have the threshold value configured, then the default threshold value is applied for the configuration with the lowest react ID.

Step 31	**end**
	Example:
	`Device(config-spolicy-inline-react)# end`
		Exits the current configuration mode and returns to privileged EXEC mode.

What to do next

To check the configuration and status of your service policy, use the **show performance monitor status** command and **show performance monitor history** command.

Verifying That Cisco Performance Monitor Is Collecting Data

To verify that Cisco Performance Monitor is collecting data, perform the following optional task.

Note Flows are correlated so that if the same policy is applied on the same input and output interface, the **show** command will display a single flow for the input and output interfaces and the interface name and direction for the flow are not displayed.

If no data is being collected, complete the remaining tasks in this section.

Before you begin

The interface to which you applied the input flow monitor must be receiving traffic that meets the criteria defined by the original flow record before you can display the flows in the flow monitor cache.

where *filter* = {**ip** {*source-addr source-prefix* | **any**} {*dst-addr dst-prefix* | **any**} | {**tcp** | **udp**} {*source-addr source-prefix* | **any**} {[**eq**| **lt**| **gt** *number*| **range** *min max*| **ssrc** {*ssrc-number* | **any**} | {{*dst-addr dst-prefix* | **any**} **eq**| **lt**| **gt** *number*| **range** *min max*| **ssrc** {*ssrc-number* | **any**}}

SUMMARY STEPS

1. **enable**
2. **show policy-map type performance-monitor** [**interface** *interface-name*][**class** *class-name*][**input** | **output**]
3. **show performance monitor status** [**interface** *interface name*[*filter*] | **policy** *policy-map-name* **class** *class-map-name*[*filter*]} | *filter*]

4. **show performance monitor history** [**interval**{**all**| *number*[**start** *number*]} | **interface** *interface name*[*filter*] | **policy** *policy-map-name* **class** *class-map-name*[*filter*]} | *filter*]

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show policy-map type performance-monitor** [**interface** *interface-name*][**class** *class-name*][**input** | **output**]

For a description of the fields displayed by this command, see *Cisco Media Monitoring Command Reference*.

The following example shows the output for one flow policy:

Example:

```
  Policy Map type performance-monitor PM-POLICY-4
    Class PM-CLASS-4
      flow monitor PM-MONITOR-4
        record PM-RECORD-4
        exporter PM-EXPORTER-4
      monitor parameters
        interval duration 30
        timeout 10
        history 10
        flows 8000
      monitor metric rtp
        min-sequential 5
        max-dropout 5
        max-reorder 5
        clock-rate default 90000
        ssrc maximum 5
```

Table 3: show policy-map type performance-monitor Field Descriptions

Field	Description
Policy Map type performance-monitor	Name of the Cisco Performance Monitor flow policy.
flow monitor	Name of the Cisco Performance Monitor flow monitor.
record	Name of the Cisco Performance Monitor flow record.
exporter	Name of the Cisco Performance Monitor flow exporter.
monitor parameter	Parameters for the flow policy.
interval duration	The configured duration of the collection interval for the policy.
timeout	The configured amount of time wait for a response when collecting data for the policy.

Field	Description
history	The configured number of historical collections to keep for the policy.
flows	The configured number of flows to collect for the policy.
monitor metric rtp	RTP metrics for the flow policy.
min-sequential	The configured minimum number of packets in a sequence used to classify an RTP flow.
max-dropout	The configured maximum number of packets to ignore ahead of the current packet in terms of sequence number.
max-reorder	The configured maximum number of packets to ignore behind the current packet in terms of sequence number.
clock-rate default	The configured clock rate for the RTP packet timestamp clock that is used to calculate the packet arrival latency.
ssrc maximum	The configured maximum number of SSRCs that can be monitored within the same flow. A flow is defined by the protocol, source/destination address, and source/destination port. The range is from 1 to 50.

Step 3 **show performance monitor status** [**interface** *interface name*[*filter*] | **policy** *policy-map-name* **class** *class-map-name*[*filter*]} | *filter*]

where *filter* = {**ip** {*source-addr source-prefix* | **any**} {*dst-addr dst-prefix* | **any**} | {**tcp** | **udp**} {*source-addr source-prefix* | **any**} {[**eq**| **lt**| **gt** *number*| **range** *min max*| **ssrc** {*ssrc-number* | **any**} | {{*dst-addr dst-prefix* | **any**} **eq**| **lt**| **gt** *number*| **range** *min max*| **ssrc** {*ssrc-number* | **any**}}

This command displays the cumulative statistics for the specified number of most recent intervals. The number of intervals is configured using the **history** command. The default settings for this commands is 10 of the most recent collection intervals. The duration of collection intervals is specified by the **interval duration** command.

To view statistics for other intervals, use the **show performance monitor history** command as described in the next step. For more information about these commands, see the *Cisco Media Monitoring Command Reference*

Step 4 **show performance monitor history** [**interval**{**all**| *number*[**start** *number*]} | **interface** *interface name*[*filter*] | **policy** *policy-map-name* **class** *class-map-name*[*filter*]} | *filter*]

where *filter* = {**ip** {*source-addr source-prefix* | **any**} {*dst-addr dst-prefix* | **any**} | {**tcp** | **udp**} {*source-addr source-prefix* | **any**} {[**eq**| **lt**| **gt** *number*| **range** *min max*| **ssrc** {*ssrc-number* | **any**} | {{*dst-addr dst-prefix* | **any**} **eq**| **lt**| **gt** *number*| **range** *min max*| **ssrc** {*ssrc-number* | **any**}}

This command displays the statistics collected by Cisco Performance Monitor during any or all intervals, including the current one. The duration of collection intervals is specified by the **interval duration** command.

For more information about this command, see the *Cisco Media Monitoring Command Reference.*

The following example shows the output for the **show performance monitor history** command**:**

Note If the same policy is applied on the same input and output interface, the display shows a single flow for the input and output interfaces and the interface name and direction for the flow are not displayed.

Example:

```
Codes: *   - field is not configurable under flow record
       NA  - field is not applicable for configured parameters
Match: ipv4 source address = 21.21.21.1, ipv4 destination address = 1.1.1.1,
transport source-port = 10240, transport destination-port = 80, ip protocol = 6,
 Policy: RTP_POL, Class: RTP_CLASS

 start time                                 14:57:34
                                            ============
 *history bucket number                    : 1
 routing forwarding-status                 : Unknown
 transport packets expected counter        : NA
 transport packets lost counter            : NA
 transport round-trip-time          (msec) : 4
 transport round-trip-time sum      (msec) : 8
 transport round-trip-time samples         : 2
 transport event packet-loss counter       : 0
 interface input                           : Null
 interface output                          : Null
 counter bytes                             : 8490
 counter packets                           : 180
 counter bytes rate                        : 94
 counter client bytes                     : 80
 counter server bytes                     : 200
 counter client packets                   : 6
 counter server packets                   : 6
 transport tcp window-size minimum        : 1000
 transport tcp window-size maximum        : 2000
 transport tcp window-size average        : 1500
 transport tcp maximum-segment-size       : 0
 application media bytes counter           : 1270
 application media bytes rate              : 14
 application media packets counter         : 180
 application media event                   : Stop
 monitor event                             : false

 [data set,id=257] Global session ID|Multi-party session ID|
 [data] 11                      |22
```

Table 4: show performance monitor status and show performance-monitor history Field Descriptions

Field	Description
history bucket number	Number of the bucket of historical data collected.

Field	Description
routing forwarding-status reason	

Field	Description
	Forwarding status is encoded using eight bits with the two most significant bits giving the status and the six remaining bits giving the reason code. Status is either unknown (00), Forwarded (10), Dropped (10) or Consumed (11). The following list shows the forwarding status values for each status category. **Unknown** • 0 **Forwarded** • Unknown 64 • Forwarded Fragmented 65 • Forwarded not Fragmented 66 **Dropped** • Unknown 128, • Drop ACL Deny 129, • Drop ACL drop 130, • Drop Unroutable 131, • Drop Adjacency 132, • Drop Fragmentation & DF set 133, • Drop Bad header checksum 134, • Drop Bad total Length 135, • Drop Bad Header Length 136, • Drop bad TTL 137, • Drop Policer 138, • Drop WRED 139, • Drop RPF 140, • Drop For us 141, • Drop Bad output interface 142, • Drop Hardware 143, **Consumed** • Unknown 192, • Terminate Punt Adjacency 193, • Terminate Incomplete Adjacency 194,

Field	Description
	• Terminate For us 195
transport packets expected counter	Number of packets expected.
transport packets lost counter	Number of packets lost.
transport round-trip-time (msec)	Number of milliseconds required to complete a round trip.
transport round-trip-time sum (msec)	Total number of milliseconds required to complete a round trip for all samples.
transport round-trip-time samples	Total number of samples used to calculate a round trip times
transport event packet-loss counter	Number of loss events (number of contiguous sets of lost packets).
interface input	Incoming interface index.
interface output	Outgoing interface index.
counter bytes	Total number of bytes collected for all flows.
counter packets	Total number of IP packets sent for all flows.
counter bytes rate	Average number of packets or bits (as configured) processed by the monitoring system per second during the monitoring interval for all flows.
counter client bytes	Number of bytes sent by the client.
counter server bytes	Number of bytes sent by the server.
counter client packets	Number of packets sent by the client.
counter servers packets	Number of packets sent by the server.
transport tcp window-size-maximum	Maximum size of the TCP window.
transport tcp window-size-minimum	Minimum size of the TCP window.
transport tcp window-size-average	Average size of the TCP window.
transport tcp maximum-segment-size	Maximum TCP segment size.
application media bytes counter	Number of IP bytes from by media applications received for a specific media stream.
application media bytes rate	Average media bit rate (bps) for all flows during the monitoring interval.
application media packets counter	Number of IP packets produced from media applications received for a specific media stream.
application media event	Bit 1 is not used. Bit 2 indicates that no media application packets were seen, in other words, a Media Stop Event occurred.

Field	Description
monitor event	Bit 1 indicates that one of the thresholds specified by a react statement for the flow was crossed at least once in the monitoring interval. Bit 2 indicates that there was a loss-of-confidence in measurement.

Displaying Option Tables.

You can view the mapping contained in the various option table by using the following **show** command .

SUMMARY STEPS

1. **enable**
2. **show metadata** {**application attributes** | **application table** | **exporter stats** | **interface table** | **metadata version table** | **sampler table** | **vrf table**}

DETAILED STEPS

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

Step 2	**show metadata** {**application attributes** | **application table** | **exporter stats** | **interface table** | **metadata version table** | **sampler table** | **vrf table**}
	Example:
		The following example shows how to display the mapping of the application ID to the application name by using the **show metadata application table** command :
		`ID        Name                   Vendor`
		`Version`
		`100673296  webex-audio           -           -`
		`100673297  webex-video           -           -`

Displaying Information Specific to the Catalyst 6500 Platform

To display or clear information for the Feature Manager and other functionality specific to the Catalyst 6500 platform, perform the following optional task.

SUMMARY STEPS

1. **enable**
2. **clear fm performance-monitor counters**
3. **debug fm performance-monitor** {**all** | **dynamic** | **event** | **unusual** | **verbose** | **vmr**}
4. **platform performance-monitor rate-limit pps** *number*
5. **show platform software feature-manager performance-monitor** {**all** | **counters** | **interface** *interface-type interface-number* | **rdt-indices** }

6. **show platform software feature-manager tcam dynamic performance-monitor** {**handle ip** *ip-address* | **interface** *interface-type interface-number* }
7. **show platform hardware acl entry interface** *interface-type interface-number* **security** {**in** | **out** } {**ip** | **ipv6** } [**detail**]
8. **show platform software ccm interface** *interface-type interface-number* **security** {**interface** *interface-type interface-number* | **class-group** *class-group-ID* }

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **clear fm performance-monitor counters**

The **clearfm performance-monitor counters** command clears counters for the Performance Monitor component of Feature Monitor.

Example:

```
Device# clear fm performance-monitor counters
Device#
```

Step 3 **debug fm performance-monitor** {**all** | **dynamic** | **event** | **unusual** | **verbose** | **vmr**}

This command enables all levels of debugging for the Performance Monitor component of Feature Manager.

Example:

```
Device# debug fm performance-monitor all
Device#
```

Step 4 **platform performance-monitor rate-limit pps** *number*

This command sets the rate limit for the Performance Monitor component of Feature Monitor.

Example:

```
Device# platform performance-monitor rate-limit pps 2000
Device#
```

Step 5 **show platform software feature-manager performance-monitor** {**all** | **counters** | **interface** *interface-type interface-number* | **rdt-indices** }

This command displays information about the Performance Monitor component of Feature Manager.

Example:

```
Device# show platform software feature-manager performance-monitor all
Device#

Interface: FastEthernet2/3
```

```
Policy: video-flow-test          Group ID: A0000001
-----------------------------------------------------------------------------
Feature: VM Ingress L3
=============================================================================
DPort  - Destination Port  SPort  - Source Port       Pro    - Protocol
RFTCM  - R-Recirc. Flag    MRLCS  - M-Multicast Flag  Res    - VMR Result
       - F-Fragment flag          - R-Reflexive flag  Prec   - Drop Precedence
       - T-Trailing Fragments     - L-Layer 3 only    GrpId  - Qos Group Id
       - C-From CPU               - C-Capture Flag    Adj.   - Adj. Index
       - M-L2 Lookup Miss         - S-RPF suppress    Pid    - NF Profile Index
+-----+--+---------------+-----------------+------+------+----+--------+-----+-------+----+--------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+-----+--+---------------+-----------------+------+------+----+--------+-----+-------+----+--------+


     1 V       224.0.0.0         0.0.0.0                  0          0      0       -----         0
       -----
       M       240.0.0.0         0.0.0.0                   0          0      0         00000    0
       0
      PERMIT_RESULT

     2 V         0.0.0.0          0.0.0.0                  0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                   0          0      0         00000     0
        0
      L3_DENY_RESULT

+-----+--+---------------+-----------------+------+------+----+--------+-----+-------+----+--------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+-----+--+---------------+-----------------+------+------+----+--------+-----+-------+----+--------+


     1 V         0.0.0.0          10.10.10.0            0          0      17        -----        0
     ---C-
       M          0.0.0.0          255.255.255 0      0          0      255      00000     0
 0
      PERMIT_RESULT

     2 V         0.0.0.0          10.10.20.0            0          0      17        -----        0
     ---C-
       M          0.0.0.0          255.255.255 0      0          0      255      00000     0
 0
      PERMIT_RESULT

     3 V         0.0.0.0          0.0.0.0                  0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                   0          0      0         00000     0
        0
L3_DENY_RESULT

+-----+--+---------------+-----------------+------+------+----+--------+-----+-------+----+--------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+-----+--+---------------+-----------------+------+------+----+--------+-----+-------+----+--------+


     1 V         0.0.0.0          0.0.0.0                  0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                   0          0      0         00000     0
        0
      PERMIT_RESULT
```

```
Interface: FastEthernet2/3
Policy: video-flow-test          Group ID: A0000001
-----------------------------------------------------------------------------
Feature: VM Egress L3
=============================================================================
+----+-+----------------+-----------------+------+--------+---+-------+----+---------+----+--------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+----+-+----------------+-----------------+------+--------+---+-------+----+---------+----+--------+


     1 V         0.0.0.0          0.0.0.0                 0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                  0          0      0         00000     0
        0
PERMIT_RESULT

     2 V         0.0.0.0          0.0.0.0                 0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                  0          0      0         00000     0
        0
      L3_DENY_RESULT

+----+-+----------------+-----------------+------+--------+---+-------+----+---------+----+--------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+----+-+----------------+-----------------+------+--------+---+-------+----+---------+----+--------+


     1 V         0.0.0.0          10.10.10.0            0          0      17        -----        0
     -----
       M          0.0.0.0          255.255.255 0      0          0      255      00000     0
 0
      PERMIT_RESULT Adjacency: 0x5512D8F4

     2 V         0.0.0.0          10.10.20.0            0          0      17        -----        0
     -----
       M          0.0.0.0          255.255.255 0      0          0      255      00000     0
 0
      PERMIT_RESULT Adjacency: 0x5512D8F4

     3 V         0.0.0.0          0.0.0.0                 0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                  0          0      0         00000     0
        0
       L3_DENY_RESULT

+----+-+----------------+-----------------+------+--------+---+-------+----+---------+----+--------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+----+-+----------------+-----------------+------+--------+---+-------+----+---------+----+--------+


     3 V         0.0.0.0          0.0.0.0                 0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                  0          0      0         00000     0
        0
     PERMIT_RESULT Adjacency: 0x5512D8F4


Adjacency: 0x5512D8F4
        FeatureId: 0x84  AdjId: 0xFFFFFFFF Flags: RecirculationAdj|
        Cause: 0x0 Priority: 0xC Device#

Interface: FastEthernet2/3
Policy: video-flow-test          Group ID: A0000001
-----------------------------------------------------------------------------
Feature: VM Ingress L3
=============================================================================
DPort  - Destination Port  SPort  - Source Port       Pro    - Protocol
RFTCM  - R-Recirc. Flag    MRLCS  - M-Multicast Flag  Res    - VMR Result
       - F-Fragment flag          - R-Reflexive flag  Prec   - Drop Precedence
       - T-Trailing Fragments     - L-Layer 3 only    GrpId  - Qos Group Id
       - C-From CPU               - C-Capture Flag    Adj.   - Adj. Index
       - M-L2 Lookup Miss         - S-RPF suppress    Pid    - NF Profile Index
+-----+--+-----------------+------------------+-------+-------+----+---------+------+--------+-----+--------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+-----+--+-----------------+------------------+-------+-------+----+---------+------+--------+-----+--------+


     1 V       224.0.0.0         0.0.0.0                  0          0      0         -----       0
       -----
       M       240.0.0.0         0.0.0.0                   0          0      0         00000    0
       0
      PERMIT_RESULT

     2 V         0.0.0.0          0.0.0.0                  0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                   0          0      0         00000     0
        0
      L3_DENY_RESULT

+-----+--+-----------------+------------------+-------+-------+----+---------+------+--------+-----+--------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+-----+--+-----------------+------------------+-------+-------+----+---------+------+--------+-----+--------+


     1 V         0.0.0.0          10.10.10.0           0          0      17        -----        0
     ---C-
       M          0.0.0.0          255.255.255 0      0          0      255      00000     0
 0
      PERMIT_RESULT

     2 V         0.0.0.0          10.10.20.0           0          0      17        -----        0
     ---C-
       M          0.0.0.0          255.255.255 0      0          0      255      00000     0
 0
      PERMIT_RESULT

     3 V         0.0.0.0          0.0.0.0                  0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                   0          0      0         00000     0
        0
L3_DENY_RESULT

+-----+--+-----------------+------------------+-------+-------+----+---------+------+--------+-----+--------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+-----+--+-----------------+------------------+-------+-------+----+---------+------+--------+-----+--------+


     1 V         0.0.0.0          0.0.0.0                  0          0      0         -----
 0      -----
```

```
       M          0.0.0.0         0.0.0.0                   0          0      0         00000     0
        0
      PERMIT_RESULT


Interface: FastEthernet2/3
Policy: video-flow-test          Group ID: A0000001
-----------------------------------------------------------------------------
Feature: VM Egress L3
=============================================================================
+----+--+---------------+-----------------+------+-------+---+--------+------+-------+-----+-------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+----+--+---------------+-----------------+------+-------+---+--------+------+-------+-----+-------+


     1 V         0.0.0.0          0.0.0.0                  0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                   0          0      0         00000     0
        0
PERMIT_RESULT

     2 V         0.0.0.0          0.0.0.0                  0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                   0          0      0         00000     0
        0
      L3_DENY_RESULT

+----+--+---------------+-----------------+------+-------+---+--------+------+-------+-----+-------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+----+--+---------------+-----------------+------+-------+---+--------+------+-------+-----+-------+


     1 V         0.0.0.0          10.10.10.0            0          0      17        -----        0
     -----
       M          0.0.0.0          255.255.255 0      0          0      255      00000     0
 0
      PERMIT_RESULT Adjacency: 0x5512D8F4

     2 V         0.0.0.0          10.10.20.0            0          0      17        -----        0
     -----
       M          0.0.0.0          255.255.255 0      0          0      255      00000     0
 0
      PERMIT_RESULT Adjacency: 0x5512D8F4

     3 V         0.0.0.0          0.0.0.0                  0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                   0          0      0         00000     0
        0
       L3_DENY_RESULT

+----+--+---------------+-----------------+------+-------+---+--------+------+-------+-----+-------+
| Indx | T |   Dest Ip Addr | Source Ip Addr | DPort | SPort | Pro | RFTCM | Prec | MRLCS | Pid |
Stats Id|
+----+--+---------------+-----------------+------+-------+---+--------+------+-------+-----+-------+


     3 V         0.0.0.0          0.0.0.0                  0          0      0         -----
 0      -----
       M          0.0.0.0         0.0.0.0                   0          0      0         00000     0
        0
     PERMIT_RESULT Adjacency: 0x5512D8F4
```

```
Adjacency: 0x5512D8F4
        FeatureId: 0x84  AdjId: 0xFFFFFFFF Flags: RecirculationAdj|

        Cause: 0x0 Priority: 0xC
```

Step 6 **show platform software feature-manager tcam dynamic performance-monitor** {**handle ip** *ip-address* | **interface** *interface-type interface-number* }

This command displays information about dynamic and static policies for a specific host.

Example:

```
Device# show platform software feature-manager tcam dynamic performance-monitor handle ip 10.1.1.0
------------------------------------------------------------------------------
HANDLE                      Feature ID   No of entries     MD5
------------------------------------------------------------------------------
10.1.1.0             VM Ingress L3                         2
```

Step 7 **show platform hardware acl entry interface** *interface-type interface-number* **security** {**in** | **out** } {**ip** | **ipv6** } [**detail**]

This command displays inbound access control list (ACL) entries for IP on an interface.

Example:

```
Device# show platform hardware acl entry interface fastEthernet 1/1 security in ip detail

mls_if_index:2000400A dir:0 feature:0 proto:0


pass#0 features
UAPRSF: U-urg, A-ack, P-psh, R-rst, S-syn, F-fin
MLGFI: M-mpls_plus_ip_pkt, L-L4_hdr_vld, G-gpid_present,F-global_fmt_match, I-ife/ofe
's' means set; 'u' means unset; '-' means don't care

INDEX  LABEL FS ACOS    AS     IP_SA      SRC_PORT       IP_DA      DST_PORT        F FF       L4PROT


TCP-F:UAPRSF MLGFI OtherL4OPs                                  RSLT                   CNT

fno:0

tcam:B, bank:0, prot:0    Aces

I  V  16375   2049  0    0     0         0.0.0.0               -         0.0.0.0               - 0
 0      0      -       -----          -
0x0000000800000038            10331192<-
I  M  16375 0x1FFF  0 0x00 0x000         0.0.0.0               -         0.0.0.0               - 0
 0    0x0
```

Step 8 **show platform software ccm interface** *interface-type interface-number* **security** {**interface** *interface-type interface-number* | **class-group** *class-group-ID* }

This command displays information about ternary content addressable memory (TCAM) Cisco CallManager (CCM) entries on an interface.

Example:

```
Device# show platform software ccm interface fastEthernet 2/3 in
```

```
 Target-Class : id 0xA0000000, dir CCM_INPUT, if_type 1, if_info 0x14823998

 Class-Group List: 0xA0000001
b1-cs217#

b1-cs217#sh platform software ccm interface fastEthernet 2/3 out

 Target-Class : id 0xA0000002, dir CCM_OUTPUT, if_type 1, if_info 0x14823998

 Class-Group List: 0xA0000001
```

This command displays information about ternary content addressable memory (TCAM) Cisco CallManager (CCM) entries for a class group

Example:

```
Device# show platform software ccm class-group A0000001
 Class-group    : video-flow-test, id 0xA0000001
 Target input   : 0xA0000000
 Target Output  : 0xA0000002
        Class   : video-flow, id 0xA98681, type 1
                Filter          : type MATCH_NUMBERED_ACCESS_GROUP, id 0xF0000002
                Filter params   : ACL Index: 101 Linktype: 7

                Feature         : PERFORMANCE_MONITOR
                Params          :
                  Feature Object : 0x54224218
                    Name           :
                    Meter context  : 0x54264440
                    Sibling        : 0x0
                    Dynamic        : FALSE
                  Feature Object : 0x54221170
                    Name           :
                    Meter context  : 0x54263858
                    Sibling        : 0x0
                    Dynamic        : FALSE
                Intf List       : 0xA0000000  0xA0000002
        Class   : class-default, id 0xADA3F1, type 39
                Filter          : type MATCH_ANY, id 0xF0000003
                Filter params   : any

                Feature         : FEATURE_EMPTY
                Params          :
                  Feature Object : 0x1741629C
                    Name           :
                    Meter context  : 0x0
                    Sibling        : 0x0
                    Dynamic        : FALSE
                Intf List       : 0xA0000000  0xA0000002
```

Displaying the Performance Monitor Cache and Clients

To display the cache and the clients for Cisco Performance Monitor, perform the following optional task.

SUMMARY STEPS

1. **enable**

2. **show performance monitor cache** [**policy** *policy-map-name* **class** *class-map-name*][**interface** *interface name*]
3. **show performance monitor clients detail all**

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show performance monitor cache** [**policy** *policy-map-name* **class** *class-map-name*][**interface** *interface name*]

Example:

```
MMON Metering Layer Stats:
  static pkt cnt: 3049
  static cce sb cnt: 57
  dynamic pkt cnt: 0
  Cache type:                            Permanent
  Cache size:                                 2000
  Current entries:                               8
  High Watermark:                                9
  Flows added:                                   9
  Updates sent            (  1800 secs)          0
IPV4 SRC ADDR    IPV4 DST ADDR    IP PROT  TRNS SRC PORT  TRNS DST PORT
ipv4 ttl ipv4 ttl min ipv4 ttl max  ipv4 dscp bytes long perm pktslong perm  user space vm
==========================================================================================
10.1.1.1         10.1.2.3              17           4000           1967
0             0             0  0x00                        80
1 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000
10.1.1.1         10.1.2.3              17           6000           1967
0             0             0  0x00                        80
1  0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000
10.1.1.1         10.1.2.3              17           4000           2000
0             0             0  0x00                        44
1  0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000
10.1.1.1         10.1.2.3               6           6000           3000
```

```
0             0             0  0x00                         84
2  0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000
10.1.1.1         10.1.2.3              17           1967           6001
0             0             0  0x00                         36
1  0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000
10.1.1.1         10.1.2.3              17           1967           4001
0             0             0  0x00                         36
1  0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000
10.1.1.1         10.1.2.3               6           3001           6001
0             0             0  0x00                        124
3  0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000
10.1.1.1         10.1.2.3              17           2001           4001
0             0             0  0x00                         44
1  0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000
```

Step 3 **show performance monitor clients detail all**

Example:

```
Client name for ID 1 : Mediatrace-131419052
   Type: Mediatrace
   Age: 443 seconds
   Monitor Object: _MMON_DYN_-class-map-69
        Flow spec: (dvmc-acl#47) 10.10.130.2 1000 10.10.132.2 2000 17
        monitor parameters
                interval duration 60
                timeout 2
                history 1
                flows 100
        monitor metric rtp
                min-sequential 10
                max-dropout 5
                max-reorder 5
                clock-rate 112 90000
                clock-rate default 90000
                ssrc maximum 20
        monitor metric ip-cbr
                rate layer3 packet 20
        Flow record: dvmc_fnf_fdef_47
                Key fields:
                        ipv4 source address
                        ipv4 destination address
                        transport source-port
                        transport destination-port
                        ip protocol
                Non-key fields:
                        monitor event
                        application media event
                        routing forwarding-status
                        ip dscp
                        ip ttl
                        counter bytes rate
                        application media bytes rate
                        transport rtp jitter mean
                        transport packets lost counter
                        transport packets expected counter
                        transport event packet-loss counter
                        transport packets lost rate
                        timestamp interval
                        counter packets dropped
                        counter bytes
                        counter packets
                        application media bytes counter
                        application media packets counter
        Monitor point: _MMON_DYN_-policy-map-70 GigabitEthernet0/3 output
        Classification Statistic:
                matched packet: 545790
                matched byte: 64403220
```

Displaying the Clock Rate for Cisco Performance Monitor Classes

To display the clock rate for one or more classes, perform the following optional task.

SUMMARY STEPS

1. **enable**
2. **show performance monitor clock rate** [**policy** *policy-map-name* **class** *class-map-name*]

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show performance monitor clock rate** [**policy** *policy-map-name* **class** *class-map-name*]

If no class name is specified, information for all classes are displayed.

Example:

```
Device# show performance monitor clock rate policy all-apps class telepresence-CS4
Load for five secs: 6%/2%; one minute: 5%; five minutes: 5% Time source is NTP, 17:41:35.508 EST Wed
 Feb 16 2011
RTP clock rate for Policy: all-apps, Class: telepresence-CS4
     Payload type     Clock rate(Hz)
     pcmu    (0  )     8000
     gsm     (3  )     8000
     g723    (4  )     8000
     dvi4    (5  )     8000
     dvi4-2  (6  )     16000
     lpc     (7  )     8000
     pcma    (8  )     8000
     g722    (9  )     8000
     l16-2   (10 )     44100
     l16     (11 )     44100
     qcelp   (12 )     8000
     cn      (13 )     8000
     mpa     (14 )     90000
     g728    (15 )     8000
     dvi4-3  (16 )     11025
     dvi4-4  (17 )     22050
     g729    (18 )     8000
     celb    (25 )     90000
     jpeg    (26 )     90000
     nv      (28 )     90000
     h261    (31 )     90000
     mpv     (32 )     90000
     mp2t    (33 )     90000
     h263    (34 )     90000
             (96 )     48000
             (112)     90000
     default           90000
```

Displaying the Current Status of a Flow Monitor

To display the current status of a flow monitor, perform the following optional task.

Before you begin

The interface to which you applied the input flow monitor must be receiving traffic that meets the criteria defined by the original flow record before you can display the flows in the flow monitor cache.

SUMMARY STEPS

1. **enable**
2. **show flow monitor type performance-monitor**

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show flow monitor type performance-monitor**

The **show flow monitor type performance-monitor** command shows the current status of the flow monitor that you specify.

Example:

```
Device# show flow monitor type performance-monitor
Flow Monitor type performance-monitor monitor-4:
  Description:           User defined
  Flow Record:           record-4
  Flow Exporter:         exporter-4
  No. of Inactive Users: 0
  No. of Active Users:   0
```

Verifying the Flow Monitor Configuration

To verify the configuration commands that you entered, perform the following optional task.

Before you begin

The interface to which you applied the input flow monitor must be receiving traffic that meets the criteria defined by the original flow record before you can display the flows in the flow monitor cache.

SUMMARY STEPS

1. **enable**
2. **show running-config flow monitor**

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show running-config flow monitor**

The **show running-config flow monitor** command shows the configuration commands of the flow monitor that you specify.

Example:

```
Device# show running-config flow monitor
Current configuration:
!
flow monitor FLOW-MONITOR-1
 description Used for basic IPv4 traffic analysis
 record netflow ipv4 original-input
!
!
flow monitor FLOW-MONITOR-2
 description Used for basic IPv6 traffic analysis
 record netflow ipv6 original-input
!
```

Verifying That Cisco IOS Flexible NetFlow and Cisco Performance Monitor Is Enabled on an Interface

To verify that Flexible NetFlow and Cisco Performance Monitor is enabled on an interface, perform the following optional task.

SUMMARY STEPS

1. **enable**
2. **show flow interface** *type number*

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Router> enable
Router#
```

Step 2 **show flow interface** *type number*

The **show flow interface** command verifies that Flexible NetFlow and Cisco Performance Monitor is enabled on an interface.

Example:

```
Router# show flow interface ethernet 0/0
Interface Ethernet0/0
  FNF:  monitor:         FLOW-MONITOR-1
        direction:       Input
        traffic(ip):     on
  FNF:  monitor:         FLOW-MONITOR-2
        direction:       Input
        traffic(ipv6):   on
```

Displaying the Flow Monitor Cache

To display the data in the flow monitor cache, perform the following optional task.

Before you begin

The interface to which you applied the input flow monitor must be receiving traffic that meets the criteria defined by the original flow record before you can display the flow data in the flow monitor cache.

SUMMARY STEPS

1. **enable**
2. **show flow monitor name** *monitor-name* **cache format record**

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show flow monitor name** *monitor-name* **cache format record**

The **show flow monitor name** *monitor-name* **cache format record** command string displays the status, statistics, and the flow data in the cache for a flow monitor.

Example:

```
Device# show flow monitor name FLOW-MONITOR-1 cache format record
Cache type:                            Normal
  Cache size:                              4096
  Current entries:                            8
  High Watermark:                             8
  Flows added:                               24
  Flows aged:                                16
    - Active timeout   (  1800 secs)          0
    - Inactive timeout (    15 secs)         16
    - Event aged                              0
    - Watermark aged                          0
    - Emergency aged                          0
IPV4 SOURCE ADDRESS:       10.251.10.1
IPV4 DESTINATION ADDRESS:  172.16.10.2
TRNS SOURCE PORT:          0
TRNS DESTINATION PORT:     2048
INTERFACE INPUT:           Et0/0
FLOW SAMPLER ID:           0
IP TOS:                    0x00
IP PROTOCOL:               1
ip source as:              0
ip destination as:         0
ipv4 next hop address:     172.16.7.2
ipv4 source mask:          /0
ipv4 destination mask:     /24
tcp flags:                 0x00
```

```
interface output:          Et1/0
counter bytes:             733500
counter packets:           489
timestamp first:           720892
timestamp last:            975032
.
.
.
IPV4 SOURCE ADDRESS:       172.16.6.1
IPV4 DESTINATION ADDRESS:  224.0.0.9
TRNS SOURCE PORT:          520
TRNS DESTINATION PORT:     520
INTERFACE INPUT:           Et0/0
FLOW SAMPLER ID:           0
IP TOS:                    0xC0
IP PROTOCOL:               17
ip source as:              0
ip destination as:         0
ipv4 next hop address:     0.0.0.0
ipv4 source mask:          /24
ipv4 destination mask:     /0
tcp flags:                 0x00
interface output:          Null
counter bytes:             52
counter packets:           1
timestamp first:           973804
timestamp last:            973804
Device# show flow monitor name FLOW-MONITOR-2 cache format record
Cache type:                            Normal
  Cache size:                            4096
  Current entries:                          6
  High Watermark:                           8
  Flows added:                           1048
  Flows aged:                            1042
    - Active timeout   (  1800 secs)       11
    - Inactive timeout (    15 secs)     1031
    - Event aged                            0
    - Watermark aged                        0
    - Emergency aged                        0
IPV6 FLOW LABEL:           0
IPV6 EXTENSION MAP:        0x00000040
IPV6 SOURCE ADDRESS:       2001:DB8:1:ABCD::1
IPV6 DESTINATION ADDRESS:  2001:DB8:4:ABCD::2
TRNS SOURCE PORT:          3000
TRNS DESTINATION PORT:     55
INTERFACE INPUT:           Et0/0
FLOW DIRECTION:            Input
FLOW SAMPLER ID:           0
IP PROTOCOL:               17
IP TOS:                    0x00
ip source as:              0
ip destination as:         0
ipv6 next hop address:     ::
ipv6 source mask:          /48
ipv6 destination mask:     /0
tcp flags:                 0x00
interface output:          Null
counter bytes:             521192
counter packets:           9307
timestamp first:           9899684
timestamp last:            11660744
.
.
.
```

```
IPV6 FLOW LABEL:            0
IPV6 EXTENSION MAP:         0x00000000
IPV6 SOURCE ADDRESS:        FE80::A8AA:BBFF:FEBB:CC03
IPV6 DESTINATION ADDRESS:  FF02::9
TRNS SOURCE PORT:           521
TRNS DESTINATION PORT:      521
INTERFACE INPUT:            Et0/0
FLOW DIRECTION:             Input
FLOW SAMPLER ID:            0
IP PROTOCOL:                17
IP TOS:                     0xE0
ip source as:               0
ip destination as:          0
ipv6 next hop address:      ::
ipv6 source mask:           /10
ipv6 destination mask:      /0
tcp flags:                  0x00
interface output:           Null
counter bytes:              92
counter packets:            1
timestamp first:            11653832
timestamp last:             11653832
```

Displaying the Current Status of a Flow Exporter

To display the current status of a flow exporter, perform the following optional task.

SUMMARY STEPS

1. **enable**
2. **show flow exporter** [*exporter-name*]

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show flow exporter** [*exporter-name*]

The **show flow exporter** command shows the current status of the flow exporter that you specify.

Example:

```
Device# show flow exporter EXPORTER-1
Flow Exporter EXPORTER-1:
  Description:              Exports to Chicago datacenter
  Transport Configuration:
    Destination IP address: 172.16.10.2
    Source IP address:      172.16.7.1
    Transport Protocol:     UDP
    Destination Port:        65
    Source Port:             56041
    DSCP:                    0x0
    TTL:                     255
```

Verifying the Flow Exporter Configuration

To verify the configuration commands that you entered to configure the flow exporter, perform the following optional task.

SUMMARY STEPS

1. **enable**
2. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show running-config flow exporter** *exporter-name*

The **show running-config flow exporter** command shows the configuration commands of the flow exporter that you specify.

Example:

```
Device# show running-config flow exporter EXPORTER-1
Building configuration...
!
flow exporter EXPORTER-1
 description Exports to datacenter
 destination 172.16.10.2
 transport udp 65
!
```

Enabling Debugging

To enable debugging for Cisco Performance Monitor, perform the following optional task in privileged EXEC mode.

SUMMARY STEPS

1. **debug performance monitor** {**database** | **dynamic** | **event** | **export** | **flow-monitor** | **metering** | **provision** | **sibling** | **snmp** | **tca** | **timer**}

DETAILED STEPS

debug performance monitor {**database** | **dynamic** | **event** | **export** | **flow-monitor** | **metering** | **provision** | **sibling** | **snmp** | **tca** | **timer**}

The **debug performance monitor** command enables debugging for the following performance monitor components:

- Flow database
- Dynamic monitoring
- Performance events
- Exporting
- Flow monitors
- Metering layer
- Provisioning
- Sibling management
- SNMP
- TCA
- Timers

The following example shows how to enable debugging for dynamic monitoring:

Example:

```
Device# debug performance monitor dynamic
```

Configuration Example for Cisco Performance Monitor

Example Monitor for Lost RTP Packets and RTP Jitter

This example show a configuration that monitors the number of lost RTP packets, the amount of RTP jitter, and other basic statistics for the **gig1** interface. In this example, Cisco Performance Monitor is also configured to make an entry in the syslog when the any of the following events occur on the interface:

- The percentage of lost RTP packets is between 5 percent and 9 percent.
- The percentage of lost RTP packets is greater than 10 percent.

- A media stop event has occurred.

```
! Set the filter spec for the flows to monitor.
access-list 101 ip permit host 10.10.2.20 any
! Use the flow record to define the flow keys and metric to collect.
flow record type performance-monitor video-monitor-record
 match ipv4 source
 match ipv4 destination
 match transport source-port
 match transport destination-port
 match rtp ssrc
 collect timestamp
 collect counter byte
 collect counter packet
 collect mse
 collect media-error
 collect counter rtp interval-jitter
 collect counter rtp packet lost
 collect counter rtp lost event
! Set the exporting server. The export message format is based on FNFv.9.
flow export video-nms-server
 export-protocol netflow-v9
 destination cisco-video-management
 transport udp 32001
! Set the flow filter in the class-map.
class-map match-all video-class
 access-group  ipv4 101
! Set the policy map with the type performance-monitor for video monitor.
policy-map type performance-monitor video-monitor
 ! Set the video monitor actions.
 class video-class
  ! Specify where the metric data is being exported to.
  export  flow video-nms-server
  flow monitor inline
   record video-monitor-record
! Set the monitoring modeling parameters.
monitor parameters
 ! Set the measurement timeout to 10 secs.
 interval duration 10
 ! Set the timeout to 10 minutes.
 timeout 10
 ! Specify that 30 flow intervals can be kept in performance database.
 history 30
 priority 7
 ! Set rtp flow verification criteria.
 monitor metric rtp
 ! Configure a RTP flow criteria: at least 10 packets in sequence.
 min-sequential   10
 ! Ignore packets that are more than 5 packet ahead in terms of seq  number.  max-dropout
 5
 ! Ignore packets that are more than 5 packets behind in terms of seq  number.
 max-reorder 5
 ! Set the clock rate frequency for rtp packet timestamp clock.
 clock-rate 89000
 ! Set the maximum number of ssrc allowed within this class.
 ssrc maximum  100
 ! Set TCA for alarm.
 react 100 transport-packets-lost-rate
  description critical  TCA
  ! Set the threshold to greater than 10%.
  threshold gt 10
  ! Set the threshold to the average number based on the last five intervals.
  threshold type average 5
  action  syslog
  alarm severity critical
 react 110 transport-packets-lost-rate
  description medium TCA
  ! Set the threshold to between 5% and 9% of packet lost.
  threshold range gt 5 le 9
  threshold type average 10
  action  syslog
  alarm type grouped percent 30
 react 3000 media-stop
  action syslog
  alarm severity critical
  alarm type grouped percent 30

interface gig1
 service-policy type performance-monitor video-mon in
```

Where to Go Next

For more information about configuring the products in the Medianet product family, see the other chapter in this guide or see the *Cisco Media Monitoring Configuration Guide*.

Additional References

Related Documents

Related Topic	Document Title
Design, configuration, and troubleshooting resources for Performance Monitor and other Cisco Medianet products, including a Quick Start Guide and Deployment Guide.	See the Cisco Medianet Knowledge Base Portal, located at http://www.cisco.com/web/solutions/medianet/knowledgebase/index.html
IP addressing commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco Media Monitoring Command Reference*
Configuration commands for Flexible NetFlow	*Cisco IOS Flexible NetFlow Command Reference*
Overview of Flexible NetFlow	“Cisco IOS Flexible NetFlow Overview”
Flexible NetFlow Feature Roadmap	“Cisco IOS Flexible NetFlow Features Roadmap”
Configuring flow exporters to export Flexible NetFlow data.	“Configuring Data Export for Cisco IOS Flexible NetFlow with Flow Exporters”

Related Topic	Document Title
Customizing Flexible NetFlow	"Customizing Cisco IOS Flexible NetFlow Flow Records and Flow Monitors"
Configuring flow sampling to reduce the overhead of monitoring traffic with Flexible NetFlow	"Using Cisco IOS Flexible NetFlow Flow Sampling to Reduce the CPU Overhead of Analyzing Traffic"
Configuring Flexible NetFlow using predefined records	"Configuring Cisco IOS Flexible NetFlow with Predefined Records"
Using Flexible NetFlow Top N Talkers to analyze network traffic	"Using Cisco IOS Flexible NetFlow Top N Talkers to Analyze Network Traffic"
Configuring IPv4 multicast statistics support for Flexible NetFlow	"Configuring IPv4 Multicast Statistics Support for Cisco IOS Flexible NetFlow"

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
• CISCO-FLOW-MONITOR-TC-MIB • CISCO-FLOW-MONITOR-MIB • CISCO-RTP-METRICS-MIB • CISCO-IP-CBR-METRICS-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3954	*Cisco Systems NetFlow Services Export Version 9* http://www.ietf.org/rfc/rfc3954.txt
RFC 3550	*RTP: A Transport Protocol for Real-Time Applications* http://www.ietf.org/rfc/rfc3550.txt

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 Cisco Performance Monitor

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 5: Feature Information for Cisco Performance Monitor

Feature Name	Releases	Feature Information
Cisco Performance Monitor 1.0	15.1(3)T 12.2(58)SE 15.1(4)M1 15.0(1)SY Cisco IOS XE Release 3.5S 15.1(1)SG Cisco IOS XE Release 3.3 SG	This feature enables you to monitor the flow of packets in your network and bec of any issues that might impact the flow before it starts to significantly impact y applications' performance. Support for this feature was added for Cisco ASR 1000 Series Aggregation Servi in Cisco IOS XE Release 3.5S. There are some limitations to the monitoring of ingress or egress data on certain interfaces for the Cisco IOS XE Release 3.3 SG and Cisco IOS release 15.1(1)SC information, see the "Limitations" section. For all other releases, the following commands were introduced or modified by t **action**(policy react and policy inline react), **alarm severity** (policy react and po react), **alarm type**(policy react and policy inline react), **class-map**, **clock-rate**(p **collect application media**, **clear fm performance-monitor counters**, **collect c** **collect flow direction**, **collect interface**, **collect ipv4**, **collect ipv4 destinatio** **ipv4 source**, **collect ipv4 ttl**, **collect monitor event**, **collect routing**, **collect t** **interval**, **collect transport event packet-loss counter**, **collect transport packe** **transport rtp jitter**, **debug fm performance-monitor counters**, **debug performance-monitor counters**, **description** (Performance Monitor), **destinat** (Flexible NetFlow), **export-protocol**, **exporter**, **flow monitor type performanc** **flow record type performance-monitor**, **flows**, **history** (monitor parameters), **duration**, **match access-group**, **match any**, **match class-map**, **match cos**, **m** **destination-address mac**, **match discard-class**, **match dscp**, **match flow**, **m** **match fr-dlci**, **match input-interface**, **match ip dscp**, **match ip precedence**, **rtp**, **match ipv4**, **match ipv4 destination**, **match ipv4 source**, **match mpls exp** **topmost**, **match not**, **match packet length** (class-map), **match precedence**, **m** **protocol**, **match qos-group**, **match source-address mac**, **match transport destination-port**, **match transport rtp ssrc**, **match transport source-port**, **m** **max-dropout** (policy RTP), **max-reorder** (policy RTP), **min-sequential** (polic **monitor metric ip-cbr**, **monitor metric rtp**, **monitor parameters**, **option** (F NetFlow), **output-features**, **platform performance-monitor rate-limit**, **policy** **performance-monitor**, **rate layer3**, **react** (policy), **record** (Performance Monito (policy), **service-policy type performance-monitor**, **show performance monit** **show performance monitor status**, **show platform hardware acl entry interf** **platform software ccm**, **show platform software feature-manager performanc** **show platform software feature-manager tcam**, **show policy-map type performance-monitor**, **snmp-server host**, **snmp-server enable traps flowmon**, **flowmon alarm history** , **source**(Flexible NetFlow), **ssrc maximum**, **templat** **timeout**, **threshold value** (policy react and policy inline react), **timeout** (monitor p **transport** (Flexible NetFlow), and **ttl** (Flexible NetFlow).

Feature Name	Releases	Feature Information
Cisco Performance Monitor (phase 2)	15.2(2)T Cisco IOS XE Release 3.5S	This feature enables you monitor IPv6 fields and also use all other Flexible Netflow and match commands not supported in the previous release. Flows are now correlated so that if the same policy is applied on the same input and interface, the show command will display a single flow for the input and output inte Support for this feature was added for Cisco ASR 1000 Series Aggregation Services in Cisco IOS XE Release 3.5S. The following commands were introduced or modified by this feature: **collect datalin** **collect ipv4 fragmentation**, **collect ipv4 section**, **collect ipv4 total-length**, **colle** **collect ipv6 destination**, **collect ipv6 extensionmap**, **collect ipv6 fragmentation**, **ipv6 hop-count**, **collect ipv6 length**, **collect ipv6 section**, **collect ipv6 source**, **c** **routing is-multicast**, **collect routing multicast replication-factor**, **collect timest** **sys-uptime**, **collect transport**, **collect transport icmp ipv4**, **collect transport icm** **collect transport tcp**, **collect transport udp**, **match application name**, **match con** **transaction-id**, **match datalink dot1q vlan**, **match datalink mac**, **match datalin** **match interface**, **match ipv4 fragmentation**, **match ipv4 section**, **match ipv4 total-** **match ipv4 ttl**, **match ipv6**, **match ipv6 destination**, **match ipv6 extension map**, **ipv6 fragmentation**, **match ipv6 hop-limit**, **match ipv6 length**, **match ipv6 sect** **match ipv6 source**, **match routing**, **match routing is-multicast**, **match routing mu** **replication-factor**, **match transport**, **match transport icmp ipv4**, **match transpo** **ipv6**, **match transport tcp**, **match transport udp**
Cisco Performance Monitor (phase 3)	15.2(3)T Cisco IOS XE Release 3.7S	This feature enables you to configure multiple exporters and monitor metadata field new TCP metrics. Support for this feature was added for Cisco ASR 1000 Series Aggregation Services in Cisco IOS XE Release 3.7S. The following commands were introduced or modified by this feature: **collect appli** **collect transport tcp bytes out-of-order**, **collect transport packets out-of-order**, **transport tcp maximum-segment-size**, **collect transport tcp window-size maxim** **collect transport tcp window-size minimum**, **collect transport tcp window-size a** **match application**, **match transport tcp bytes out-of-order**, **match transport pa** **out-of-order**, **match transport tcp maximum-segment-size**, **match transport tcp** **window-size maximum**, **match transport tcp window-size minimum**, **match tra** **tcp window-size average**
Performance Monitoring - IPv6 support	Cisco IOS XE Release 3.6S	This feature enables you to attach a monitor to IPv6 interfaces. Support for this feature was added for Cisco ASR 1000 Series Aggregation Services in Cisco IOS XE Release 3.6S.
Performance Monitoring - transport packet out of order	Cisco IOS XE Release 3.6S	This feature enables you to monitor the total number of out-of-order TCP packets. Support for this feature was added for Cisco ASR 1000 Series Aggregation Services in Cisco IOS XE Release 3.6S. The following commands were introduced or modified by this feature: **collect transp** **bytes out-of-order** and **collect transport packets out-of-order**.

Feature Name	Releases	Feature Information
Flexible NetFlow: IPFIX Export Format	15.2(4)M Cisco IOS XE Release 3.7S	Enables sending export packets using the IPFIX export protocol. The export of e fields from NBAR is only supported over IPFIX. Support for this feature was added for Cisco ASR 1000 Series Aggregation Servi in Cisco IOS XE Release 3.7S. The following command was introduced: **export-protocol**.
Flexible NetFlow: Export to an IPv6 Address	Cisco IOS XE Release 3.7S	This feature enables Flexible NetFlow to export data to a destination using an IP Support for this feature was added for Cisco ASR 1000 Series Aggregation Servi in Cisco IOS XE Release 3.7S. The following command was introduced: **destination**.
Flexible NetFlow: Extracted Fields Support	Cisco IOS XE Release 3.7S	Enables the collection of extracted fields using NBAR. The export of extracted fi supported over IPFIX. Support for this feature was added for Cisco ASR 1000 Series Aggregation Servi in Cisco IOS XE Release 3.7S. The following commands were introduced or modified by this feature: **collect ht** **collect nntp group-name**, **collect pop3 server** , **collect rtsp host-name**, **collect** **destination**, **collect sip source**, **collect smtp server**, ,and **collect smtp sender**.

Feature Name	Releases	Feature Information
Application Visibility and Control (AVC) 2.0, which includes the following features: • Enable visualization of application usage under performance-monitoring policy • Enable performance of application usage • Enable Prime integration with router packet capture • Enable visualization of service path • FNF: Account On Resolution (AOR) for WAAS Segment • FNF: Account On Resolution (AOR) for performance monitoring policy-map	Cisco IOS XE Release 3.8S	AVC 2.0 provides extensive new functionality, including the integration of AVC wit Media Monitoring technology. This book only describes how to configure a flow record for AVC 2.0. For a comple explanation of AVC 2.0, see the *AVC Configuration Guide* at http://www.cisco.com/en/US/docs/ios-xml/ios/avc/configuration/xe-3s/avc-xe-3s-boo