Implementing Video Monitoring

Configuring Video Monitoring is a four-step procedure, which includes configuring the relevant class-maps and policy maps, and binding the video monitoring policy to an interface.

Prerequisites for Implementing Video Monitoring

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- You must install and activate packages for advanced video services. For detailed information about optional package installation, see *Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*.
- You must install and activate a package for the multicast routing software and enable multicast routing on the system. Video monitoring is supported on interfaces that are multicast-enabled. For detailed information about multicast routing, refer to the chapter Implementing Layer 3 Multicast Routing on Cisco ASR 9000 Series Routers.

Information About Implementing Video Monitoring

Video Monitoring

Poor video experience is a major cause for concern among service providers in terms of service costs and loss of revenue. To avoid the service costs of help desk time, NOC (network operation center) troubleshooting resources, and truck rolls, the capability of monitoring video traffic is essential. On Cisco IOS XR software, problems in video flows can be easily diagnosed by video monitoring.

Introduction to Video Monitoring

Packet loss is one common cause of video quality degradation. Its impact is more significant on compressed video flows. The video traffic transported through the service provider IP network is mostly compressed video – MPEG or similar encoding. Because of the way compression occurs, the traffic is extremely loss-sensitive. The video is encoded with an independent frame (I-frame) every few seconds, with subsequent frames being deltas from the I-frame. If the loss is in an I-frame, a 3 ms loss of traffic (roughly one IP packet) can result in a viewing degradation for up to 1.2 seconds.

Jitter is a key flow characteristic that requires careful buffer provisioning in the end device. The set top box (STB) that displays the media on a screen needs to decode the video in real-time. It buffers the incoming video stream so that it can decode and display the image smoothly. Large network jitter can lead to buffer underrun or overrun on the STB. Depending on how large the jitter is, this will create a visual artifact or even a "black screen" at the display.

End-to-end delay in transmission is not significant for a broadcast-only application. However, as the video applications get to be more interactive, the end-to-end latency (delay) becomes a critical Quality of Experience (QoE) component. Data loss is a major contributor for poor QoE.

Three main contributors to poor QoE can be summed up as:

- Packet Loss
- Jitter
- Delay

Video Monitoring plays a very significant role in improving video quality, and thus, in enhancing the QoE. Video monitoring is implemented on the routers and enables network operators to measure and track video transport performance on a per-flow basis. The video packets flow through a router. We can use the packet headers and compute a metric that gives us a measure of the network performance impacting the quality of the video. This information from multiple routers is compared for the same flow to get a clear end-to-end picture of the video issues in the network and the affected flows.

Problems in video flows (and more generally, any streaming flow) can be diagnosed by video monitoring. The purpose of video monitoring is to detect perturbations and anomalies introduced by the network that cause a degraded QoE; that is, it measures the transport performance for streaming (video) traffic. Encoding errors, audio-video-lag, and other errors too cause poor QoE. However, these are introduced by the encoding device and not the network. These latter errors are not monitored.

Key Features Supported on Video Monitoring

Direct Measurements from Data Plane

Video monitoring plays a significant role in improving video quality and therefore enhances the QoE. Video monitoring implemented on Cisco ASR 9000 Series Routers enable the network operator to measure and track video transport performance on a per-flow basis in real time. In contrast to the conventional traffic monitoring solutions, (where sampled flows have to be sent to the control plane or additional hardware, such as dedicated blade on the router), video monitoring on Cisco ASR 9000 Series Router performs the monitoring operation on the data plane itself. This enables video monitoring to analyze forwarded packets in real time, to compute a metric that provides a measure of the network performance impacting the quality of the video.

Local Storage and Remote Access

Video monitoring measures packet loss and jitter at wire-speed, and stores collected information on the router, in order that the network operator can access it through a user interface. Furthermore, the performance metrics measured and stored on multiple routers can be accessed through standard SNMP from a remote operation center. These metrics provide a clear end-to-end picture of the video flow that can be composed and analyzed.

Proactive and Reactive Usages

Video monitoring on Cisco ASR 9000 Series Routers serve both reactive and proactive usage for service providers. It can be used to verify the quality of video service, before scaling up the service coverage to new customers. Also, it is a powerful tool for analysis and can be used to troubleshoot customer calls. Network operators can configure video monitoring to raise an alarm for various events such as variation in packet loss, jitter, flow rate, number of flows, and so on. Such an alarm can be configured to get triggered at any possible value or range.

Flow on Video Monitoring

Video monitoring uses four pieces of packet header fields to distinguish a unique flow - source IP address, destination IP address, source UDP port, and destination UDP port (this implies protocol ID is always UDP).

Unicast and Multicast

Video monitoring supports not only the monitoring of flows with IPv4 multicast destination address in the IP header, but also supports the monitoring of flows with unicast destination addresses. The support for video monitoring functionality for unicast flows provides backward compatibility to ASR 9000 Ethernet Line Card, and is also available on ASR 9000 Enhanced Ethernet Line Card .

Flow Rate Types and Protocol Layer

Video monitoring monitors CBR (constant bit rate) flows at the IP layer. In other words, video monitoring can monitor CBR-encoded media streams (for example, MPEG-2) encapsulated in UDP datagram, inside an IPv4 packet. Video monitoring allows users to configure packet rate at IP layer, or bit rate at media layer (along with the number and size of media packets).

Metrics

Video monitoring supports both packet loss and jitter metrics that follow MDI (media delivery index, RFC 4445) definition at the IP-UDP level. The MDI metrics are MLR (media loss rate) and DF (delay factor). Video monitoring uses MRV (media rate variation) which is an extension of MDI MLR; that is, MLR captures only loss, while MRV captures both loss and excess. Video monitoring DF is the same as MDI definition, where DF represents one nominal packet inter-arrival time in addition to the monitored MDI jitter. Along with the two key metrics, Video monitoring supports packet count, byte count, packet rate, bit rate, packet size, TTL (Time to Live) field in IP header, number of flows, raised alarms, and time stamp for various events.

Note The term MDI jitter, is used to signify the correctness of DF metric measured by Video monitoring. MDI jitter is measured by comparing the actual packet arrival time against the nominal arrival reference, while simple inter-packet jitter is measured by the time difference between two consecutive packet arrivals. The former captures the performance of CBR flow more precisely than the latter.

Number of Flows

In the current release, video monitoring on Cisco ASR 9000 Series Router supports 1024 flows per NP(network processor) on ASR 9000 Ethernet Line Card and a maximum of 4096 flows per NP on ASR 9000 Enhanced Ethernet Line Card, for combined unicast and multicast traffic. The number of maximum flows for each line card or for each system varies, depending on the number of NPs on the line card and the number of line cards on the system. Per-chassis flow scale depends on the number of NPs on the chassis.

For example, if you have a Cisco ASR 9000 Series Router box with 4 ASR 9000 Ethernet Line Cards, and if each LC has 8 NPs, per-chassis flow scales up to 1K*8 = 8K flows for each chassis.

High Availability Features

Video monitoring on Cisco ASR 9000 Series Router supports high availability at various levels. It supports process OIR (online insertion and removable), line card OIR, RSP (route switch processor) fail over, and router reload. Configuration is persistent for all high availability scenarios. Monitored statistics data are preserved at process OIR and RSP FO.

Interface Types and Direction

To activate video monitoring, you must configure video monitoring service policy on an interface. There are four types of interfaces to which you can attach the video monitoring policy; these are main interface, subinterface, ethernet bundle interface, and ethernet bundle subinterface. Video monitoring supports only layer 3 interfaces and not layer 2 interfaces. Video monitoring can be configured only on the input direction of the interface.

Flow Rate and DF Precision

Video monitoring on Cisco ASR 9000 Series Router offers DF metric performance of 1 ms precision.

Video monitoring supports standard definition (SD) video traffic (mostly compressed) of up to 100 Mbps flow rate. For uncompressed video streams, flow rate of max 3 Gbps is supported.

User Interface for Input

Video monitoring supports traditional CLI (command line interface) input for configuration that follows MQC (modular QoS configuration) syntax. You can configure video monitoring by configuring access control list (ACL), class map, and policy map; it can be activated by attaching the service policy to an interface. In-place policy modification is not supported. Once attached to an interface, the configured service policy can be modified only after detaching it from the interface.

User Interface for Output

Video monitoring offers various show and clear commands for retrieving the monitored statistics. Refer the Video Monitoring Commands on Cisco ASR 9000 Series Routers module in the *Multicast Command Reference for Cisco ASR 9000 Series Routers* for a detailed description of the video monitoring commands.

You can configure TCA (threshold crossing alert) as a part of the policy map to enable video monitoring to generate syslog message for various conditions. You can also retrieve standing alarms by using **show** command or through a SNMP pull. XML is supported by video monitoring.

Number of Class Maps and Policy Maps

To use video monitoring, you must configure class map and policy map that acts as a filter to determine which flow to monitor on the data plane. Video monitoring supports a maximum of 1024 class maps per policy-map, and a maximum of 1024 class maps per system. It supports a maximum of 256 policy maps on the system.

Video PIE Installation

Video monitoring requires video PIE installation. Depending on the RSP type, the video pie name has two versions:

- asr9k-video-p.pie (RSP2 version)
- asr9k-video-px.pie (RSP3 version)

Video Monitoring Trap and Clone

Trap and clone is an extension to the basic performance monitoring service feature, where the packets from a selected number of flows can be filtered (trapped), duplicated (cloned), and sent to a remote device on the network for a more fine-grained analysis of the video quality. The cloned packets are replicated by the multicast forwarding process to the interface specified in the performance traffic clone profile. The remote device can perform a deeper analysis of the data at the MPEG layer level. This device can be used both as a debugging and a monitoring tool. Also, this device can act as a service engine blade on the same router. For multicast flows, the trap and clone functionality is fully backward-compatible; however, for unicast flows, it is supported with Layer 3 Switched Port Analyzer (SPAN) on Typhoon LCs.

Note L3 SPAN does not support SNMP. For more information on L3 SPAN, refer to Configuring SPAN.

Video Monitoring Terminology

To implement and configure video monitoring service on Cisco ASR 9000 Series Routers, you must first understand video monitoring terminology and concepts.

Interval duration and interval updates

Video monitoring analyzes continuously all packets on the data plane for a time period called interval duration, which is configured by the user. Statistics are exported periodically at the end of each interval duration. These exported statistics are called interval updates. The status of a video monitoring flow and its transition is described solely in reference to these interval updates. Also, all exported video monitoring flow statistics are stored in terms of these interval updates.

The interval duration is a vital video monitoring parameter. Video monitoring configuration anchors upon interval duration for functions such as frequency of export, number of exports to store, time to delete inactive flows, and so on. All video monitoring functionalities, including raising alarm (for stopped flows and flows with performance degradation), are based on the contents of interval updates.

Video monitoring flows

A video monitoring flow is an instance of a packet stream whose header fields match the configured class map (and its associated access control list). A unique flow is local to the interface to which a video monitoring service policy is attached. A video monitoring flow is composed of a series of stored interval updates. A

unique flow that is created on video monitoring after a monitoring interval is called a new flow. Therefore, a packet stream that lives for a period shorter than one monitoring interval is not exported as a video monitoring flow, and is therefore not stored.

Flow stop

If the router stops receiving packets on a monitored flow for one full interval update or longer, the monitored flow is considered as being stopped.

Flow resumption

When a stopped video monitoring flow resumes receiving packets, a normal interval update is exported in the next monitoring interval. A resumed flow has one or more zero intervals, followed by a normal interval update.

Flow switchover

A video monitoring flow on an ethernet bundle interface, or on an ethernet bundle sub-interface, may move from one physical member interface to another; that is, the packet stream stops flowing on one interface and starts flowing on another interface. This is defined as a flow switchover. In such a case, if both interfaces are on the same line card, video monitoring treats the pre-switchover flow and the post-switchover flow as the same flow. Otherwise, it treats them as two different flows.

Flow deletion

If a stopped video monitoring flow continues to export zero intervals for a configured timeout (in terms of the number of monitoring intervals), the flow is considered dead and is marked for deletion. The duration for which the user can control inactive flows is indicated using the timeout parameter. The actual deletion for all the marked flows takes place after some delay by the periodic sweeping function, which is executed every 150 seconds for Trident LC, and executed every 60 seconds for Typhoon LC. Once deleted, all exported statistics (series of interval updates including zero intervals) are completely removed from storage.

Implementing Video Monitoring

Configuring Video Monitoring is a four-step procedure, which includes configuring the relevant class-maps and policy maps, and binding the video monitoring policy to an interface.

Creating IPv4 Access Lists

This step is similar to typical IPv4 access list creation and configuration. An example configuration of ACL for video monitoring is presented here for quick reference. For more details, refer to the *Implementing Access lists and Prefix lists chapter* of the *IP Addresses and Services Configuration Guide for Cisco ASR 9000 Series Routers*.

This task configures a standard IPv4 access list.

Standard access lists use source addresses for matching operations.

Note Video Monitoring policy allows **deny** statements in ACL configuration, but **deny** statements are treated as **permit**. Also, log or log-input is not supported in ACL configuration.

SUMMARY STEPS

1. **configure**
2. **ipv4 access-list** *name*
3. [*sequence-number*] **remark** *remark*
4. [*sequence-number*] **permit udp** *source* [*source-port*] **destination** [*destination-port*]
5. Repeat Step 4 as necessary, adding statements by sequence number. Use the **no** *sequence-number* command to delete an entry.
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**ipv4 access-list** *name* **Example:** `RP/0/RSP0/CPU0:router# ipv4 access-list acl_1`	Enters IPv4 access list configuration mode and configures access list acl_1.
Step 3	[*sequence-number*] **remark** *remark* **Example:** `RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 remark Do not allow user1 to telnet out`	(Optional) Allows you to comment on the **permit** statement that follows in the named access list. • The remark can be up to 255 characters; anything longer is truncated. • Remarks can be configured before or after **permit** statements, but their location details should be consistent.
Step 4	[*sequence-number*] **permit udp** *source* [*source-port*] **destination** [*destination-port*] **Example:** `RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 permit udp 172.16.0.0/24 eq 5000 host 225.0.0.1 eq 5000`	Allows you to specify the source and destination ports with these conditions. • Video monitoring supports only udp. • Use the *source* keyword to specify the network or host number from which the packet is being sent. • Use the optional *source-wildcard* argument to specify the wildcard bits to be applied to the source. • Use the *destination* keyword to specify the network or host number to which the packet is being sent.

	Command or Action	Purpose
		• Use the optional *destination-wildcard* argument to specify the wildcard bits to be applied to the destination.
Step 5	Repeat Step 4 as necessary, adding statements by sequence number. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise an access list.
Step 6	**commit**	

Configuring class-map

This task sets up the flow classifier. This may match either an individual flow, or it may be an aggregate filter matching several flows.

SUMMARY STEPS

1. **configure**
2. **class-map type traffic** *class-map-name*
3. **match access-group ipv4** *acl-name*
4. **end-class-map**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**class-map type traffic** *class-map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# class-map type traffic class1`	Enters the class-map mode. The class-map type must always be entered as traffic.
Step 3	**match access-group ipv4** *acl-name* **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# match access-group ipv4 acl1`	Enter the ACL to be matched for this class. Only one ACL can be matched per class.
Step 4	**end-class-map** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	Completes the class-map configuration.
Step 5	**commit**	

Configuring policy-map

The policy map for video monitoring is of the performance-traffic type. Only one level of hierarchy is supported for video monitoring policy-maps. This means that no hierarchical policy map configuration is supported for video monitoring.

The policy map configuration for video monitoring has these three parts:

- Flow parameters configuration: Specifies the different properties of the flow that are monitored such as interval duration, required history intervals, timeout, etc.
- Metric parameters configuration: Specifies the metrics that need to be calculated for the flow that are monitored.
- React parameters configuration: Specifies the parameters, based on which, alerts are generated for the flow.

The configuration hierarchy is from *policy* to *class* to *flow*. This means that all the parameters that are specified above are applied to all flows that match a particular class, in the policy-map. While specifying flow and react parameters for flows matching a given class is optional, its metric parameters is mandatory.

Configuring policy-map with metric parameters

The metric parameters in a policy map can be:

- Layer 3 packet rate or
- Media bit rate (with the number of media packet counts and size in the UDP payload specified).

Note Layer 3 packet rate and Media rate have mutually exclusive configuration commands.

The configuration for each metric parameter is described in this section.

Layer 3 packet-rate

SUMMARY STEPS

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *traffic class-name*
4. **monitor metric ip-cbr**
5. **rate layer3 packet** *packet-rate* **pps**
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type** *performance-traffic policy-map-name* **Example:**	Enters the policy-map mode. The policy-map type should always be entered as performance traffic.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	
Step 3	**class type** *traffic class-name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic class-name`	Enter the class-map to be matched for this policy. Multiple classes can be specified for a single policy.
Step 4	**monitor metric ip-cbr** **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c)# monitor metric ip-cbr`	Enters the IP-CBR metric monitor submode. **Note** Currently only ip-cbr metric monitoring is supported for video monitoring.
Step 5	**rate layer3 packet** *packet-rate* **pps** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-ipcbr)# rate layer3 packet packet-rate pps`	Specifies the IP layer3 packet rate in packets per second (pps).
Step 6	**commit**	

Media bit-rate

The metric parameters for media bit-rate consist of the media bit rate, media packet count and packet size. The rate media option enables the user to specify the number of media payload packets (that is MPEG-2 datagrams) that is present in one UDP packet, and the size of each of such media payload. It is mandatory to specify the media bit rate.There are no defaults for packet count and packet size in Cisco IOS XR Software Release 3.9.1. These values must be configured.

Note With the media bit rate configured to 1052800 bps, media packet count to 7, and media packet size to 188 bytes, the media packet rate is 100 pps at layer 3. The calculation is: 1052800 / (7 *188*x 8) = 100 pps.

SUMMARY STEPS

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *traffic class-name*
4. **monitor metric ip-cbr**
5. **rate media** *bit -rate* **{bps|kbps|mbps|gbps}**
6. **media packet count in-layer3** *packet-count*
7. **media packet size packet-size**

8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type** *performance-traffic policy-map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	Enters the policy-map mode. The policy-map type should always be entered as performance traffic.
Step 3	**class type** *traffic class-name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic class-name`	Enters the class-map to be matched for this policy. Multiple classes can be specified for a single policy.
Step 4	**monitor metric ip-cbr** **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c)# monitor metric ip-cbr`	Enters the IP-CBR metric monitor submode. **Note** Currently only ip-cbr metric monitoring is supported for video monitoring.
Step 5	**rate media** *bit -rate* **{bps|kbps|mbps|gbps}** **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-ipcbr)# rate media 100 mbps`	Specifies the media bit rate for the flow in bps, kbps, mbps or gbps. The configuration can be committed here. Optional parameters can also be specified. **Note** The default unit of media bit-rate is kbps.
Step 6	**media packet count in-layer3** *packet-count* **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-ipbr)# media packet count in-layer3 10`	Specifies the number of media packets for each IP payload.
Step 7	**media packet size packet-size** **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-ipcbr)# media packet size 188`	Specifies the size in bytes for each media packet in the IP payload.
Step 8	**commit**	

Configuring policy-map with flow parameters

The flow parameters in a policy map are optional.

For video monitoring, the data plane continuously monitors the flows and the metrics that are exported at the end of every interval. The duration of this interval and the number of such intervals that need to be stored for each flow (history) can also be optionally specified by the user. You can specify these flow parameters for each flow:

- Interval Duration: The time interval at whose end, metrics are exported. This is specified in multiples of 5 (any value between 10 and 300 seconds). The default value is 30.

- History: The number of intervals containing flow information (flow ID, metrics, etc.) that needs be stored for each flow. This can be any value between 1 and 60. The default value is 10.
- Timeout: The timeout in multiples of interval duration after which an inactive flow is marked for deletion. This can be any value between 2 and 60. The default value is 0. (Note: the timeout value of 0 has a special meaning: the flow will never be timed out and is therefore a static flow).
- Max Flows per class: The maximum number of flows that need to be monitored for each class in the policy. This can be any value between 1 and 1024. The default value is 1024.

SUMMARY STEPS

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *traffic class-name*
4. monitor parameters
5. {**interval duration** *duration* | **flows** *number of flows* | **history** *intervals* | **timeout** *duration*}
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type** *performance-traffic policy-map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	Enters the policy-map mode. The policy-map type should always be entered as performance traffic.
Step 3	**class type** *traffic class-name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic class-name`	Enter the class-map to be matched for this policy. Multiple classes can be specified for a single policy.
Step 4	monitor parameters **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c)# monitor parameters`	Enters the flow monitor submode.
Step 5	{**interval duration** *duration* | **flows** *number of flows* | **history** *intervals* | **timeout** *duration*} **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-fparm)#`	• Select the **interval duration** option to specify the interval duration per flow; range is 10 to 300 (must be in multiples of 5). The default value is 30. • Select the **history** option to specify the maximum number of interval data that will be stored per flow. It

	Command or Action	Purpose
	`interval duration 10`	can be any value between 1 and 60. The deafult value is 10. • Select the **timeout** option to specify the timeout in multiples of the interval duration after which an inactive flow will be marked for deletion. Range is between 2 and 60. The default value is 0, indicating a static flow. • Select the **flows** option to specify the maximum number of flows that can be monitored per class. Range is between 1 and 1024. The default valueis 1024.
Step 6	**commit**	

Configuring policy-map with react parameters

The react parameters in a policy map are optional.

The react parameters are a direct reference for the user to indicate the flow quality. The flow is continuously monitored, and at the end of the interval duration, the statistics are examined to determine whether the threshold specified by the user for the specific parameter has exceeded. If it has, a syslog alarm is generated on the console. Once the alarm is set, no further syslog notifications are issued for the condition.

The following react parameters are used to configure the policy-map:

- Media Rate variation (MRV): video monitoring reacts and generates an alarm if the MRV statistic of the flow crosses the user-specified threshold.
- Delay Factor: video monitoring reacts and generates an alarm if the Delay Factor statistic of the flow crosses the user-specified threshold.
- Media-Stop: video monitoring reacts and generates an alarm if a flow stops; this is to indicate that no packets were received for the flow during one full monitoring interval.
- Packet-Rate: video monitoring reacts and generates an alarm if the packet rate of the flow crosses the user-specified threshold.
- Flow-Count: video monitoring reacts and generates an alarm if the flow count for each class crosses the user-specified threshold.

SUMMARY STEPS

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *traffic class-name*
4. **react react-id** {**mrv** | **delay-factor** | **packet-rate** | **flow-count** | **media-stop**}
5. **threshold type immediate**
6. **threshold value** {**ge** | **gt** | **le** | **lt** | **range**} *limit*
7. **action syslog**
8. **alarm severity** {**error** | **critical** | **alert** | **emergency**}
9. **alarm type** {**discrete** | **grouped**}

10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type** *performance-traffic policy-map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	Enters the policy-map mode. The policy-map type should always be entered as performance traffic.
Step 3	**class type** *traffic class-name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic class-name`	Enter the class-map to be matched for this policy. Multiple classes can be specified for a single policy.
Step 4	**react react-id** {**mrv** | **delay-factor** | **packet-rate** | **flow-count** | **media-stop**} **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c)# react 1 mrv`	Enters the react parameter configuration submode. The react ID specified here needs to be unique for each class. **Note** For the media-stop react parameter, the threshold-type and threshold-value options are not applicable. For the flow-count react parameter, the alarm-type option is not applicable.
Step 5	**threshold type immediate** **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# threshold type immediate`	Specifies the trigger type for the threshold. Currently, the available threshold type is immediate.
Step 6	**threshold value** {**ge** | **gt** | **le** | **lt** | **range**} *limit* **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# threshold value ge 50`	Specifies the trigger value range for the threshold.
Step 7	**action syslog** **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# action syslog`	The **action** keyword specifies the action to be taken if the threshold limit is surpassed. Currently, syslog action is the only option available.

	Command or Action	Purpose
Step 8	**alarm severity** {**error** | **critical** | **alert** | **emergency**} **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# alarm severity critical`	Specifies the alarm severity for syslog.
Step 9	**alarm type** {**discrete** | **grouped**} **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# alarm type discrete`	Specifies the alarm type. Discrete alarm is raised for all the flows that exceed the threshold value. Grouped alarm is raised when a certain number or percentage of the flows exceeds the threshold value.
Step 10	**commit**	

Video Monitoring Metrics

Video monitoring supports RTP, MDI and MPLS metrics in this release.

- The variations of RTP supported are RTP-MMR, RTP-Voice, RTP-J2k, and RTP-Custom
- The variations of MDI supported are MDI-MPEG, and MDI-MPEG over RTP
- The variations of MPLS supported are RSVP-TE, P2MP-TE, LDP, and MLDP

Configuring policy-map with rtp metric parameters

The configuration for each rtp metric parameter is described in this section.

SUMMARY STEPS

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *traffic class-name*
4. **monitor** *parameters*
5. **timeout** *duration*
6. **exit**
7. **monitor metric**[**rtp** | **rtp-j2k** | **rtp-mmr** | **rtp-voice**]
8. **clock-rate** *value*
9. **max-dropout** *value*
10. **max-misorder** *value*
11. **min-sequential** *value*
12. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type** *performance-traffic policy-map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	Enters the policy-map mode. The policy-map type should always be entered as performance traffic.
Step 3	**class type** *traffic class-name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic class-name`	Enter the class-map to be matched for this policy. Multiple classes can be specified for a single policy.
Step 4	**monitor** *parameters* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# monitor parameters`	Enters the flow monitor submode.
Step 5	**timeout** *duration* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-mparm)# timeout 2`	The timeout in multiples of interval duration after which an inactive flow is marked for deletion. This can be any value between 2 and 60.
Step 6	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-mparm)# exit`	Exits from the flow monitor submode.
Step 7	**monitor metric**[**rtp** | **rtp-j2k** | **rtp-mmr** | **rtp-voice**] **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c)# monitor metric rtp`	• Enters the corresponding rtp metric monitor submode. The available options are: • rtp - This option is used for custom rtp traffic. • rtp-j2k -This option is used to monitor RTP JPEG 2000 traffic. • rtp-mmr - This option is used to monitor Microsoft Mediaroom traffic. • rtp-voice - This option is used to monitor RTP voice traffic.

	Command or Action	Purpose
		Note When rtp-j2k, rtp-mmr and rtp-voice metrics are used for monitoring, frequency mapping in the dynamic range is configured automatically for specific frequencies. The rtp metric parameter is used for custom rtp traffic. You need to configure the frequency mapping dynamically for the rtp metric parameter. • Enters the flow monitor submode.
Step 8	**clock-rate** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-rtp)# clock-rate 97`	This option is available with the rtp monitor metric only. Enter the dynamic payload type value. Range is from 96 to 27. The RTP clock rate used for generating the RTP timestamp is independent of the number of channels and encoding. The RTP clock rate equals the number of sampling periods per second. The clock frequency for most video streams is 90 kHz. RTP supports all static payload type codes and allows a user to configure dynamic payload type frequency mapping. The available payload type values are: • 8kHz frequency • 16kHz frequency • 11.025kHz frequency • 22.050kHz frequency • 44.1kHz frequency • 48kHz frequency • 90kHz frequency (default frequency for RTP) • 27000kHz frequency • 148500kHz frequency • 148.5/1.001MHz frequency
Step 9	**max-dropout** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-rtp)# max-dropout 20`	This option is available with the rtp monitor metric only. Enter the maximum dropout value for RTP flow. The range enforced at policy map creation time is from 1 to 65536. The range enforced at bind time is from 0 to 255. In order to identify an out-of-order packet, a sliding window is maintained to accept non-sequential packets as long as they are with-in the window. Max-dropout provides the look-ahead configuration for sliding window. A packet with sequence number x is considered valid if x is no more than max-dropout ahead of current sequence number.

	Command or Action	Purpose
		For RTP, 128 clock frequency-payload type mapping tables are supported.
Step 10	**max-misorder** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-rtp)# max-misorder 20`	This option is available with the rtp monitor metric only. Enter the maximum misorder value. The range enforced at policy map creation time is from 1 to 65536. The range enforced at bind time is from 0 to 255. A packet with sequence number x is considered valid if x is no more than max-misorder behind the current sequence number. A sequence number is considered valid only if it is neither more than max-dropout ahead of max seq (currently seen maximum sequence number) nor more than max-misorder behind.
Step 11	**min-sequential** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-rtp)# min-sequential 20`	This option is available with the rtp monitor metric only. Enter the minimum sequential value. The range enforced at policy map creation time is from 1 to 65536. The range enforced at bind time is from 0 to 255. Since UDP header does not have any protocol specific information, there is no way to uniquely identify an RTP packet. Instead, a heuristic way of examining RTP headers of N packet is used in PD to identify the flow. The number of packets is defined by metric parameter of min-sequential.
Step 12	**commit**	

Configuring policy-map with rtp react parameters

The configuration for each rtp metric with react parameter is described in this section.

SUMMARY STEPS

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *traffic class-name*
4. **monitor** *parameters*
5. **timeout** *duration*
6. **exit**
7. **monitor metric**[**rtp** | **rtp-j2k** | **rtp-mmr** | **rtp-voice**]
8. **react react-id** {**rtp-loss-fraction** | **rtp-jitter** | **rtp-out-of-order** | **rtp-loss-pkts** | **rtp-transport-availability** | **rtp-error-seconds** | **flow-count** | **packet-rate**}
9. **action** [**snmp** | **syslog** | **clone**]
10. **alarm type** [**discrete** | **grouped** { **count** *number* | **percent** *percentage*}]
11. **alarm severity** [**alert** | **critical** | **emergency** | **error**]
12. **threshold** {**ge** | **gt** | **le** | **lt** | **range**} *limit*
13. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type** *performance-traffic policy-map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	Enters the policy-map mode. The policy-map type should always be entered as performance traffic.
Step 3	**class type** *traffic class-name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic class-name`	Enter the class-map to be matched for this policy. Multiple classes can be specified for a single policy.
Step 4	**monitor** *parameters* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# monitor parameters`	Enters the flow monitor submode.
Step 5	**timeout** *duration* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-mparm)# timeout 2`	The timeout in multiples of interval duration after which an inactive flow is marked for deletion. This can be any value between 2 and 60.
Step 6	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-mparm)# exit`	Exits from the flow monitor submode.
Step 7	**monitor metric**[**rtp** | **rtp-j2k** | **rtp-mmr** | **rtp-voice**] **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c)# monitor metric rtp`	• Enters the corresponding rtp metric monitor submode. The available options are: • rtp - This option is used for custom rtp traffic. • rtp-j2k -This option is used to monitor RTP JPEG 2000 traffic. • rtp-mmr - This option is used to monitor Microsoft Mediaroom traffic. • rtp-voice - This option is used to monitor RTP voice traffic.

	Command or Action	Purpose
		Note When rtp-j2k, rtp-mmr and rtp-voice metrics are used for monitoring, frequency mapping in the dynamic range is configured automatically for specific frequencies. The rtp metric parameter is used for custom rtp traffic. You need to configure the frequency mapping dynamically for the rtp metric parameter. • Enters the flow monitor submode.
Step 8	**react react-id** {**rtp-loss-fraction** | **rtp-jitter** | **rtp-out-of-order** | **rtp-loss-pkts** | **rtp-transport-availability** | **rtp-error-seconds** | **flow-count** | **packet-rate**} **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c)# react 1 rtp-loss-fraction`	Enters the react parameter configuration submode. The react ID specified here needs to be unique for each class. The available options are: • rtp-error-seconds - This option is used for RTP error seconds. Error seconds signifies the amount of time the stream was errored. • rtp-jitter - This option is used for RTP jitter. RTP jitter signifies the average interpacket jitter based on RTP timestamp. • rtp-loss-fraction - This option is used for RTP loss fraction. Loss fraction signifies the percentage of packets that are lost. • rtp-loss-pkts - This option is used for RTP loss packets. Loss pakets signifies the number of packets that are lost. • rtp-max-jitter -This option is used for RTP max jitter. Maximum instantaneous jitter during an time interval. • rtp-out-of-order - This option is used for RTP out-of-order packets. Out-of-order packets signifies the number of misordered packets. • rtp-transport-availability - This option is used for RTP transport availability. Transport availability signifies the percentage of time during which the stream does not have any errors. For example, if the RTP error seconds is zero, the RTP transport availablity is hundred percent. • flow-count - This option is used for Flow Count. Flow count signifies the number of flows on a policy. • packet-rate - This option is used for Packet Rate. Packet rate signifies the number of packets during a given time interval.

	Command or Action	Purpose
Step 9	**action** [**snmp** | **syslog** | **clone**] **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# action snmp`	The action keyword specifies the action to be taken if the threshold limit is surpassed.
Step 10	**alarm type** [**discrete** | **grouped** { **count** *number* | **percent** *percentage*}] **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# alarm type discrete`	Specifies the alarm type. Discrete alarm is raised for all the flows that exceed the threshold value. Count alarms are grouped based on number of flows. Percent alarms are grouped based on percentage of flows.
Step 11	**alarm severity** [**alert** | **critical** | **emergency** | **error**] **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# alarm severity critical`	Specifies the alarm severity for syslog.
Step 12	**threshold** {**ge** | **gt** | **le** | **lt** | **range**} *limit* **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# threshold value ge 50`	Specifies the trigger value range for the threshold.
Step 13	**commit**	

Configuring policy-map with mdi metric parameters

The configuration for each mdi metric parameter is described in this section.

SUMMARY STEPS

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *class-map-name*
4. **monitor** *parameters*
5. **timeout** *duration*
6. **exit**
7. **monitor metric**[**mdi mpeg** \| **mdi mpeg rtp**]
8. **max-dropout** *value*
9. **monitor pids** *id*
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type** *performance-traffic policy-map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	Enters the policy-map mode. The policy-map type should always be entered as performance traffic.
Step 3	**class type** *class-map-name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic class-name`	Enter the class-map to be matched for this policy. Multiple classes can be specified for a single policy.
Step 4	**monitor** *parameters* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# monitor parameters`	Enters the flow monitor submode.
Step 5	**timeout** *duration* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-mparm)# timeout 2`	The timeout in multiples of interval duration after which an inactive flow is marked for deletion. This can be any value between 2 and 60.
Step 6	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-mparm)# exit`	Exits from the flow monitor submode.
Step 7	**monitor metric**[**mdi mpeg** | **mdi mpeg rtp**] **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c)# monitor metric mdi mpeg`	Enters the corresponding mdi metric monitor submode. The mdi mpeg rtp option signifies the presence of an rtp header beofre the mpeg header. A maximum of 7 mpeg packets per IP packet are allowed. If a packet contains more than 7 mpeg packets, then the ip packet is ignored. If encapusulation does not match, the flows will not be learned.
Step 8	**max-dropout** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-mdi)#`	Enables packet filtering based on lower bound of stream rate. Range is 1 to 4294967294.

	Command or Action	Purpose
	`max-dropout 20`	
Step 9	**monitor pids** *id* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-mdi)# monitor pids 200`	Enable static PID monitoring. The range enforced at policy map creation time is from 1 to 65536. The range enforced at bind time is from 16 to 8191.
Step 10	**commit**	

Configuring policy-map with mdi react parameters

The configuration for each mdi metric with react parameter is described in this section.

SUMMARY STEPS

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *traffic class-name*
4. **monitor** *parameters*
5. **timeout** *duration*
6. **exit**
7. **react react-id** {**mdi-mlr** | **mdi-mdc** | **mdi-transport-availability** | **mpeg-loss-pkts** | **mdi-error-seconds** | **rtp-error-seconds** | **flow-count** | **mdi-jitter** | **packet-rate** | **media-stop**}
8. **action** [**snmp** | **syslog** | **clone**]
9. **alarm type** [**discrete** | **grouped** { **count** *number* | **percent** *percentage*}]
10. **alarm severity** [**alert** | **critical** | **emergency** | **error**]
11. **threshold** {**ge** | **gt** | **le** | **lt** | **range**} *limit*
12. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type** *performance-traffic policy-map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	Enters the policy-map mode. The policy-map type should always be entered as performance traffic.
Step 3	**class type** *traffic class-name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type`	Enter the class-map to be matched for this policy. Multiple classes can be specified for a single policy.

	Command or Action	Purpose
	`traffic class-name`	
Step 4	**monitor** *parameters* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# monitor parameters`	Enters the flow monitor submode.
Step 5	**timeout** *duration* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-mparm)# timeout 2`	The timeout in multiples of interval duration after which an inactive flow is marked for deletion. This can be any value between 2 and 60.
Step 6	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c-mparm)# exit`	Exits from the flow monitor submode.
Step 7	**react react-id** {**mdi-mlr** | **mdi-mdc** | **mdi-transport-availability** | **mpeg-loss-pkts** | **mdi-error-seconds** | **rtp-error-seconds** | **flow-count** | **mdi-jitter** | **packet-rate** | **media-stop**} **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c)# react 1 rtp-loss-fraction`	Enters the react parameter configuration submode. The react ID specified here needs to be unique for each class. The available options are: • mdi-error-seconds - MDI error seconds • mdi-mdc - MDI Media Disc. Count • mdi-mlr - MDI Media Loss Rate • mdi-transport-availability - MDI transport availability • mpeg-loss-pkts - MPEG loss packets • flow-count - Flow Count • mdi-jitter - MDI Jitter • packet-rate - Packet Rate • media-stop - Media Stop Event
Step 8	**action** [**snmp** | **syslog** | **clone**] **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# action snmp`	The action keyword specifies the action to be taken if the threshold limit is surpassed.
Step 9	**alarm type** [**discrete** | **grouped** { **count** *number* | **percent** *percentage*}]	Specifies the alarm type. Discrete alarm is raised for all the flows that exceed the threshold value.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router(config- pmap-c-react)# alarm type discrete`	Count alarms are grouped based on the number of flows and percent alarms are grouped based on the percentage of flows.
Step 10	**alarm severity** [**alert** | **critical** | **emergency** | **error**] **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# alarm severity critical`	Specifies the alarm severity for syslog.
Step 11	**threshold** {**ge** | **gt** | **le** | **lt** | **range**} *limit* **Example:** `RP/0/RSP0/CPU0:router(config- pmap-c-react)# threshold value ge 50`	Specifies the trigger value range for the threshold.
Step 12	**commit**	

Configuring flow monitor

Perform this step to configure flow monitor.

SUMMARY STEPS

1. **configure**
2. **flow monitor-map performance-traffic** *monitor-name*
3. **exporter** *exporter-map-name*
4. **record** { **default-rtp | default-mdi** }
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**flow monitor-map performance-traffic** *monitor-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# flow monitor-map performance-traffic m1`
	`RP/0/RSP0/CPU0:router(config-fmm)#`
		Configures the flow monitor map.

Step 3	**exporter** *exporter-map-name* **Example:**	Enter the flow exporter map name.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-fmm)# exporter e1` `RP/0/RSP0/CPU0:router(config-fmm)#`	

Step 4	**record** { **default-rtp** | **default-mdi** }
	Example:
	`RP/0/RSP0/CPU0:router(config-fmm)# record default-rtp`
	`RP/0/RSP0/CPU0:router(config-fmm)#`
		Enter the flow record map name. The available options are:
		• default-rtp - Default MDI record format
		• default-mdi - Default RTP record format

Step 5	**commit**	

Configuring service policy on an interface

The configured policy-map must be attached to an interface in ingress direction in order to enable the Video Monitoring service.

For ethernet bundle interface, service policy can be attached to only the bundle parent interface and not to the physical member interfaces. For ethernet bundle sub-interfaces, it can be attached to only sub-interfaces. For VLAN sub-interfaces, the service policy cannot be attached to the main interface.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **service-policy type performance-traffic input** *policy-name*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface type interface-path-id`	Configures an interface and enters interface configuration mode. • The type argument specifies an interface type. For more information on interface types, use the question mark (?) online help function. • The instance argument specifies either a physical interface instance or a virtual instance. • The naming notation for a physical interface instance is rack/slot/module/port. The slash (/) between values is required as part of the notation.

	Command or Action	Purpose
		• The number range for a virtual interface instance varies, depending on the interface type.
Step 3	**service-policy type performance-traffic input** *policy-name* **Example:** `RP/0/RSP0/CPU0:router(config-if)# service-policy type performance-traffic input policy1`	Attaches the policy to the interface in the ingress direction.
Step 4	**commit**	

Configuring Trap and Clone on an interface

As trap and clone is an extension of the existing video monitoring service, the current control plane infrastructure can be extended to accommodate the configurations for trap and clone.

You can use the flow tuple information (source and destination IP addresses) to install the trap, which eventually leads the matched packets to be further analyzed by a remote device or a local probe.

These steps show how the trap and clone process works in a generic video monitoring scenario:

- You must enable video monitoring by installing the appropriate packages (multicast and video PIEs) and configure ACL, class map, policy map, and bind the policy map to an interface.
- You must configure trap and clone by specifying which flows to clone by specifying the source and the destination of the flows.
- The trap gets installed in the data plane by the VidMon control plane and VidMon data plane starts cloning the packets for the specified flows.
- The cloned packets are forwarded to the remote monitoring device for further analysis.

Note You can use the **show performance traffic clone profile** command to verify the installed traps. The video monitoring trap and clone feature is supported only for multicast traffic, and for unicast flows the user is required to configure SPAN. In multicast, the video monitoring trap and clone feature is implemented using static IGMP groups on the clone interface. The clone interface can be on a dedicated port connected to a local probe.

SUMMARY STEPS

1. **configure**
2. **performance traffic clone profile** *profile_name*
3. **interface** *type interface-path-id*
4. **flow ipv4 source** <*source-ip*> **destination** <*destination-ip*>
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**performance traffic clone profile** *profile_name* **Example:** `RP/0/RSP0/CPU0:router(config)# performance traffic clone profile profile1`	Enters the performance traffic clone profile mode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-perf-traf-clone-profile)# interface GigabitEthernet 0/0/0/1`	Configures the egress interface to a clone profile.
Step 4	**flow ipv4 source** <*source-ip*> **destination** <*destination-ip*> **Example:** `RP/0/RSP0/CPU0:router(config-perf-traf-clone-profile)# flow ipv4 source 23.1.1.1 destination 224.2.2.2`	Configures the traffic flows that needs to be cloned, to the clone profile. **Note** Multiple flows can be associated with a single clone profile. Similarly, a single flow can be associated with the multiple clone profiles.
Step 5	**commit**	

Configuration Examples for Implementing Video Monitoring

Scenario-1

An ethernet bundle interface has three physical members over which multicast video traffic is flowing at 300 pps for each flow.

Use video monitoring to monitor all the flows on this ethernet bundle, and raise a critical-level alarm, if the per-flow traffic load is over 10 % of expected rate. Raise an error-level alarm if the delay factor is greater than 4 ms. Report the collected statistics every 10 seconds. As long as the flow is active, keep the reported statistics for 10 minutes. Remove flow statistics if no packets are received for 30 seconds.

Example

```
ipv4 access-list sample-acl
 10 permit udp any any
!
class-map type traffic match-any sample-class
 match access-group ipv4 sample-acl
 end-class-map
!
policy-map type performance-traffic sample-policy
```

```
 class type traffic sample-class
  monitor parameters
   interval duration 10
   history 60
   timeout 3
  !
  monitor metric ip-cbr
   rate layer3 packet 300 pps
  !
  react 100 mrv
   threshold type immediate
   threshold value gt 10.00
   action syslog
   alarm severity error
   alarm type discrete
  !
  react 101  delay-factor
   threshold type immediate
   threshold value gt 4.00
   action syslog
   alarm severity error
   alarm type discrete
  !
 !
 end-policy-map
!
interface Bundle-Ether10
 ipv4 address 172.192.1.1 255.255.255.0
 service-policy type performance-traffic input sample-policy
!
interface TenGigE0/6/0/0
 bundle id 10 mode on
!
interface TenGigE0/6/0/1
 bundle id 10 mode on
!
interface TenGigE0/6/0/2
 bundle id 10 mode on
!
```

Scenario-2

A VLAN subinterface is carrying 100 video streams with a common multicast group address of 225.0.0.1 and varying UDP port numbers. The expected packet rate at IP layer is unknown, but the media bit rate is known to be 1052800 bps. The media payload is known to contain MPEG-2 encoded CBR flows and default packetization is used (that is, in one UDP payload, there are seven MPEG packets, where each packet is 188 bytes long).

Do not monitor over 100 flows. Do not timeout and delete any flow even if flow stops, but raise an error-level alarm if the percentage of the stopped flows is over 90 %.

Example

```
ipv4 access-list sample-acl
 10 permit udp any host 225.0.0.1
!
class-map type traffic match-any sample-class
 match access-group ipv4 sample-acl
 end-class-map
!
```

```
policy-map type performance-traffic sample-policy
 class type traffic sample-class
  monitor parameters
   flows 100
!
  monitor metric ip-cbr
   rate media 1052800 bps
  !
  react 100 media-stop
   action syslog
   alarm severity error
   alarm type grouped percent 90
  !
 end-policy-map
!
interface GigabitEthernet0/0/0/0
 no shutdown
!
interface GigabitEthernet0/0/0/0.1
 encapsulation dot1q 500
 ipv4 address 172.192.1.1 255.255.255.0
 service-policy type performance-traffic input sample-policy
!
```

Under **monitor metric ip-cbr**, these two lines need not be configured as they are defaults:

- media packet count in-layer3 7
- media packet size 188

However, if these parameters are different from default values, they need to be configured.

Scenario-3

A main interface has three groups of multicast streams where the first group has UDP destination port of 1000, the second group has 2000, and the third group has 3000 and 4000. These three groups of streams flow at 100 pps, 200 pps, and 300 pps respectively.

Limit the maximum number of flows in each group to 300 flows and raise the error-level alarm, when they reach 90 % of the provisioned flow capacity.

Example

```
ipv4 access-list sample-acl-1
 10 permit udp any any eq 1000
!
ipv4 access-list sample-acl-2
 10 permit udp any any eq 2000
!
ipv4 access-list sample-acl-3
 10 permit udp any any eq 3000
 20 permit udp any any eq 4000
!
class-map type traffic match-any sample-class-1
 match access-group ipv4 sample-acl-1
 end-class-map
!
class-map type traffic match-any sample-class-2
 match access-group ipv4 sample-acl-2
 end-class-map
```

```
!
class-map type traffic match-any sample-class-3
 match access-group ipv4 sample-acl-3
 end-class-map
!
policy-map type performance-traffic sample-policy
 class type traffic sample-class-1
  monitor parameters
   interval duration 10
   history 60
   timeout 3
   flows 300
  !
  monitor metric ip-cbr
   rate layer3 packet 100 pps
  !
  react 100 flow-count
   threshold type immediate
   threshold value gt 270
   action syslog
   alarm severity error
  !
class type traffic sample-class-2
  monitor parameters
   interval duration 10
   history 60
   timeout 3
   flows 300
  !
  monitor metric ip-cbr
   rate layer3 packet 200 pps
  !
  react 100 flow-count
   threshold type immediate
   threshold value gt 270
   action syslog
   alarm severity error
  !
class type traffic sample-class-1
  monitor parameters
   interval duration 10
   history 60
   timeout 3
   flows 300
  !
  monitor metric ip-cbr
   rate layer3 packet 300 pps
  !
  react 100 flow-count
   threshold type immediate
   threshold value gt 270
   action syslog
   alarm severity error
  !
 !
 end-policy-map
!
interface GigabitEthernet0/0/0/0
 ipv4 address 172.192.1.1 255.255.255.0
 service-policy type performance-traffic input sample-policy
!
```

Scenario-4

A 10GE main interface receives six high definition (HD) video streams from the digital contents manager (DCM), directly connected to six HD cameras in a sports stadium. Each HD video stream is uncompressed and its bandwidth is as high as 1.611 Gbps at layer 2, which is equivalent to 140625 pps. These six streams are received with multicast groups of 225.0.0.1 through 225.0.0.6, and the UDP port number is 5000.

Raise a critical-level alarm when the delay factor of any flow is above 2 ms, or media loss ratio is above 5 %. Use 10s interval and keep maximum history. Do not monitor more than 6 flows on this interface. Do not time out inactive flows.

Example

```
ipv4 access-list sample-acl
 10 permit udp any eq 5000 225.0.0.0/24 eq 5000
!
class-map type traffic match-any sample-class
 match access-group ipv4 sample-acl
 end-class-map
!
policy-map type performance-traffic sample-policy
 class type traffic sample-class
  monitor parameters
   interval duration 10
   history 60
   flows 6
  !
  monitor metric ip-cbr
   rate layer3 packet 140625 pps
  !
  react 100 mrv
   threshold type immediate
   threshold value gt 5.00
   action syslog
   alarm severity critical
   alarm type discrete
  !
  react 200 delay-factor
   threshold type immediate
   threshold value gt 2.00
   action syslog
   alarm severity critical
   alarm type discrete
  !
 end-policy-map
!
interface TenGigE0/2/0/0
 ipv4 address 172.192.1.1 255.255.255.0
 service-policy type performance-traffic input sample-policy
!
```

Scenario-5

An ethernet interface is configured on a Cisco ASR 9000 Series Routers over which multicast video traffic is flowing. Use video monitoring to monitor the performance of all video flows on this ethernet interface. Use the video monitoring trap and clone feature to trap these flow packets and clone (or duplicate) them to a specified egress interface.

Configure a trap and clone profile containing flows that are to be cloned to the specified egress interface. Add a description to the profile.

Example

```
Performance traffic clone profile profile1
   Description video flows monitored by vidmon
   Interface GigE 0/1/1/1
   flow ipv4 source 23.1.1.1 destination 231.2.2.2
```

Scenario-6

A 100GE main interface is receiving 5 high definition (HD) video streams of unicast traffic. Each HD video stream is uncompressed and its bit rate is 3 Gbps. It is known that each stream is CBR flow and has packet rate of 284954 pps. The source of these streams is known as 192.1.1.2 and destinations are from 10.1.1.1 through 10.1.1.5. UDP port 7700 is used for both source and destination.

Raise a critical-level alarm when the delay factor of any of the flow is above 5 ms or CBR flow rate drops over 10% of expected nominal rate. Use 30 s interval and keep 10 intervals as history. Since this port is known to receive additional low rate VoD flows in near future, allow maximum flow count as 4000. Monitor the streams destined to 10.1.1.0/24 subnet only. When quality degradation is detected, report the alarm to NMS system in addition to the syslog output.

Example

```
ipv4 access-list sample-acl
 10 permit udp 192.1.1.2/32 eq 7700 10.1.1.0/24 eq 7700
!
class-map type traffic match-any sample-class  match access-group ipv4 sample-acl
end-class-map
!
policy-map type performance-traffic sample-policy  class type traffic sample-class
  monitor parameters
   interval duration 30
   history 10
   flows 4000
  !
  monitor metric ip-cbr
   rate layer3 packet 284954 pps
  !
  react 100 mrv
   threshold type immediate
   threshold value lt 10.00
   action syslog
   action snmp
   alarm severity critical
   alarm type discrete
  !
  react 200 delay-factor
   threshold type immediate
   threshold value gt 5.00
   action syslog
   action snmp
   alarm severity critical
   alarm type discrete
  !
 end-policy-map
!
```

```
interface HundredGigE0/1/0/1
 ipv4 address 172.192.1.1 255.255.255.0
 service-policy type performance-traffic input sample-policy
!
```

Scenario-7

Use video monitoring to monitor all the vidmon-rtp traffic.

Example

```
ipv4 access-list uc
 10 permit udp any 20.0.0.0/24
!
class-map type traffic match-any ucast
 match access-group ipv4 uc
 end-class-map
!
interface TenGigE0/2/0/10
 ipv4 address 10.0.0.1 255.255.255.0
 service-policy type performance input vidmon-rtp
 load-interval 30
!
policy-map type performance-traffic vidmon-rtp
 class type traffic ucast
  monitor parameters
   interval duration 10
   history 60
   timeout 2
  !
  monitor metric rtp
   clock-rate 96  48kHz
   clock-rate 97  27000kHz
   clock-rate 99  148500kHz
   clock-rate 100  148351.648kHz
  !
 !
 react 101  flow-count
   threshold type immediate
   threshold value gt 0
   action syslog
   alarm severity alert
  !
  react 102  media-stop
   action syslog
   alarm severity critical
   alarm type discrete
  !
 !
 end-policy-map
!
```

Scenario-8

Use video monitoring to monitor all the vidmon-rtp-j2k traffic.

Example

```
policy-map type performance-traffic vidmon-rtp-j2k
 class type traffic ucast
  monitor parameters
   interval duration 10
   history 60
   timeout 2
  !
  monitor metric rtp-j2k
 !
 end-policy-map
!
```

Scenario-9

Use video monitoring to monitor all the mdi mpeg traffic.

Example

```
policy-map type performance-traffic ipcbr-mdi
 class type traffic ucast
  monitor parameters
   interval duration 10
   history 60
   timeout 2
  !
  monitor metric mdi mpeg
   filter packet-rate 22 pps
  !
 !
 end-policy-map
!
```

Scenario-10

Use video monitoring to monitor all the mdi mpeg rtp traffic.

Example

```
policy-map type performance-traffic rtp-mdi
 class type traffic ucast
  monitor parameters
   interval duration 10
   history 60
   timeout 2
  !
  monitor metric mdi mpeg rtp
  !
 !
 end-policy-map
!
```

Additional References

Related Documents

Related Topic	Document Title
Multicast command reference document	*Multicast Command Reference for Cisco ASR 9000 Series Routers*
Getting started material	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
Modular quality of service command reference document	*Modular Quality of Service Command Reference for Cisco ASR 9000 Series Routers*

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: https://mibs.cloudapps.cisco.com/ITDIT/MIBS/servlet/index

RFCs

RFCs	Title
RFC4445	Proposed Media Delivery Index (MDI)

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport