CHAPTER 6

Medianet Management and Visibility Design Considerations

This chapter provides a high-level overview of various functionalities that can be used to provide management and visibility into video flows within an enterprise medianet. This functionality can be divided into the following two broad categories:

- Network-embedded—Management functionality embedded within the IP network infrastructure itself (that is, routers, switches, and so on). Network-embedded management functionality may benefit a single video application solution, or may benefit all video application solutions, depending on the specific functionality.
- Application-specific—Management functionality embedded within the components that comprise individual video application solutions, such as Cisco TelePresence, Cisco Digital Media Systems, Cisco IP Video Surveillance, and Cisco Desktop Video Collaboration. Although individual video application solutions co-exist over a converged IP network infrastructure, the application-specific management functionality may be unique to the video solution.

Note Management applications that make use of the functionality embedded within both the IP network infrastructure and/or individual components of video application solutions to provide a centralized point of monitoring, control, and reporting within the medianet infrastructure, may be considered a third category of functionality. Examples of such applications are the Cisco QoS Policy Manager (QPM), which provides centralized QoS provisioning and monitoring for Cisco router platforms. Future revisions of this design chapter may include discussion around these applications.

In this design guide, management functionality is presented using the International Organization for Standardization (ISO)/International Telecommunications Union (ITU) Fault, Configuration, Accounting, Performance, and Security (FCAPS) model. The five major categories of network management defined within the FCAPS model are as follows:

- Fault management—Detection and correction of problems within the network infrastructure or end device.
- Configuration management—Configuration of network infrastructure components or end devices, including initial provisioning and ongoing scheduled changes.
- Accounting management—Scheduling and allocation of resources among end users, as well as billing back for that use if necessary.
- Performance management—Performance of the network infrastructure or end devices, including maintaining service level agreements (SLAs), quality of service (QoS), network resource allocation, and long-term trend analysis.

- Security management—Maintaining secure authorization and access to network resources and end devices, as well as maintaining confidentiality of information crossing the network infrastructure.

Note Note that the security management aspects of the medianet infrastructure are only briefly discussed in this chapter. A separate chapter of this design guide deals with medianet security design considerations.

Network-Embedded Management Functionality

The following sections highlight functionality embedded within network infrastructure devices that can be used to provide visibility and management of video flows within an enterprise medianet. Although specific examples within each section discuss the use of a particular functionality for a specific video application solution (Cisco TelePresence, Cisco Digital Media Systems, Cisco IP Video Surveillance, or Cisco Desktop Video Collaboration), the features discussed can generally provide benefit across multiple video application solutions. A complete list of network-embedded management functionality is outside the scope of this document. Instead, for brevity, only specific features relevant to medianet management and visibility are discussed. Table 6-1 provides a high level summarization of the functionality discussed in following sections.

Table 6-1 Summary of Network-Embedded Management Functionality

Management Product /Tool	Management Functionality	Description
NetFlow	Performance and security management	• NetFlow services embedded within Cisco router and Cisco Catalyst switch platforms provide the ability to collect and export flow information that can be used to determine the amount of video traffic crossing key points within a medianet. Flow information collected at a NetFlow collector, such as the Cisco Network Analysis Module (NAM) can be used to provide ongoing monitoring and/or reports that may be used to determine whether adequate bandwidth is provisioned per service class to support the video traffic applications. • NetFlow export version 9 provides the ability to export multicast flows as well, providing some visibility into the amount of multicast traffic crossing key points within the medianet infrastructure. • Netflow can also be used to identify anomalous flows within the medianet infrastructure, alerting security operations staff of potential worm propagation or a DDoS attack. For further information, see the *Cisco SAFE Reference Guide* at the following URL: http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/SAFE_RG/SAFE_rg.html.
Cisco Network Analysis Module (NAM)	Performance management	• The Cisco Catalyst 6500 Series Network Analysis Module (NAM) provides the ability to monitor and generate reports regarding data flows within a medianet. Data flows from the supervisor of a Cisco Catalyst 6500 switch platform, SPAN/RSPAN ports, or NetFlow Data Export (NDE) from other routers and switches within the medianet infrastructure can be analyzed. • The NAM provides the ability to monitor and generate reports on traffic flows aggregated by Differentiated Services Code Point (DSCP) marking. This can assist in providing visibility into the amount of traffic per service class crossing key points within the medianet, and can aid in provisioning adequate bandwidth per service class across the network infrastructure.
IP service level agreements (IPSLAs)	Performance management	• IPSLA functionality embedded within Cisco Catalyst switches, Cisco IOS routers, and Cisco TelePresence endpoints can be used as a pre-assessment tool, to determine whether the medianet infrastructure has the capability to support additional video flows before becoming production resources. • IPSLAs may be used cautiously to perform ongoing performance monitoring of the medianet infrastructure to determine whether a particular video class is experiencing degradation because of packet loss and/or jitter.

Table 6-1 Summary of Network-Embedded Management Functionality (continued)

Management Product /Tool	Management Functionality	Description
Router and switch command-line interface	Performance management and fault management	• The traceroute utility can be used to determined the Layer 3 hop path of video flows through a medianet infrastructure. • After the path has been determined, high-level CLI commands such as **show interface summary** and **show interface** can be used on each router and switch along the path to determine quickly whether drops or errors are occurring on relevant interfaces. • Other platform-specific commands can be used to display packet drops per queue on Cisco Catalyst switch platforms. When separate traffic service classes (corresponding to different video applications) are mapped to different queues, network administrators can use these commands to determine whether particular video applications are experiencing degradation because of packet loss within the medianet infrastructure. • When policy maps are used to map specific traffic service classes (corresponding to different video applications) to software queues within Cisco router platforms, or hardware queues within certain Cisco Catalyst switch platforms, the **show policy-map** command can be used to display the amount of traffic per service class as well as drops experienced by the particular service class. Network administrators can use this command to determine whether adequate bandwidth is provisioned, as well as to determine whether particular video applications are experiencing degradation because of packet loss within the medianet infrastructure.
Syslog	Security management and fault management	• Telemetry using syslog can be used to provide some key fault management information on network infrastructure devices within a medianet, such as CPU utilization, memory utilization, and link status. • For further information regarding network security best practices, see the *Cisco SAFE Reference Guide* at the following URL: http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/SAFE_RG/SAFE_rg.html.

Table 6-1 Summary of Network-Embedded Management Functionality (continued)

Management Product /Tool	Management Functionality	Description
Simple Network Management Protocol (SNMP)	Security management, fault management, and performance management	• Telemetry using SNMP can also be used to provide key fault management information on network infrastructure devices within a medianet. • SNMP can be used to collect statistics from network infrastructure devices for performance management purposes. • SNMP traps can be generated for authentication failures to devices, providing an additional layer of security management.
AAA services	Security management	• AAA services can be used to provide centralized access control for security management, as well as an audit trail providing visibility into access of network infrastructure devices. • For further information regarding network security best practices, see the *Cisco SAFE Reference Guide* at the following URL: http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/SAFE_RG/SAFE_rg.html.

NetFlow

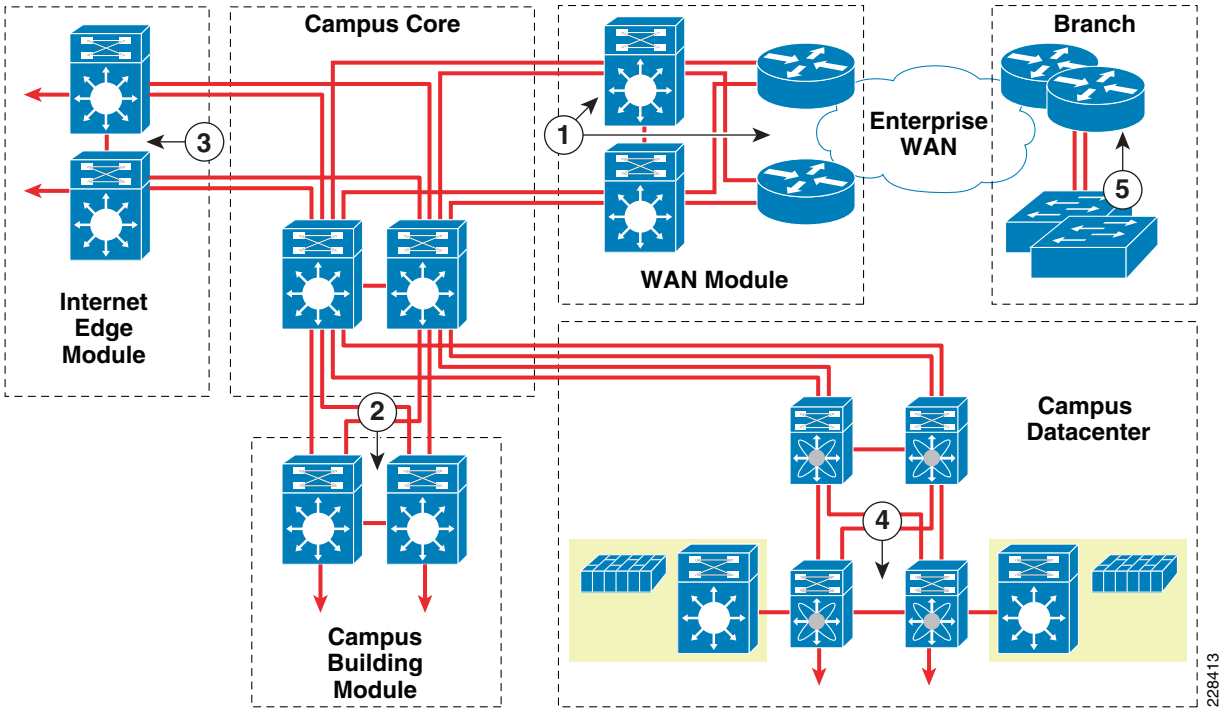
NetFlow services provide network administrators access to information regarding IP flows within their networks. IP flows are unidirectional streams of packets flowing through a network device. They share common properties such as source address, destination address, protocol, port, DSCP value, and so on. Network devices, such as switches and routers, can collect and store flow data in the form of flow records within a NetFlow table or cache. Flow records can then be periodically exported from the NetFlow cache to one or more NetFlow management collectors located centrally within a data center or campus service module. NetFlow collectors aggregate exported NetFlow records to provide monitoring and reporting information regarding the IP traffic flows within the network.

NetFlow provides a means of gaining additional visibility into the various video flows within an enterprise medianet. From an FCAPS perspective, this visibility can be used for either performance management purposes or for accounting management purposes. More specifically, NetFlow data can assist in determining whether sufficient bandwidth has been provisioned across the network infrastructure to support existing video applications. NetFlow data records can be exported in various formats depending on the version. The most common formats are versions 1, 5, 7, 8, and 9. NetFlow export version 9 is the latest version, which has been submitted to the IETF as informational RFC 3954, providing a model for the IP Flow Information Export (IPFIX) working group within the IETF. NetFlow version 9 provides a flexible and extensible means of exporting NetFlow data, based on the use of templates that are sent along with the flow record. Templates contain structural information about the flow record fields, allowing the NetFlow collector to interpret the flow records even if it does not understand the semantics of the fields. For more information regarding NetFlow version 9, see the following URL: http://www.ietf.org/rfc/rfc3954.txt.

NetFlow Strategies Within an Enterprise Medianet

Simply enabling NetFlow on every interface on every network device, exporting all the flow data to a central NetFlow collector, and then aggregating the flow data into a single set of information across the entire enterprise medianet, is generally considered only marginally useful for anything but small networks. This strategy typically results in information overload, in which a lot of statistics are collected, yet the network administrator has no idea where traffic is flowing within the network infrastructure. An alternative strategy is to collect flow information based on specific requirements for the flow data itself. One such strategy is to selectively enable NetFlow to collect traffic flows on certain interfaces at key points within the enterprise medianet. The data from each collection point in the network can then be kept as separate information sets, either at a single NetFlow collector or in multiple NetFlow collectors, rather than aggregated together. This can be used to provide a view of what traffic is flowing through the different points within the enterprise medianet. Depending on the capabilities of the NetFlow collector, this can be done in various ways. Some NetFlow collectors allow different UDP port numbers to be used for flows from different devices. This allows the aggregation of NetFlow information from multiple interfaces on a single router or switch to appear as a single data set or source. It also allows the flows from a redundant set of routers or switches to appear as a single data set or source. Other NetFlow collectors, such as the Cisco Network Analysis Module (NAM), use a fixed port (UDP 3000) for flows from devices. Flows from multiple interfaces on the same device can be aggregated into a single custom data source. Flows from multiple devices, such as a redundant pair of routers or switches, appear as separate data sources. However, the use of Virtual Switching System (VSS) on a pair of Cisco Catalyst 6500 Series switches allows flows from multiple interfaces on the redundant switch pair to appear as a single data source on the NAM. Figure 6-1 shows an example of some key network points within an enterprise medianet where NetFlow collection can be enabled. Note that pairs of Cisco Catalyst 6500 Series Switches can be VSS-enabled, although not specifically shown.

Figure 6-1 Example of Collecting NetFlow Data at Key Network Points



This example is not the only recommended model for enabling NetFlow within an enterprise medianet, but is an example of a methodology for collecting NetFlow data to gain some useful insight regarding video flows at various points within the network infrastructure. You can choose to selectively enable NetFlow collection at one or more strategic aggregation points in the network, such as the distribution layer within different modules of a campus, depending on the desired visibility for video flows. For example, NetFlow statistics can be collected at the ingress interfaces of the distribution layer switch pairs at each module within a campus. In other words, statistics can be collected for traffic flows exiting the core and entering each campus module. Statistics gathered from this type of NetFlow deployment can be used to determine the following video traffic flows:

- Aggregated flows outbound across the corporate WAN to all the branch locations
- Flows into each building within the campus
- Aggregated flows outbound toward the Internet

This model can be useful because many video flows emanate from a central point within a campus data center or campus service module, and flow out to users within each campus building or each branch location. For example, unicast or broadcast enterprise TV as well as video-on-demand (VoD) flows to desktop devices often follow this flow pattern. Likewise, because of the nature of TelePresence video, the majority of the video flows within a multipoint meeting are from a centralized Cisco TelePresence Multipoint Switch, potentially located within a data center or campus service module, out to the Cisco TelePresence System endpoints located within the campus buildings and branch locations. Additional flow information can be gathered by implementing NetFlow bidirectionally at the distribution layer of each module. Note that this can preferably be done by enabling NetFlow statistics collection in an ingress direction on other interfaces. Although video broadcasts, VoD, and multipoint TelePresence tend to follow a flow model where the majority of traffic emanates from a central point outward to the endpoints, Cisco IP video surveillance follows the opposite model. The majority of traffic in a Cisco IP video surveillance deployment flows from cameras deployed within the campus buildings back to the Video Surveillance Operations Manager (VSOM) server potentially deployed within a data center or campus service module. However, note that implementing NetFlow collection bidirectionally can result in some duplication of flow information when multiple collection points exist within the network infrastructure.

Additional flow information can also be gathered by implementing NetFlow at the branch router itself, to gain insight into the flows into and out of individual branch locations, if that level of detail is needed. Keep in mind, however, that the NetFlow data export uses some of the available branch bandwidth. Also, NetFlow in Cisco IOS router platforms is performed in software, potentially resulting in somewhat higher CPU utilization depending on the platform and the amount of flow statistics collected and exported. The use of flow filters and/or sampling may be necessary to decrease both CPU utilization and bandwidth usage because of NetFlow flow record exports. Even with the campus distribution switches, it may be desirable to implement flow filters and/or sampling to decrease CPU and bandwidth usage. Note that data sampling may distort statistics regarding how much traffic is flowing across a single point in the network. However, the relative percentages of the flows can still be useful from a bandwidth allocation perspective. An alternative strategy may be to SPAN the flow traffic from the Cisco Catalyst switch to a separate device, such as the Cisco Service Control Engine (SCE), which can then perform analysis of the flows and export records to a centralized NetFlow collector for monitoring and reporting.

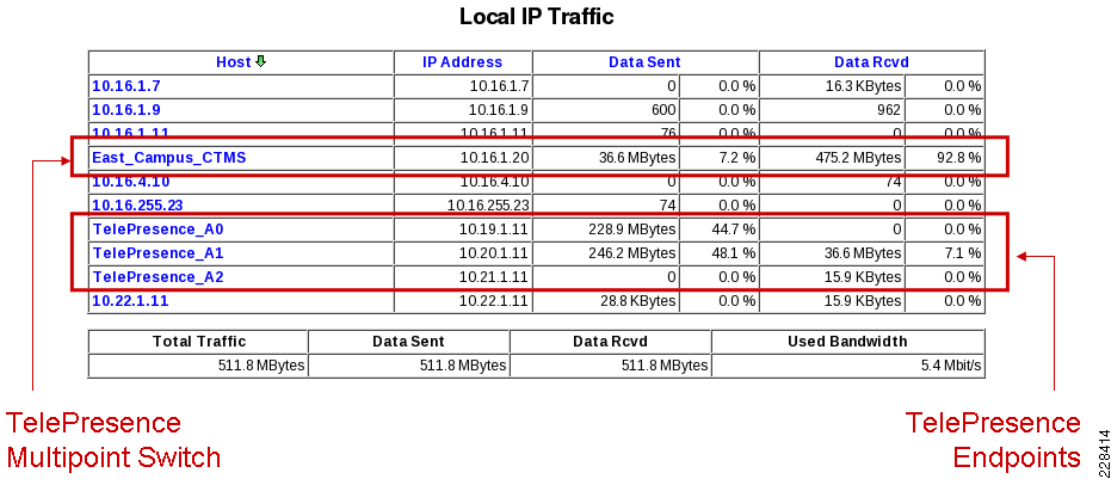
NetFlow Collector Considerations

The aggregation capabilities of the NetFlow collector determine to a large extent the usefulness of the NetFlow data from a medianet perspective. Most NetFlow collectors provide monitoring and historical reporting of aggregate bit rates, byte counts, and packet counts of overall IP data. Typically, this can be further divided into TCP, UDP, and other IP protocols, such as Internet Control Message Protocol (ICMP). However, beyond this level of analysis, some NetFlow collectors simply report Real-Time

Transport Protocol (RTP) traffic as “Other UDP “or “VoIP”, because RTP can use a range of UDP ports. Further, the ability to drill down to monitor and generate reports that show the specific hosts and flows that constitute RTP video and/or VoIP traffic, versus all UDP flows, may be limited. Further, both VoD, and in some cases video surveillance traffic, can be sent using HTTP instead of RTP. Therefore, generating useful reports showing medianet-relevant information, such as how much video data (RTP- and/or HTTP-based) is crossing a particular point within the network, may not be straightforward.

For devices such as TelePresence endpoints and IP video surveillance cameras, you can often simply assume that most of the data generated from the device is video traffic, and therefore use the overall amount of IP traffic from the device as a good estimate of the overall amount of video traffic generated by the device. Figure 6-2 shows a sample screen capture from a generic NetFlow collector, showing flow information from Cisco TelePresence System endpoints to a Cisco TelePresence Multipoint Switch, in a multipoint call.

Figure 6-2 Sample Host Level Reporting From a NetFlow Collector Showing TelePresence Endpoints



Note Figure 6-2 shows a screen capture from the open source ntop NetFlow collector.

The IP addresses of the TelePresence devices have been replaced by a hostname to more easily identify the endpoints. As can be seen, both the actual traffic sent and received, in terms of bytes, as well as the percentage of the overall traffic seen across this particular interface over time are recorded. Such information may be useful from the perspective of determining whether the percentage of bandwidth allocated for TelePresence calls relative to other traffic, across the interfaces of this particular collection point, matches the actual data flows captured over an extended period of time. However, this information must also be used with caution. Flow records are exported by NetFlow based on the following:

- The flow transport has completed; for example, when a FIN or RST is seen in a TCP connection.
- The flow cache has become full. The cache default size is typically 64 K flow cache entries on Cisco IOS platforms. This can typically be changed to between 1024 and 524,288 entries.
- A flow becomes inactive. By default on Cisco IOS platforms, a flow unaltered in the last 15 seconds is classified as inactive. This can typically be set between 10 and 600 seconds.
- An active flow has been monitored for a specified number of minutes. By default on Cisco IOS platforms, active flows are flushed from the cache when they have been monitored for 30 minutes. You can configure the interval for the active timer between 1 and 60 minutes.

- Routing device default timer settings are 15 seconds for the inactive timer and 30 minutes for the active timer. You can configure your own time interval for the inactive timer between 10 and 600 seconds. You can configure the interval for the inactive timer between 10 and 600 seconds.

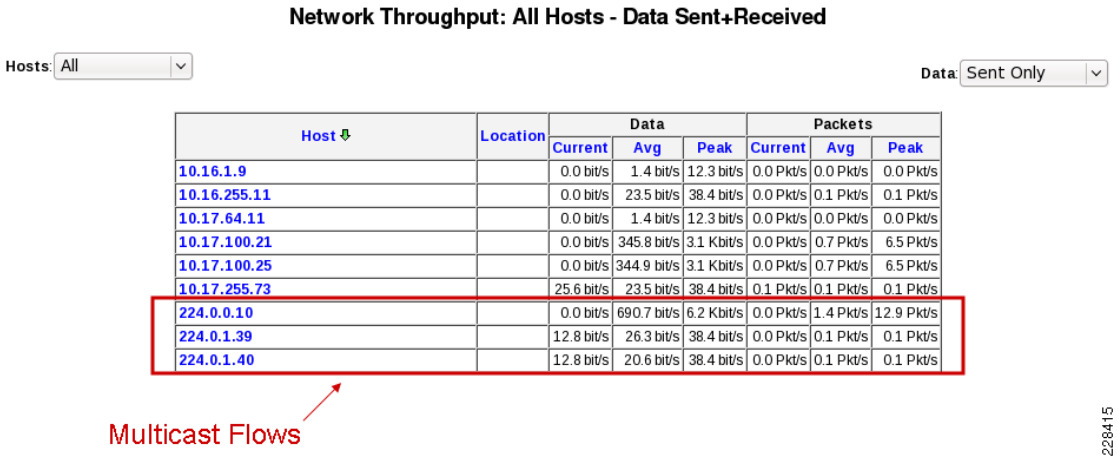
Long-lived flows such as TelePresence meetings may export flow data while the meeting is still ongoing. Therefore, the amount of data sent and/or received may not reflect the entire flow. In addition, the percentage of overall traffic does not indicate a particular timeframe, but more likely the percentage since collection began on the Netflow collector. The network administrator would benefit more from information that indicated the percentage of traffic during specific time intervals, such as peak times of the work day. Finally, the percentage of overall traffic represents an average over time, not peak usage, which may again be necessary to truly determine whether sufficient bandwidth is provisioned per service class across the medianet infrastructure.

The aggregation of flows based on type of service (ToS) may be useful from a medianet perspective, to characterize the amount or relative percentage of video traffic flows at given points within the network; provided the enterprise has deployed a QoS model that differentiates the various video flows into different service classes. This methodology also assumes a NetFlow collector capable of reporting flows based on ToS markings. NetFlow collectors such as the Cisco NAM Traffic Analyzer provide the ability to monitor and/or generate reports that show traffic flows based on DSCP values. NAM Analysis of NetFlow Traffic, page 6-15 discusses this functionality. If a NetFlow collector that provides aggregation and reporting based on medianet-relevant parameters is not available, it may be necessary in some situations to develop custom applications that show the appropriate level of flow details to provide relevant reporting information from an enterprise medianet perspective.

NetFlow Export of Multicast Traffic Flows

From a medianet perspective, NetFlow version 9 offers the advantage of being able to export flow data from multicast flows. Multicast is often used to efficiently broadcast live video events across the enterprise IP infrastructure, rather than duplicate multiple unicast streams to each endpoint. Figure 6-3 shows an example of multicast flows exported to a generic NetFlow collector.

Figure 6-3 Example of Multicast Flows Captured By a NetFlow Collector

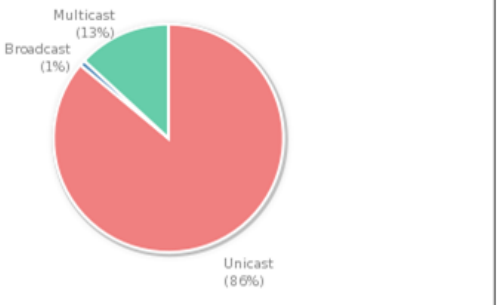


Note Figure 6-3 shows a screen capture from the open source ntop NetFlow collector.

Besides individual flows, which may be challenging to identify from all the other flow data, some NetFlow collectors can generate aggregate reporting information regarding the total amount of unicast, broadcast, and multicast flows seen at a given point within the network infrastructure. An example is shown in Figure 6-4.

Figure 6-4 Example of Aggregated Flow Data Reported By a NetFlow Collector

Dropped	0.0%	0
Total Received		74,010,349,321
Total Packets Processed		74,010,349,321
Unicast	86.1%	63,725,896,287
Broadcast	0.8%	560,214,024
Multicast	13.1%	9,724,239,010



228416

Note Figure 6-4 shows a screen capture from the open source ntop NetFlow collector.

The combination of individual multicast flow information as well as aggregated flow information may be useful in determining whether sufficient bandwidth has been provisioned across a particular point within the medianet infrastructure to support existing multicast flows.

NetFlow Configuration Example

The configuration snippets in Example 6-1 and Example 6-2 show a basic NetFlow configuration on a Cisco Catalyst 6500 Series Switch as well as on a Cisco IOS router platform. Note that this example shows no flow filtering or sampling, which may be necessary to decrease CPU and/or bandwidth utilization for NetFlow collection in production environments.

Example 6-1 NetFlow Configuration on a Cisco Catalyst 6500 Series Switch

```
mls netflow                                  ! Enables NetFlow on the PFC
mls flow ip interface-full                   ! Sets the NetFlow flow mask
mls nde sender                               ! Enables NetFlow device export
!
!
~
!
!
interface TenGigabitEthernet6/1
 description CONNECTION TO ME-EASTCORE-1 TEN5/4
 ip address 10.16.100.13 255.255.255.252
 ip flow ingress                             ! Enables MSFC NetFlow ingress on the interface
```

```
 ip multicast netflow ingress         ! Enables multicast NetFlow ingress on the interface
 ip pim sparse-mode
 no ip route-cache
 load-interval 30
 wrr-queue bandwidth 5 35 30
 priority-queue queue-limit 30
 wrr-queue queue-limit 5 35 30
 wrr-queue random-detect min-threshold 3 60 70 80 90 100 100 100 100
 wrr-queue random-detect max-threshold 1 100 100 100 100 100 100 100 100
 wrr-queue random-detect max-threshold 2 100 100 100 100 100 100 100 100
 wrr-queue random-detect max-threshold 3 70 80 90 100 100 100 100 100
 wrr-queue cos-map 1 1 1
 wrr-queue cos-map 2 1 0
 wrr-queue cos-map 3 1 2
 wrr-queue cos-map 3 2 3
 wrr-queue cos-map 3 3 6
 wrr-queue cos-map 3 4 7
 priority-queue cos-map 1 4 5
 mls qos trust dscp
!
interface TenGigabitEthernet6/2
 description CONNECTION TO ME-EASTCORE-2 TEN1/1
 ip address 10.16.100.1 255.255.255.252
 ip flow ingress                          ! Enables MSFC NetFlow ingress on the interface
 ip multicast netflow ingress         ! Enables multicast NetFlow ingress on the interface
 ip pim sparse-mode
 no ip route-cache
 load-interval 30
 udld port
 wrr-queue bandwidth 5 35 30
 priority-queue queue-limit 30
 wrr-queue queue-limit 5 35 30
 wrr-queue random-detect min-threshold 3 60 70 80 90 100 100 100 100
 wrr-queue random-detect max-threshold 1 100 100 100 100 100 100 100 100
 wrr-queue random-detect max-threshold 2 100 100 100 100 100 100 100 100
 wrr-queue random-detect max-threshold 3 70 80 90 100 100 100 100 100
 wrr-queue cos-map 1 1 1
 wrr-queue cos-map 2 1 0
 wrr-queue cos-map 3 1 2
 wrr-queue cos-map 3 2 3
 wrr-queue cos-map 3 3 6
 wrr-queue cos-map 3 4 7
 priority-queue cos-map 1 4 5
 mls qos trust dscp
!
!
~
!
!
ip flow-export source Loopback0   ! Sets the source interface of NetFlow export packets
ip flow-export version 9          ! Sets the NetFlow export version to version 9
ip flow-export destination 10.17.99.2 3000 ! Sets the address & port of the NetFlow
                                           collector
```

Example 6-2 NetFlow Configuration on a Cisco IOS Router

```
interface GigabitEthernet2/0
 description CONNECTS to BRANCH LAN SWITCH
 ip address 10.31.0.1 255.255.255.252
 ip flow ingress                  ! Enables NetFlow collection ingress on the interface
!
~
!
```

```
ip flow-export source Loopback0   ! Sets the source interface of NetFlow export packets
ip flow-export version 9          ! Sets the NetFlow export version to version 9
ip flow-export destination 10.16.4.10 2061   ! Sets the address and port of the NetFlow
                                             collector
```

For more information regarding the configuration of NetFlow on Cisco IOS routers, see the *Cisco IOS NetFlow Configuration Guide, Release 12.4* at the following URL: http://www.cisco.com/en/US/docs/ios/netflow/configuration/guide/12_4/nf_12_4_book.html.

For more information regarding the configuration of NetFlow on Cisco Catalyst 6500 Series Switch platforms, see the following documents:

- Configuring NetFlow— http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/netflow.html
- Configuring NDE— http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/nde.html

Cisco Network Analysis Module

The Cisco Network Analysis Module (NAM) enables network administrators to understand, manage, and improve how applications and services are delivered over network infrastructures. The NAM offers the following services:

- Flow-based traffic analysis of applications, hosts, and conversations
- Performance-based measurements on application, server, and network latency
- Quality of experience metrics for network-based services such as VoIP
- Problem analysis using packet captures

From an FCAPS management perspective, the NAM is most applicable as a performance management tool within an enterprise medianet, although both the packet capture and the monitoring statistics can also be used for fault management purposes. The current release of NAM software is version 4.1. The NAM software runs on the platforms listed in Table 6-2. Specific hardware configurations and OS versions required for support of NAM modules and/or software can be found in the documentation for each specific platform.

Table 6-2 NAM Platform Support

Cisco Product Platform	NAM Model
Cisco Catalyst 6500 Series Switches and Cisco 7600 Series Routers	WS-SVC-NAM-1-250S or WS-SVC-NAM-2-250S
Cisco 3700 Series Routers; Cisco 2811, 2821, and 2851 Series ISRs; Cisco 3800 Series ISRs; Cisco 2911, 2921, and 2951 Series ISR G2s; Cisco 3900 Series ISR G2s	NME-NAM-120S
Cisco WAVE-574 and Cisco WAE-674 with NAM 4.1 Software Running on a Virtual Blade	NAM-WAAS-VB
Standalone NAM Appliance	NAM 2204 or NAM 2220 Appliance

This document discusses only the use of the Cisco Catalyst 6500 Series Network Analysis Module (WS-SVC-NAM-2). Specific testing was performed with a WS-SVC-NAM-2 with a WS-SUP32P-10GE supervisor within a Cisco Catalyst 6506-E chassis. Other platforms may have slightly different functionality. The WS-SVC-NAM-2 can analyze and monitor network traffic in the following ways:

- The NAM can analyze chassis traffic via Remote Network Monitoring (RMON) support provided by the Cisco Catalyst 6500 Series supervisor engine.
- The NAM can analyze traffic from local and remote NetFlow Data Export (NDE).
- The NAM can analyze Ethernet LAN traffic via Switched Port Analyzer (SPAN), remote SPAN (RSPAN), or VLAN ACL (VACL); allowing the NAM to serve as an extension to the basic RMON support provided by the Cisco Catalyst 6500 Series supervisor engine.

This document discusses only certain functionality of the NAM as it relates to gaining visibility into video flows within an enterprise medianet. A comprehensive discussion of the configuration and monitoring functionality of the NAM is outside the scope of this document. For the end-user and configuration guides for the Cisco Network Analysis Module Software, see the following URL: http://www.cisco.com/en/US/products/sw/cscowork/ps5401/tsd_products_support_series_home.html.

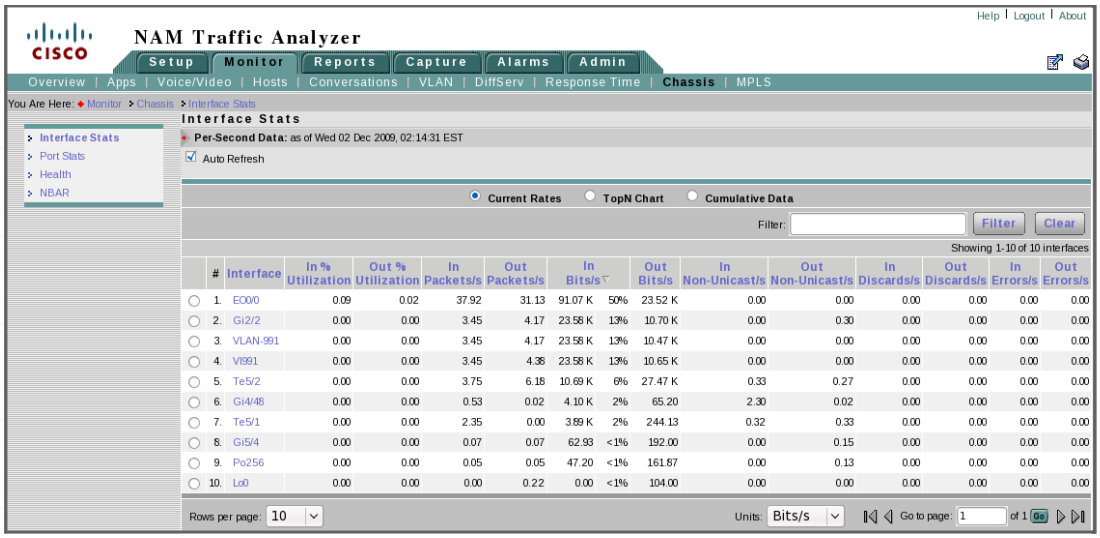
NAM Analysis of Chassis Traffic

The WS-SVC-NAM-2 has the ability to collect basic traffic statistics, per interface, from the supervisor line card within the Cisco Catalyst 6500 chassis. These statistics can be viewed as current rates or as cumulative data collected over time. Current rate data includes statistics such as the following:

- Input and output percentage utilization of the interface
- Input and output packets/second
- Input and output bit or byte rates
- Input and output non-unicast (multicast and broadcast) packets/second
- Input and output discards/second
- Input and output errors/second.

Figure 6-5 shows an example of the monitoring output.

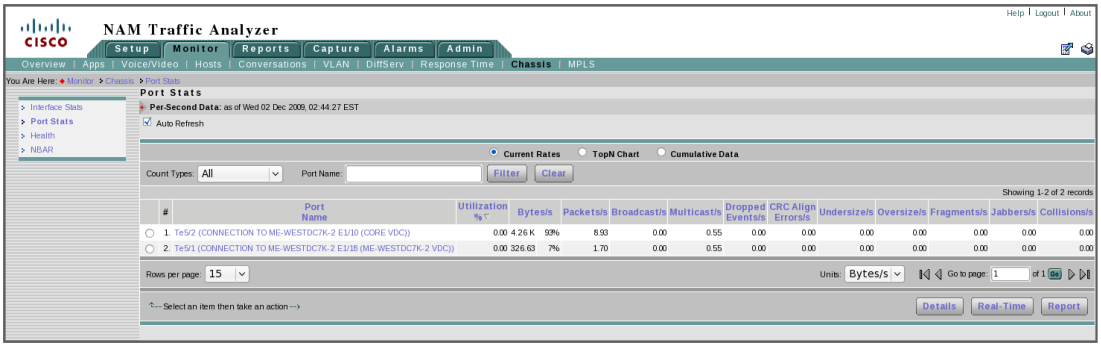
Figure 6-5 Example of WS-SVC-NAM-2 Traffic Analyzer Chassis Interface Statistics



From a medianet management perspective, these statistics can be used for high-level troubleshooting of traffic crossing the particular chassis, because even small rates of packet discards or interface errors can result in degraded video quality. Note, however, that the interface statistics alone cannot be used to determine whether video traffic is being discarded, because packet discards can be occurring only within a particular switch port queue that may or may not hold video traffic. However, as is discussed in Router and Switch Command-Line Interface, page 6-35, many router and switch platforms can show drops down to the level of individual queues within the CLI.

If mini-RMON port statistics are enabled, the WS-SVC-NAM-2 provides slightly more information regarding the types of errors encountered per interface, including undersized and oversized packets, Cyclic Redundancy Check (CRC) errors, fragments, jabbers, and collisions. Figure 6-6 provides an example showing port statistics on the uplink ports of a Cisco Catalyst 6500 switch.

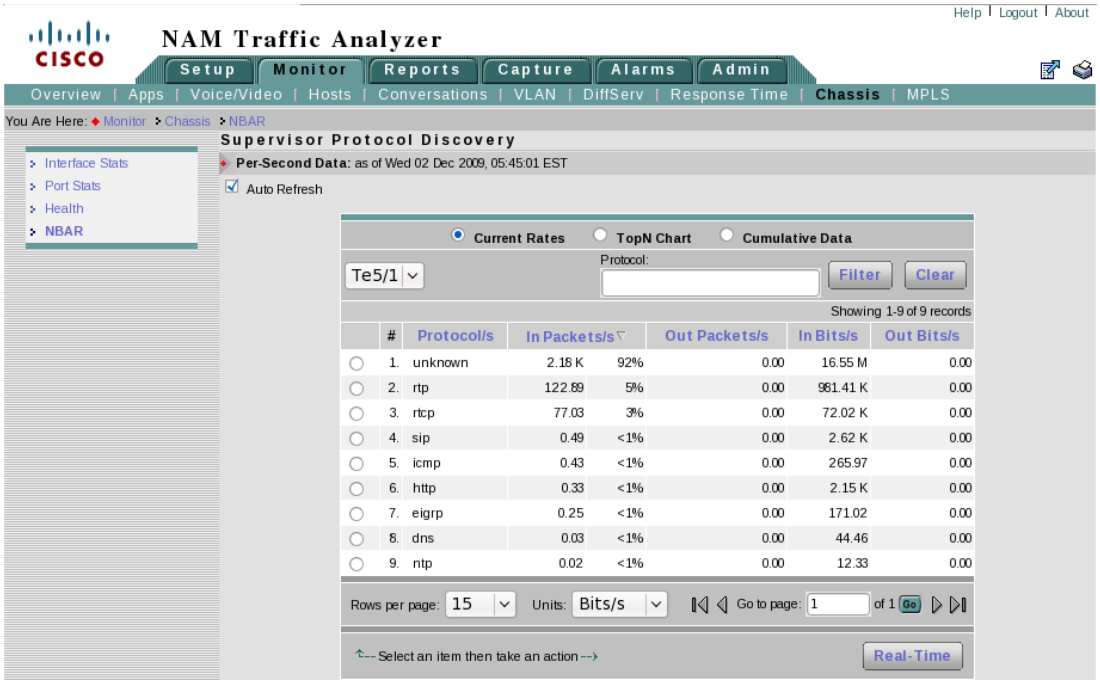
Figure 6-6 Example of W-SVC-NAM-2 Traffic Analyzer Chassis Port Statistics



The port statistics can also provide information regarding the amount multicast traffic crossing the interfaces. When viewed as current rates, the NAM port statistics show the number of multicast packets/second seen by the interface. These can be graphed in real time as well, or viewed as cumulative data. The port statistics do not show current rates in terms of bits/second or bytes/seconds for multicast data, which would be useful for determining bandwidth provisioning for multicast traffic. However, the design engineer can still gain some visibility into the amount of multicast traffic crossing a particular interface on the Cisco Catalyst 6500 through the WS-SVC-NAM-2 port statistics.

If the WS-SVC-NAM-2 is installed within a Cisco Catalyst 6500 chassis that contains a Sup-32 PISA supervisor, you have the option of enabling Network-Based Application Recognition (NBAR) analysis of traffic forwarded through the supervisor, on a per-interface basis. NBAR adds the ability to analyze the traffic statistics collected through the supervisor at the protocol level. An example is shown in Figure 6-7.

Figure 6-7 Example of NAM Traffic Analyzer with NBAR Enabled



As before, the data can be viewed as current rates or as cumulative data. Individual protocol rates can also be graphed in real-time. As can be seen in Figure 6-7, NBAR has the ability to identify audio and video media as RTP streams, along with Real-Time Control Protocol (RTCP) control channels. NBAR can also identify signaling protocols, such as SIP. Therefore, NBAR can provide useful information regarding how much video traffic is crossing interfaces of the particular Cisco Catalyst 6500 chassis. This information may be used in determining whether sufficient bandwidth has been provisioned for a particular type of traffic, such as RTP. However, the combination of the NAM with NBAR still does not specifically identify a particular type of RTP flow as possibly being an IP video surveillance flow or a desktop video conferencing flow. Also, because different RTP flows from different video applications can be configured for different service classes, they may be placed into separate egress queues on the Cisco Catalyst 6500 switch ports. Therefore, simply knowing the aggregate bit rate of RTP flows through an interface still does not necessarily provide the level of detail to determine whether sufficient bandwidth is allocated per service class, and therefore per queue, on the particular Cisco Catalyst switch port. As is discussed in the next section, the NetFlow Data Export and SPAN monitoring functionality of the NAM can provide further detailed information to assist in determining whether sufficient bandwidth has been provisioned per service class.

NAM Analysis of NetFlow Traffic

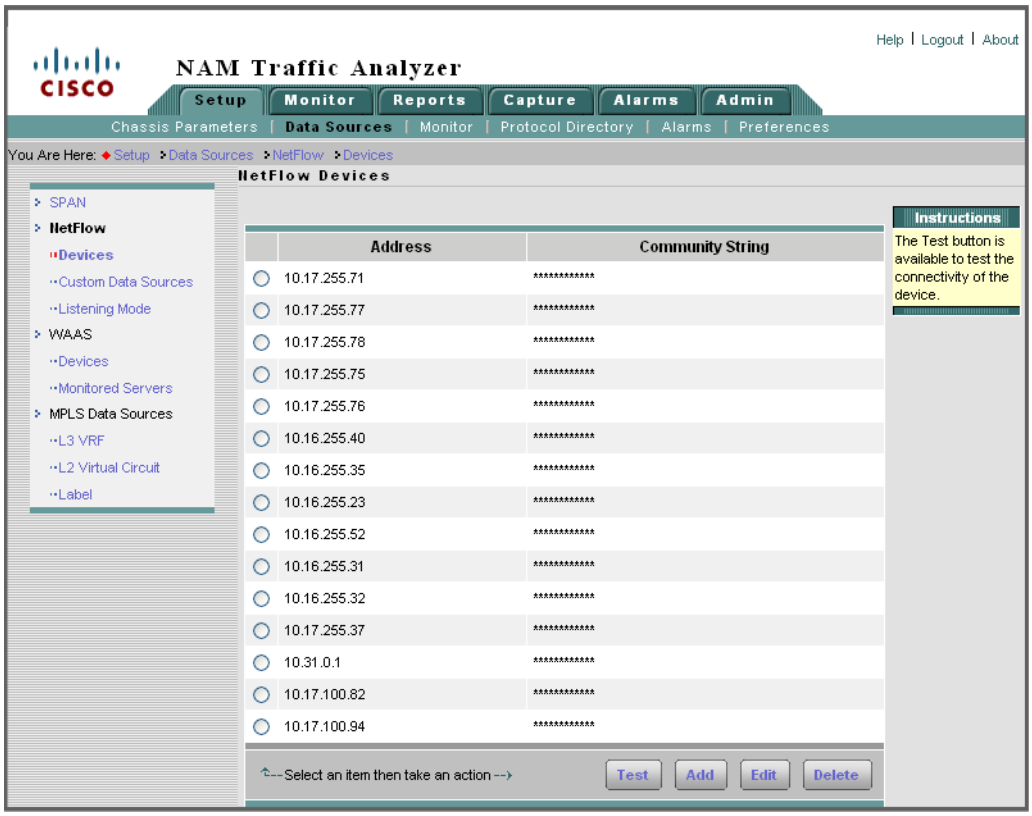
As mentioned in NetFlow Strategies Within an Enterprise Medianet, page 6-6, the NAM Traffic Analyzer can also function as a NetFlow collector. This allows the NAM to analyze traffic flows from remote devices within the enterprise medianet, without having to use the SPAN and RSPAN functionality

of Cisco Catalyst switches. Although NetFlow provides less information than a SPAN or RSPAN of the actual traffic, the overall bandwidth utilization can be significantly less, and NetFlow can therefore be far more scalable as a mechanism to view traffic flow data throughout a medianet. In this type of configuration, NetFlow traffic statistics can be collected from remote switches and routers throughout the enterprise medianet and forwarded to one or more WS-SVC-NAM-2 modules centrally located, perhaps within a Cisco Catalyst 6500 service switch within a campus data center service module. Alternatively NetFlow traffic may be forwarded to a NAM 2200 Series Appliance.

To configure NetFlow collector functionality within the NAM, each remote NetFlow Data Export (NDE) device must be added to the NetFlow Devices screen of the NAM web-based GUI, as shown in Figure 6-8. An optional SNMP v1/2c read-only community string can be configured to allow the NAM to include the configured description next to interface definitions.

Note Note that the NAM does not currently support SNMP v3.

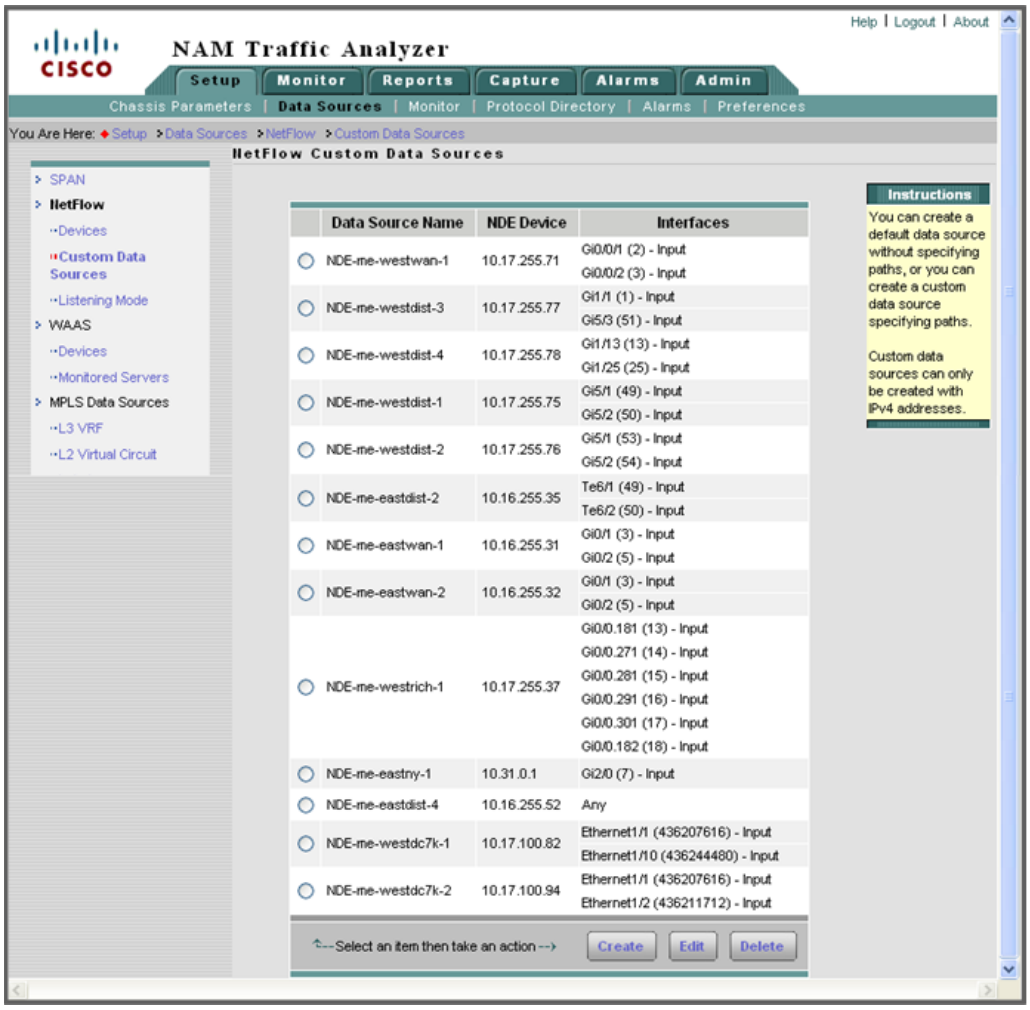
Figure 6-8 Configuration of NetFlow Devices within the NAM



The NAM allows multiple interfaces on a single physical device to be treated a single NetFlow custom data source. Interfaces from different devices cannot currently be aggregated into a single data source. This means that redundant pairs of switches or routers that load balance traffic (as shown in Figure 6-1) appear as multiple data sources to the NAM Traffic Analyzer. You may have to manually combine the results from the individual NetFlow data sources to gain an understanding of the total traffic flows through a given set of redundant devices. The exception to this is if VSS is deployed across a pair of redundant Cisco Catalyst 6500 switches. VSS allows a redundant pair of Cisco Catalyst 6500s to appear

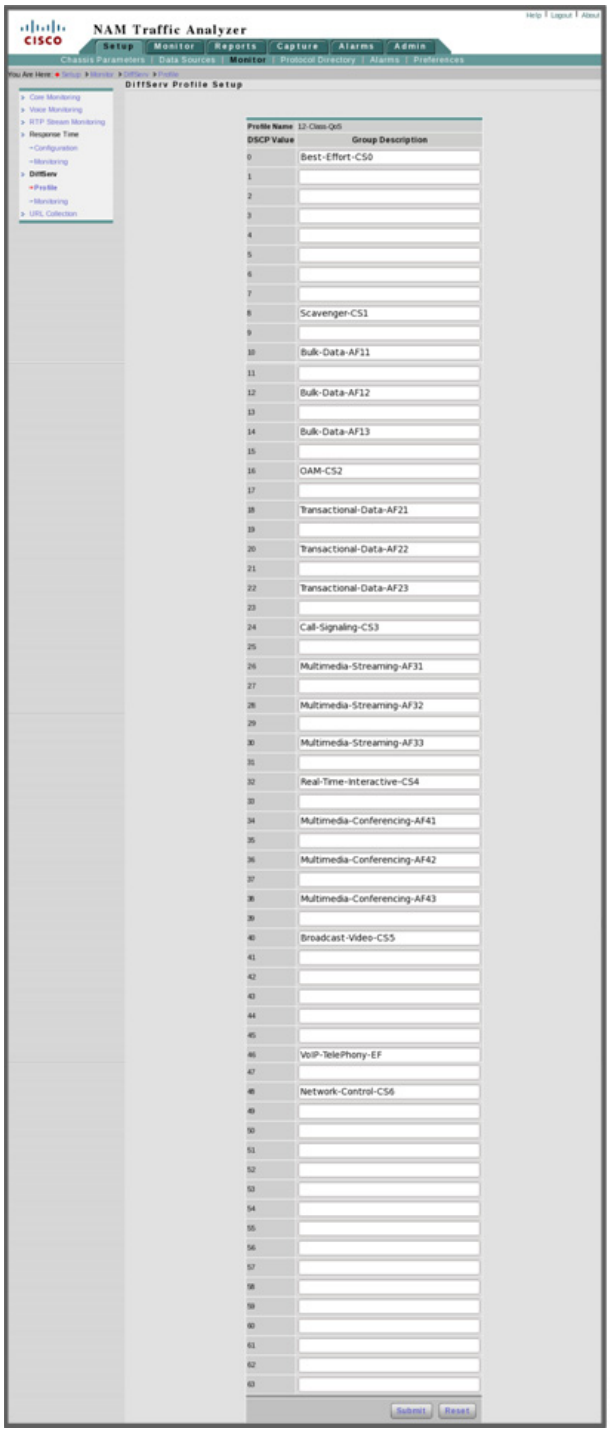
as a single device. Therefore, the NetFlow statistics from multiple interfaces on both switches can appear as a single data set. Figure 6-9 shows an example of how multiple interfaces on a single device are aggregated into a single NetFlow custom data source on the NAM.

Figure 6-9 Configuration of Custom Data Sources on the NAM



From a medianet management perspective, one of the attractive features of the NAM as a NetFlow collector is its ability to monitor and generate reports on traffic flows, based on their DSCP values. If the various video applications running over the network are separated into different service classes, gaining visibility into the amount of traffic per service class allows you to gain visibility into the amount of traffic that a particular application is generating across key parts of the medianet infrastructure. To accomplish this, you first need to create a diffserv aggregation profile that maps traffic with different DSCP values into aggregation groups for reporting. An example of an aggregation profile based on the Cisco enterprise 12-class QoS model is shown in Figure 6-10.

Figure 6-10 Diffserv Profile Based on the Cisco Enterprise 12-Class QoS Model

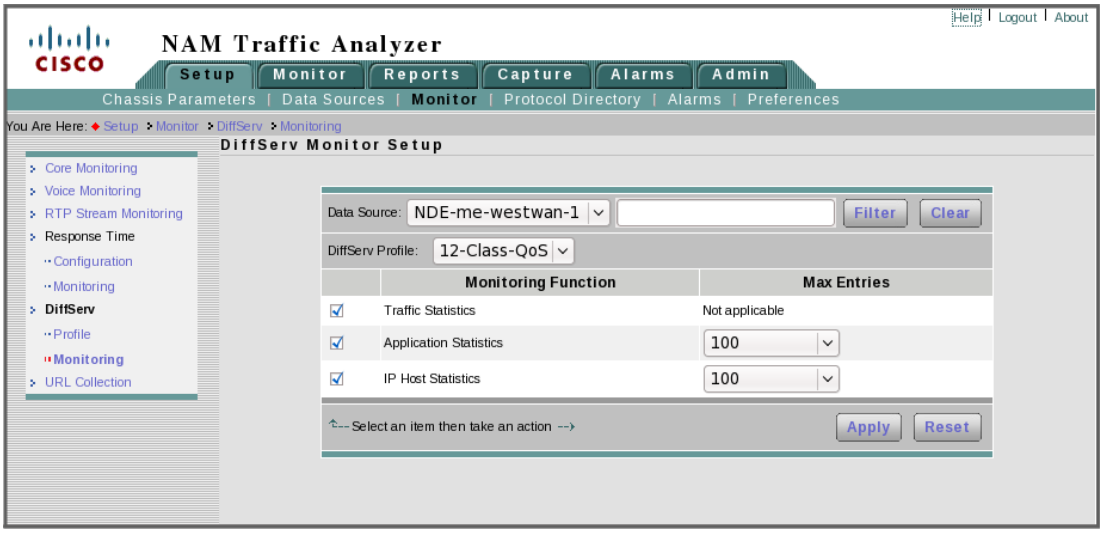


As can be seen, each of the DSCP markings corresponds to one of the 12 QoS classes. Because assured forwarding (AF) traffic may be marked down (for example from AFx1 to AFx2 or AFx3) within service provider Multiprotocol Label Switching (MPLS) networks, these have been added to the diffserv aggregation profile as separate aggregation groups in the example above. This can provide additional value in that you may be able determine whether traffic within the particular assured forwarding traffic classes is being marked down because the traffic rates are outside the contracted rates of the service

provider network. The downside to this approach, however, is that you may have to manually combine the monitoring and reporting statistics from the separate aggregation groups to gain a view of all the traffic within a single assured forwarding (AFx1, AFx2, and AFx3) class. Alternatively, the AFx1, AFx2, and AFx3 traffic can be placed into a single aggregation group (for instance AF41, AF42, and AF42 all placed into a multimedia-conferencing group). This makes it easier to view the overall amount of traffic within a particular AF class, but at the loss of the information regarding if or how much of the traffic was marked down.

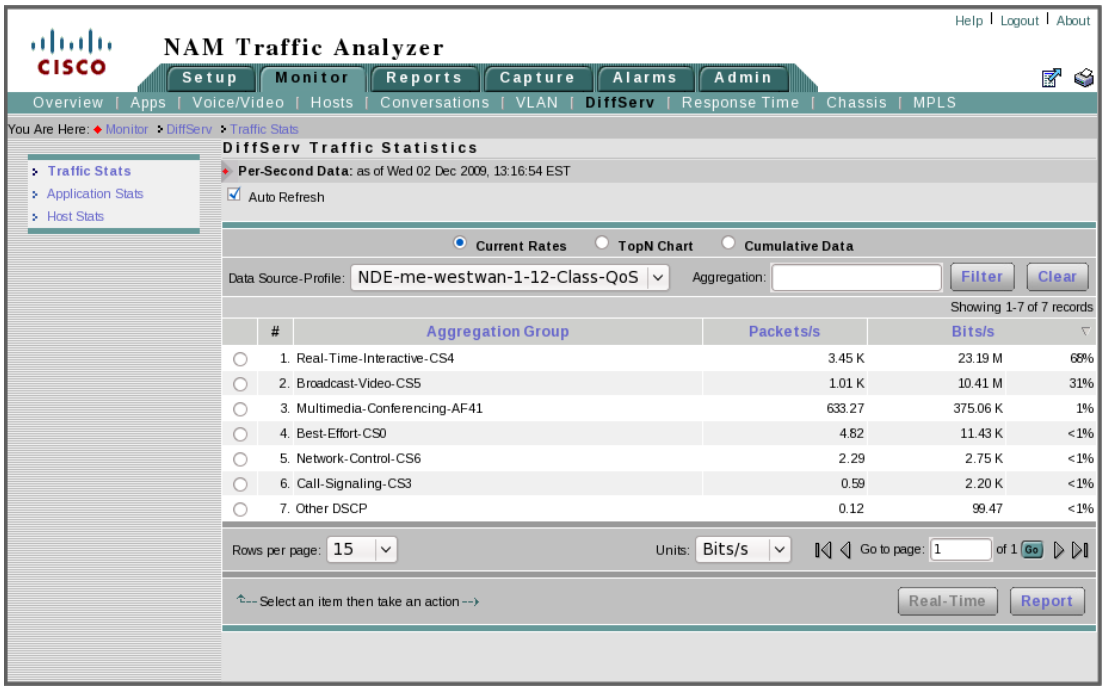
After the diffserv aggregation profile has been created, it must be applied to monitor each data source in which it is desired to see traffic statistics, application statistics, and/or host statistics; based on the aggregation groupings defined within in the profile. An example of this is shown in Figure 6-11, in which the 12-Class-QoS diffserv profile has been applied to the NDE source corresponding to a WAN edge router.

Figure 6-11 Application of the Diffserv Aggregation Profile to an NDE Source



When applied, the traffic, application, and/or IP host statistics can be viewed as current rates or cumulative data. Figure 6-12 shows an example of the output from the traffic statistics shown as current rates.

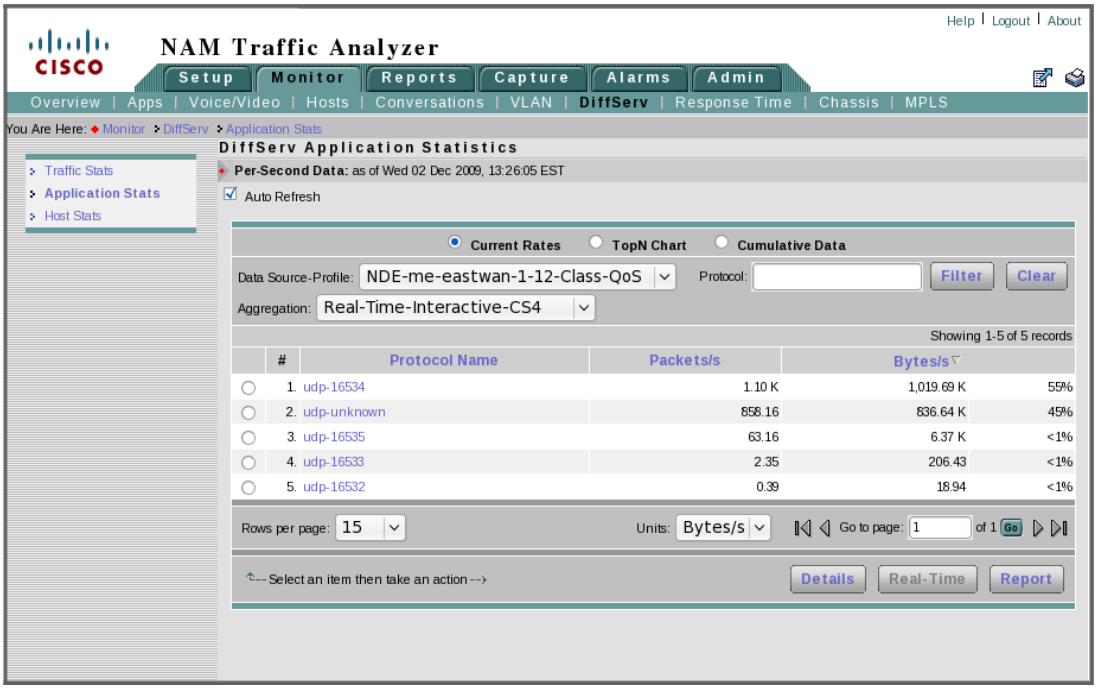
Figure 6-12 Traffic Statistics per Service Class from an NDE Source



The example above shows the breakout of traffic flows from a campus core to a WAN edge switch (as shown in Figure 6-1). This level of traffic analysis may be used to assist in determining whether the provisioning of traffic on existing WAN policy maps is appropriate for the actual traffic levels that cross the WAN interfaces. Policy maps on Cisco IOS router platforms are often configured to allow applications to exceed the allocated bandwidth for a particular service class, if available bandwidth exists on the WAN link. Therefore, just because no drops are being seen on a particular service class on a WAN link, does not mean the provisioned bandwidth is sufficient for the traffic within that service class. The service class may be borrowing from other service classes. Visibility into the amount of actual traffic flows per service class can help ensure that you allocate the appropriate amount of bandwidth per service class.

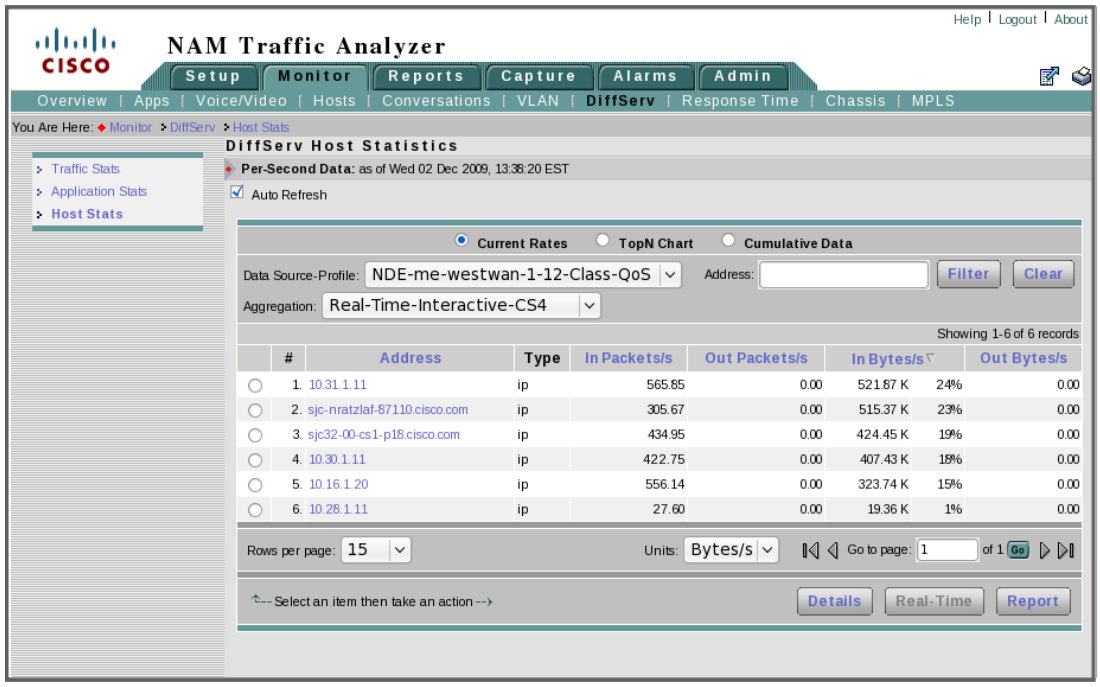
You can drill down further into each service class to identify particular application flows, based on their TCP or UDP port numbers. This is done through the Diffserv Application Statistics screen, as shown in Figure 6-13.

Figure 6-13 Application Statistics per Service Class from an NDE Source



Here, note again what was previously mentioned in NetFlow Collector Considerations, page 6-7. The NAM itself cannot identify the particular application flows per service class as being IP video surveillance flows, TelePresence flows, or VoD flows. However, if different video applications are separated into different service classes, you may be able to determine to which video application the flows belong. For example, in the network used for the example in Figure 6-13, only Cisco TelePresence traffic was placed in the Real-Time Interactive service class. Therefore, you can easily identify that the flows within Figure 6-13 represent TelePresence meetings. By selecting any one of the flows and clicking the Details button, you can see the host IP addresses that generated the flows. Alternatively, you can drill down into each service class to identify particular hosts responsible for the flows, based on their IP addresses. This is done through the Diffserv Application Hosts screen, as shown in Figure 6-14.

Figure 6-14 Host Statistics per Service Class from an NDE Source

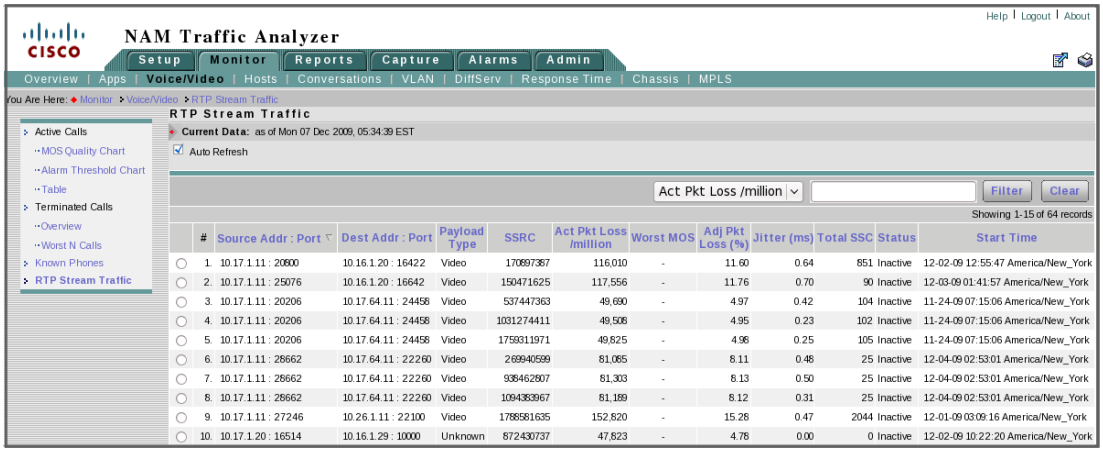


Note that if the particular device is an application-specific video device, such as a Cisco TelePresence System endpoint or an IP video surveillance camera, DNS address translation may be useful to provide a meaningful name that indicates the type of video device instead of an IP address.

NAM Analysis of SPAN/RSPAN Traffic

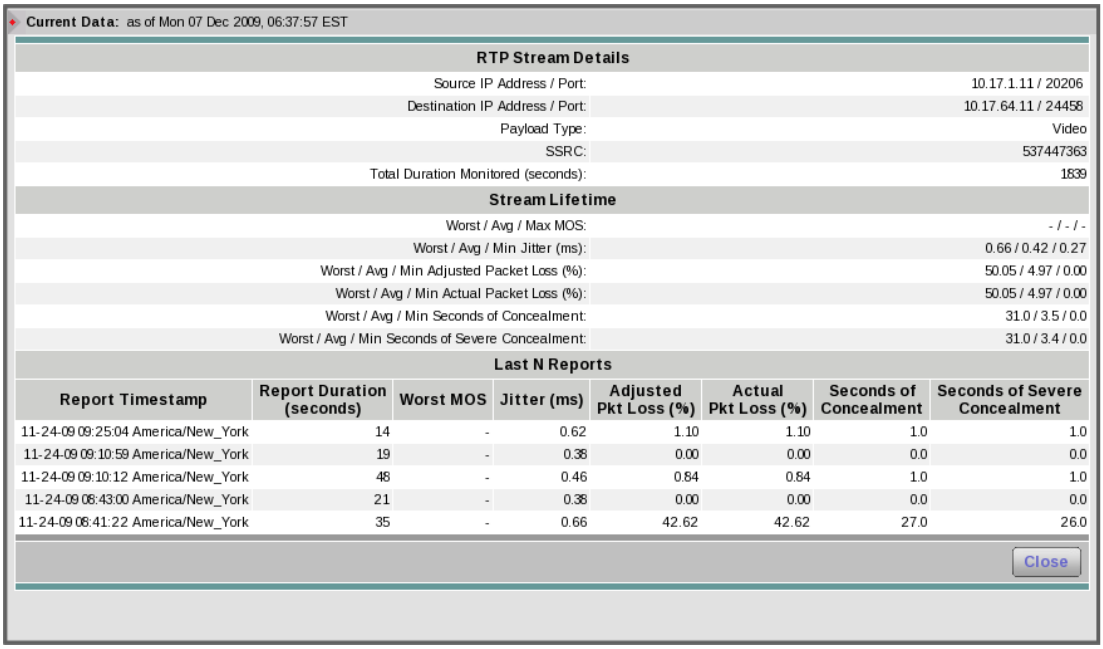
When configured to analyze traffic that has been sent to the WS-SVC-NAM-2 via the Cisco Catalyst 6500 SPAN or RSPAN features, the NAM provides the same ability to monitor and generate reports for traffic based on service class, as was discussed in the previous section. In addition, the NAM can provide more detailed monitoring of RTP streams included within the SPAN or RSPAN traffic flows. An example of the RTP stream traffic is shown in Figure 6-15.

Figure 6-15 NAM RTP Stream Traffic



As can be seen from Figure 6-15, the NAM has the ability to collect detailed performance data from RTP flows down to individual synchronization sources (SSRCs), including packet loss counts for the session, packet loss percentages for the session, jitter, and whether the RTP session is still active. Further information can be viewed by highlighting and selecting the details regarding each individual flow, as shown in Figure 6-16.

Figure 6-16 RTP Flow Details



Here you can view the flow in increments over time, to see whether the packet loss and high jitter levels were a one-time event during the session, or were continuous throughout the session. This level of detail can be used to assist in identifying performance issues within the network down to the level of individual cameras within a multi-screen (CTS-3000) TelePresence meeting.

Cisco IP Service Level Agreements

Cisco IPSLAs are an active traffic monitoring utility for measuring network performance. IPSLA support is included within most Cisco IOS router platforms, Cisco Catalyst switch platforms (including the Cisco Catalyst 6500, Cisco Catalyst 4500, and Cisco Catalyst 3750E Series), and some Cisco video endpoints such as Cisco TelePresence Systems endpoints. IPSLAs operate in a sender/responder configuration. Typically a Cisco IOS router or switch platform is configured as a source (the IPSLA sender) of packets, otherwise known as IPSLA probes, which are crafted specifically to simulate a particular IP service on the network. These packets are sent to the remote device (the IPSLA responder), which may loop the packets back to the IPSLA Sender. In this manner, enterprise medianet service level parameters such as latency, jitter, and packet loss can be measured. There are a variety of Cisco IPSLA operations, meaning that various types of IP packets can be generated by the IPSLA sender and returned by the IPSLA responder. Depending on the particular platform, these can include the following operations:

- UDP jitter
- ICMP path jitter
- UDP jitter for VoIP
- UDP echo
- ICMP echo
- ICMP path echo
- HTTP
- TCP connect
- FTP
- DHCP
- DNS
- Data Link Switching Plus (DLSW+)
- Frame Relay

For a discussion of each of the different IPSLA operations and how to configure them on Cisco IOS router platforms, see the *Cisco IOS IPSLAs Configuration Guide, Release 12.4* at the following URL: http://www.cisco.com/en/US/docs/ios/12_4/ip_sla/configuration/guide/hsla_c.html.

IPSLAs as a Pre-Assessment Tool

From an FCAPS management perspective, IPSLAs are most applicable as a performance management tool within an enterprise medianet. They can be used to pre-assess the ability of the IP network infrastructure to support a new service, such as the deployment of Cisco TelePresence, between two points within the network. Because most video flows are RTP-based, the UDP jitter IPLSA operation typically has the most relevance from a medianet pre-assessment perspective.

Note Note that many video flows use other transport protocols. For example, both VoD and MJPEG-based IP video surveillance may use HTTP as the transport protocol instead of RTP.

The usefulness of IPSLAs as a pre-assessment tool depends to a large extent on the knowledge of the medianet video flow that is to be simulated by IPSLA traffic, and whether an IPSLA operation can be crafted to accurately replicate the medianet video flow. This can be particularly challenging for high

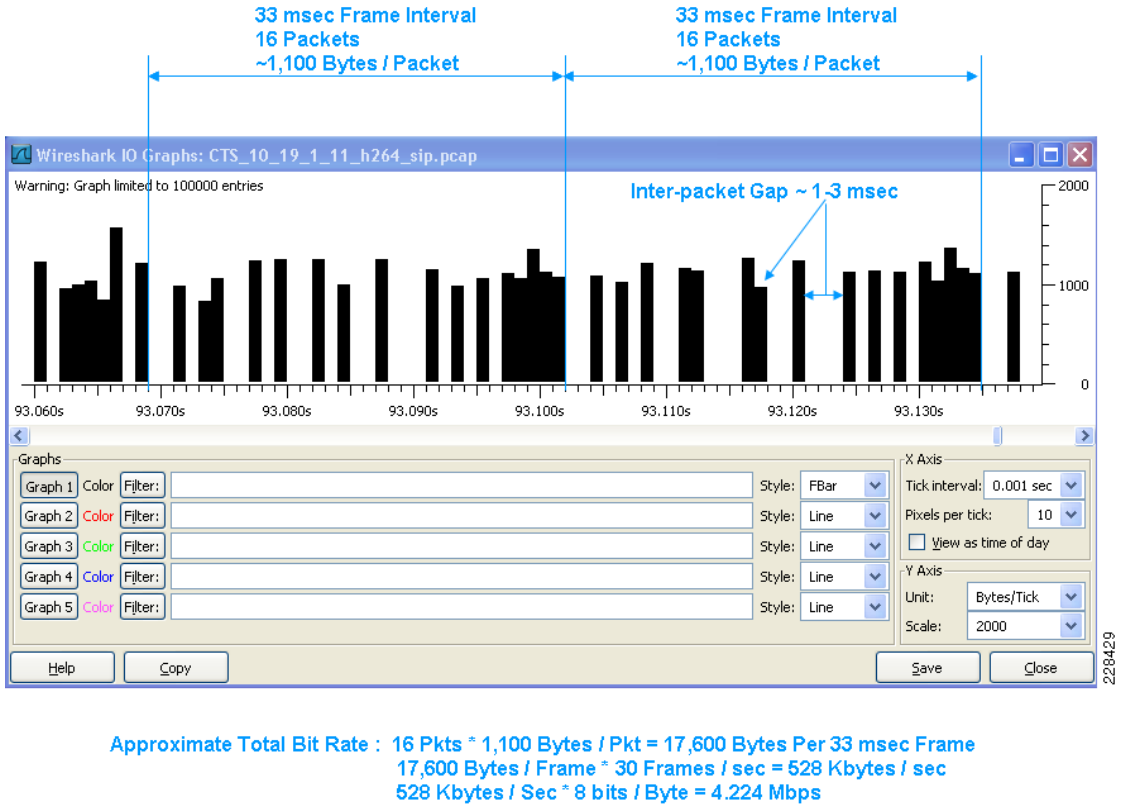
definition video for several reasons. First, video flows are sent as groups of packets every frame interval. These groups of packets can be bunched-up at the beginning of the frame interval, or spread evenly across the frame interval, depending on how the video application (that is, the transmitting codec) is implemented. Also, each packet within a single frame can vary in size. Operations such as the UDP jitter IPSLA operation transmit fixed-sized packets at regular intervals, similarly to VoIP. Second, high definition video frames often consist of more than ten packets per frame, meaning that the interval between the individual packets sent within a single video frame can vary from less than one millisecond to several milliseconds. Observations on a highly loaded Cisco Catalyst 6500 with a Sup-32 processor have shown that individual UDP jitter IPSLA operations can generate packets with intervals of 4 milliseconds or greater with large packet payload sizes. Smaller platforms such as the Cisco 2800 Series ISR may be capable of generating packets with intervals of only 8–12 milliseconds or greater, depending on the loading of the platform.

Note Although some platforms allow configuration of intervals down to one millisecond, the design engineer may find it necessary to capture a data trace of the IPSLA probes to determine the actual frame rate generated by the IPSLA sender. Partly because the loading on the CPU affects the rate at which IPSLA probes are generated, pre-assessments services of deployments such as Cisco TelePresence are often performed with dedicated ISR router platforms. Likewise, some organizations deploy router platforms at campus and branch locations dedicated for IPSLA functions.

Crafting one or more UDP jitter IPSLA operations that accurately replicate the size of the individual packets sent, the interval between individual packets sent, and the frame-based nature of video can be challenging. These attributes are important to factor in because network parameters such as jitter and packet loss are often largely dependent on the queue depths and buffer sizes of networking gear along the path between the endpoints. Sending a smooth flow of evenly spaced packets, or larger packets less frequently, may result in significantly different results than the actual video flows themselves.

As an example, to accurately pre-assess the ability of the network to handle a flow such as a TelePresence endpoint, you must craft a sequence of packets that accurately simulates the endpoint. Figure 6-17 shows a close-up of the graph from a data capture of the video stream from a Cisco TelePresence CTS-1000 running Cisco TelePresence System version 1.6 software.

Figure 6-17 Detailed Graph of a CTS-1000 Video Stream (Version 1.6)

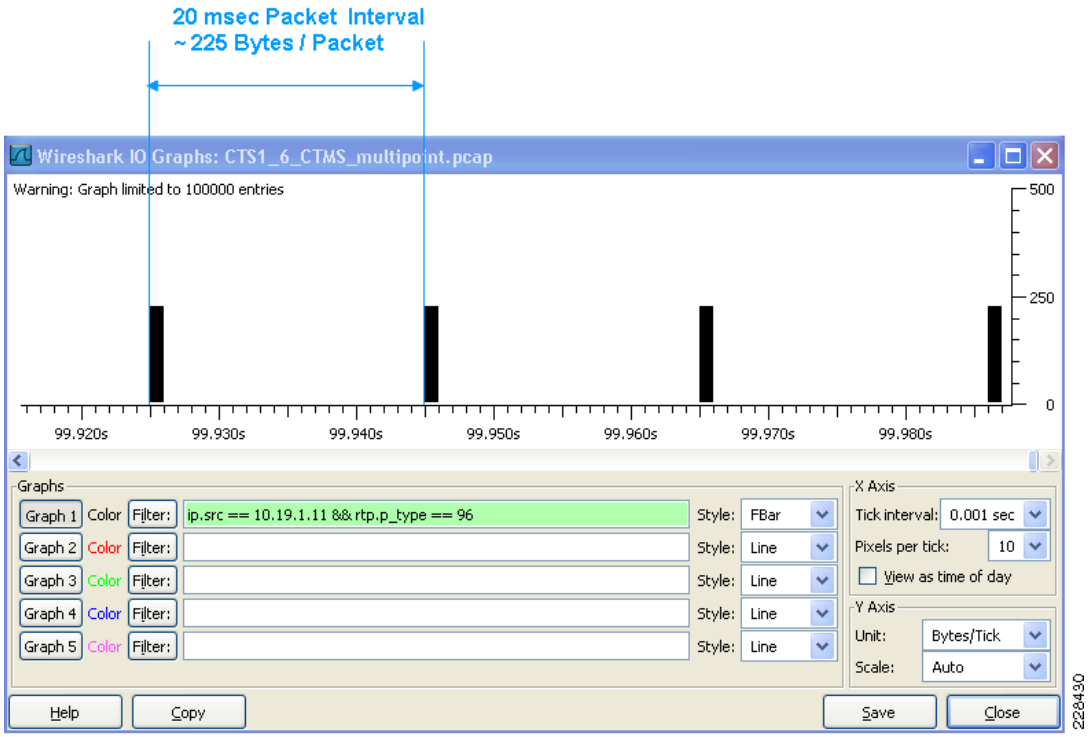


Approximate Total Bit Rate : 16 Pkts * 1,100 Bytes / Pkt = 17,600 Bytes Per 33 msec Frame
17,600 Bytes / Frame * 30 Frames / sec = 528 Kbytes / sec
528 Kbytes / Sec * 8 bits / Byte = 4.224 Mbps

As can be seen from Figure 6-17, TelePresence video packets average slightly under 1100 bytes in size. Each video frame consists of approximately 16 packets, spaced from 1–3 msec apart, spread across the 33 msec frame interval. Based on this analysis, a UDP jitter IP SLA operation consisting of 1060 byte packets with a interval of 2 msec between packets, sent with a ToS value equivalent to CS4 traffic, would simulate the size and packet rate of a TelePresence video stream. The overall data rate would be approximately 1060 bytes/packet * 500 packets/sec * 8 bits/byte = 4.24 Mbps.

Figure 6-18 shows a close-up of the graph from a data capture of a single audio stream from a TelePresence CTS-1000 running Cisco TelePresence System version 1.6 software.

Figure 6-18 Detailed Graph of a CTS-1000 Audio Stream (Version 1.6)

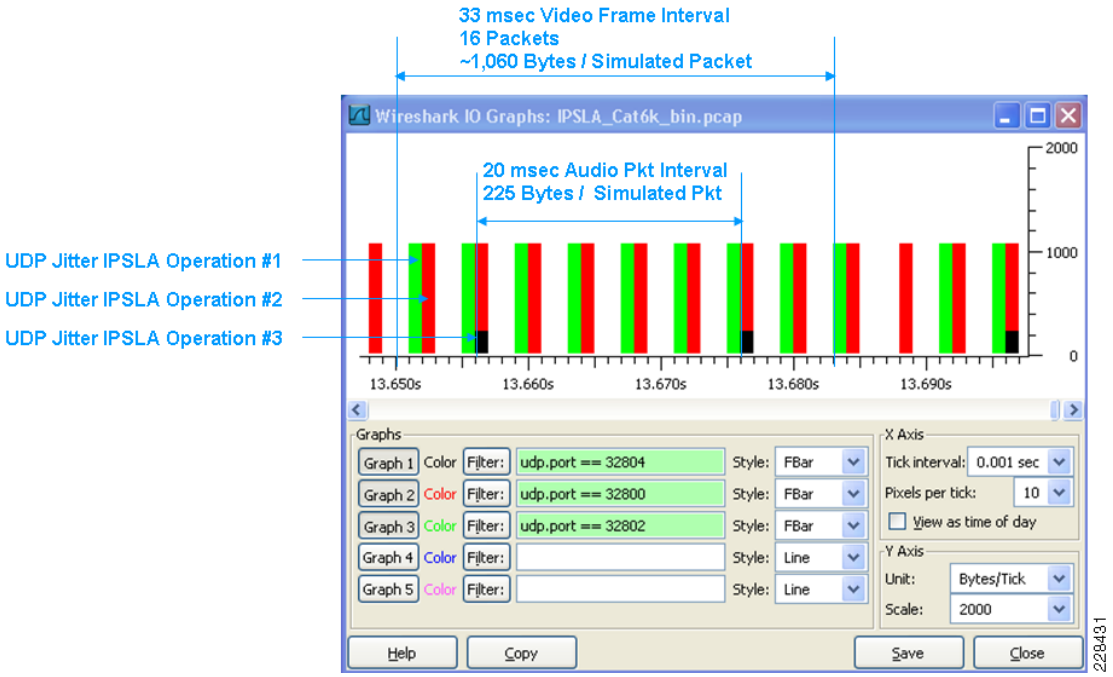


Approximate Total Bit Rate : 225 Bytes / Packet * 50 Packets / Sec
11,250 Bytes / Sec * 8 Bits / Byte = 90 Kbps

As shown in Figure 6-18, TelePresence audio packets are approximately 225 bytes in size, sent every 20 msec. Based on this analysis, a UDP jitter IP SLA operation consisting of 225-byte packets with a interval of 20 msec between packets, sent with a ToS value equivalent to CS4 traffic (because Cisco TelePresence sends audio with the same marking as video) would simulate the size and packet rate of a single TelePresence audio stream. The overall data rate would be approximately 225 bytes/packet * 50 packets/sec * 8 bits/byte = 90 Kbps.

As previously mentioned, however, a lightly loaded Cisco Catalyst 6500 with Sup-32 processor was observed to be able to generate packets with a minimum packet interval of only 4 milliseconds. Therefore, one method of simulating the number of packets and their sizes within the TelePresence video stream is to implement two UDP jitter IPSLA operations on the Cisco Catalyst 6500, each with a packet interval of 4 milliseconds, and to schedule them to start simultaneously. A third UDP jitter IPSLA operation can also be run simultaneously to simulate the audio stream. Figure 6-19 shows a close-up of the graph from a data capture of the actual data stream from these UDP jitter IPSLA operations.

Figure 6-19 Detailed Graph of Multiple UDP Jitter IPSLA Operations Simulating a Cisco TelePresence CTS-1000



From Figure 6-19, it appears that UDP jitter IPSLA operations #1 and #2 space their packets each 1 millisecond apart. However, this is just because of the graphing of the data points. The actual data trace reveals that the Cisco Catalyst 6500 switch sends packets from both UDP jitter IPSLA operations roughly back-to-back every four milliseconds. Therefore, the IPSLA-simulated video packets are slightly more clumped together than actual TelePresence video packets, but still considered acceptable from a pre-assessment perspective. The third UDP jitter IPSLA operation generates a simulated audio stream of packets every 20 milliseconds. Note that a CTS-1000 can receive up to three audio streams and an additional auxiliary video stream for presentations. Simulation of these streams are not shown in this example for simplicity. However, the method discussed above can be extended to include those streams as well if needed. Likewise, the method may be used to simulate other video flows simply by capturing a data trace, analyzing the flow, and setting up the appropriate IPSLA operations.

The configuration snippet on Example 6-3 shows the configuration of the UDP jitter IPSLA operations on the Cisco Catalyst 6500 switch that were used to create the simulation from which the data in Figure 6-19 was captured.

Example 6-3 IPSLA Sender Configuration on a Cisco Catalyst 6500 Series Switch

```
ip sla monitor 24
 type jitter dest-ipaddr 10.24.1.11 dest-port 32800 source-ipaddr 10.16.1.1 source-port
32800 num-packets 16500 interval 2
 request-data-size 1018
 tos 128
ip sla monitor 25
 type jitter dest-ipaddr 10.24.1.11 dest-port 32802 source-ipaddr 10.16.1.1 source-port
32802 num-packets 16500 interval 2
 request-data-size 1018
 tos 128
ip sla monitor 26
 type jitter dest-ipaddr 10.24.1.11 dest-port 32804 source-ipaddr 10.16.1.1 source-port
32804 num-packets 3300 interval 20
```

```
 request-data-size 183
 tos 128
!
ip sla monitor group schedule 1 24,25,26 schedule-period 1 frequency 70 start-time now
life 700
!
~
```

Even though the packet interval has been configured at 2 milliseconds for *ip sla monitor 25* and *ipsla monitor 26*, the real interval between packets was observed to be 4 milliseconds. Sending 16,500 packets spaced at 4 milliseconds apart takes approximately 66 seconds. The configuration of *ip sla monitor group schedule 1* with a schedule period of one second causes the three UDP jitter operations to simultaneously start. The frequency of 70 seconds ensures that the previous operations complete before they begin again. The operation was set to run for approximately 10 intervals, or 700 seconds. Note that the length of time needed to perform a real assessment of a network to support a service such as a CTS-1000 is completely at the discretion of the network administrator. The aggregated output from the IPSLA tests can be displayed via the **show ip sla monitor statistics aggregated details** command on the Cisco Catalyst 6500 switch, as shown in Example 6-4. It shows the packet loss; minimum and maximum jitter; and minimum, maximum and average latency for each of the three UDP jitter IPSLA operations.

Example 6-4 IPSLA Aggregated Statistics on a Cisco Catalyst 6500 Series Switch

```
me-eastdc-1#show ip sla monitor statistics aggregated details
Round trip time (RTT)   Index 24
Start Time Index: .10:53:07.852 EST Mon Nov 16 2009
Type of operation: jitter
Voice Scores:
MinOfICPIF: 0   MaxOfICPIF: 0   MinOfMOS: 0     MaxOfMOS: 0
RTT Values
        Number Of RTT: 94674
        RTT Min/Avg/Max: 1/1/7
Latency one-way time milliseconds
        Number of Latency one-way Samples: 0
        Source to Destination Latency one way Min/Max: 0/0
        Destination to Source Latency one way Min/Max: 0/0
        Source to Destination Latency one way Sum/Sum2: 0/0
        Destination to Source Latency one way Sum/Sum2: 0/0
Jitter time milliseconds
        Number of Jitter Samples: 94664
        Source to Destination Jitter Min/Max: 1/4
        Destination to Source Jitter Min/Max: 1/6
        Source to destination positive jitter Min/Avg/Max: 1/1/4
        Source to destination positive jitter Number/Sum/Sum2: 1781/1849/2011
        Source to destination negative jitter Min/Avg/Max: 1/1/4
        Source to destination negative jitter Number/Sum/Sum2: 1841/1913/2093
        Destination to Source positive jitter Min/Avg/Max: 1/1/6
        Destination to Source positive jitter Number/Sum/Sum2: 3512/3532/3632
        Destination to Source negative jitter Min/Avg/Max: 1/1/6
        Destination to Source negative jitter Number/Sum/Sum2: 3447/3468/3570
        Interarrival jitterout: 0       Interarrival jitterin: 0
Packet Loss Values
        Loss Source to Destination: 0           Loss Destination to Source: 0
        Out Of Sequence: 0      Tail Drop: 0    Packet Late Arrival: 0
Number of successes: 10
Number of failures: 0
Failed Operations due to over threshold: 0
Failed Operations due to Disconnect/TimeOut/Busy/No Connection: 0/0/0/0
Failed Operations due to Internal/Sequence/Verify Error: 0/0/0
Distribution Statistics:
Bucket Range: 0-19 ms
```

```
Avg. Latency: 0 ms
Percent of Total Completions for this Range: 100 %
Number of Completions/Sum of Latency: 10/3
Sum of RTT squared low 32 Bits/Sum of RTT squared high 32 Bits: 3/0
Operations completed over thresholds: 0

Round trip time (RTT)   Index 25
Start Time Index: .10:53:07.856 EST Mon Nov 16 2009
Type of operation: jitter
Voice Scores:
MinOfICPIF: 0   MaxOfICPIF: 0   MinOfMOS: 0     MaxOfMOS: 0
RTT Values
        Number Of RTT: 94672
        RTT Min/Avg/Max: 1/1/8
Latency one-way time milliseconds
        Number of Latency one-way Samples: 0
        Source to Destination Latency one way Min/Max: 0/0
        Destination to Source Latency one way Min/Max: 0/0
        Source to Destination Latency one way Sum/Sum2: 0/0
        Destination to Source Latency one way Sum/Sum2: 0/0
Jitter time milliseconds
        Number of Jitter Samples: 94662
        Source to Destination Jitter Min/Max: 1/4
        Destination to Source Jitter Min/Max: 1/7
        Source to destination positive jitter Min/Avg/Max: 1/1/3
        Source to destination positive jitter Number/Sum/Sum2: 2498/2559/2691
        Source to destination negative jitter Min/Avg/Max: 1/1/4
        Source to destination negative jitter Number/Sum/Sum2: 2553/2620/2778
        Destination to Source positive jitter Min/Avg/Max: 1/1/7
        Destination to Source positive jitter Number/Sum/Sum2: 4470/4511/4725
        Destination to Source negative jitter Min/Avg/Max: 1/1/6
        Destination to Source negative jitter Number/Sum/Sum2: 4413/4448/4622
        Interarrival jitterout: 0       Interarrival jitterin: 0
Packet Loss Values
        Loss Source to Destination: 0           Loss Destination to Source: 0
        Out Of Sequence: 0      Tail Drop: 0    Packet Late Arrival: 0
Number of successes: 10
Number of failures: 0
Failed Operations due to over threshold: 0
Failed Operations due to Disconnect/TimeOut/Busy/No Connection: 0/0/0/0
Failed Operations due to Internal/Sequence/Verify Error: 0/0/0
Distribution Statistics:
Bucket Range: 0-19 ms
Avg. Latency: 0 ms
Percent of Total Completions for this Range: 100 %
Number of Completions/Sum of Latency: 10/5
Sum of RTT squared low 32 Bits/Sum of RTT squared high 32 Bits: 5/0
Operations completed over thresholds: 0

Round trip time (RTT)   Index 26
Start Time Index: .10:53:02.892 EST Mon Nov 16 2009
Type of operation: jitter
Voice Scores:
MinOfICPIF: 0   MaxOfICPIF: 0   MinOfMOS: 0     MaxOfMOS: 0
RTT Values
        Number Of RTT: 16500
        RTT Min/Avg/Max: 1/1/8
Latency one-way time milliseconds
        Number of Latency one-way Samples: 0
        Source to Destination Latency one way Min/Max: 0/0
        Destination to Source Latency one way Min/Max: 0/0
        Source to Destination Latency one way Sum/Sum2: 0/0
        Destination to Source Latency one way Sum/Sum2: 0/0
Jitter time milliseconds
```

```
        Number of Jitter Samples: 16490
        Source to Destination Jitter Min/Max: 1/4
        Destination to Source Jitter Min/Max: 1/6
        Source to destination positive jitter Min/Avg/Max: 1/1/4
        Source to destination positive jitter Number/Sum/Sum2: 440/457/505
        Source to destination negative jitter Min/Avg/Max: 1/1/4
        Source to destination negative jitter Number/Sum/Sum2: 496/512/558
        Destination to Source positive jitter Min/Avg/Max: 1/1/6
        Destination to Source positive jitter Number/Sum/Sum2: 571/587/679
        Destination to Source negative jitter Min/Avg/Max: 1/1/6
        Destination to Source negative jitter Number/Sum/Sum2: 513/529/621
        Interarrival jitterout: 0       Interarrival jitterin: 0
Packet Loss Values
        Loss Source to Destination: 0           Loss Destination to Source: 0
        Out Of Sequence: 0      Tail Drop: 0    Packet Late Arrival: 0
Number of successes: 10
Number of failures: 0
Failed Operations due to over threshold: 0
Failed Operations due to Disconnect/TimeOut/Busy/No Connection: 0/0/0/0
Failed Operations due to Internal/Sequence/Verify Error: 0/0/0
Distribution Statistics:
Bucket Range: 0-19 ms
Avg. Latency: 1 ms
Percent of Total Completions for this Range: 100 %
Number of Completions/Sum of Latency: 10/10
Sum of RTT squared low 32 Bits/Sum of RTT squared high 32 Bits: 10/0
Operations completed over thresholds: 0
```

For the example above, the IPSLA responder was an actual Cisco TelePresence CTS-1000. Only IPSLA responder operations can be configured on Cisco TelePresence System endpoints; they cannot function as IPSLA sources. Configuration is only via the SSH CLI, as shown in Example 6-5.

Example 6-5 IPSLA Responder Configuration on a CTS-1000

```
admin: utils ipsla responder initiators add net 10.16.1.0/24
admin: utils ipsla responder enable start
```

The configuration above enables the IPSLA responder function for initiators (senders) on the 10.16.1.0/24 subnet. This corresponds to the source of the IPSLA packets from the Cisco Catalyst 6500. By default, the range of ports enabled on the CTS-1000 is from 32770 to 33000. However, the port range can be enabled by including start and end ports within the **utils ipsla responder enable** command. For a discussion of all the commands available via the SSH CLI, including all the IPSLA commands, see the *Cisco TelePresence System Release 1.6 Command-Line Interface Reference Guide* at the following URL: http://www.cisco.com/en/US/docs/telepresence/cts_admin/1_6/CLI/cts1_6cli.html.

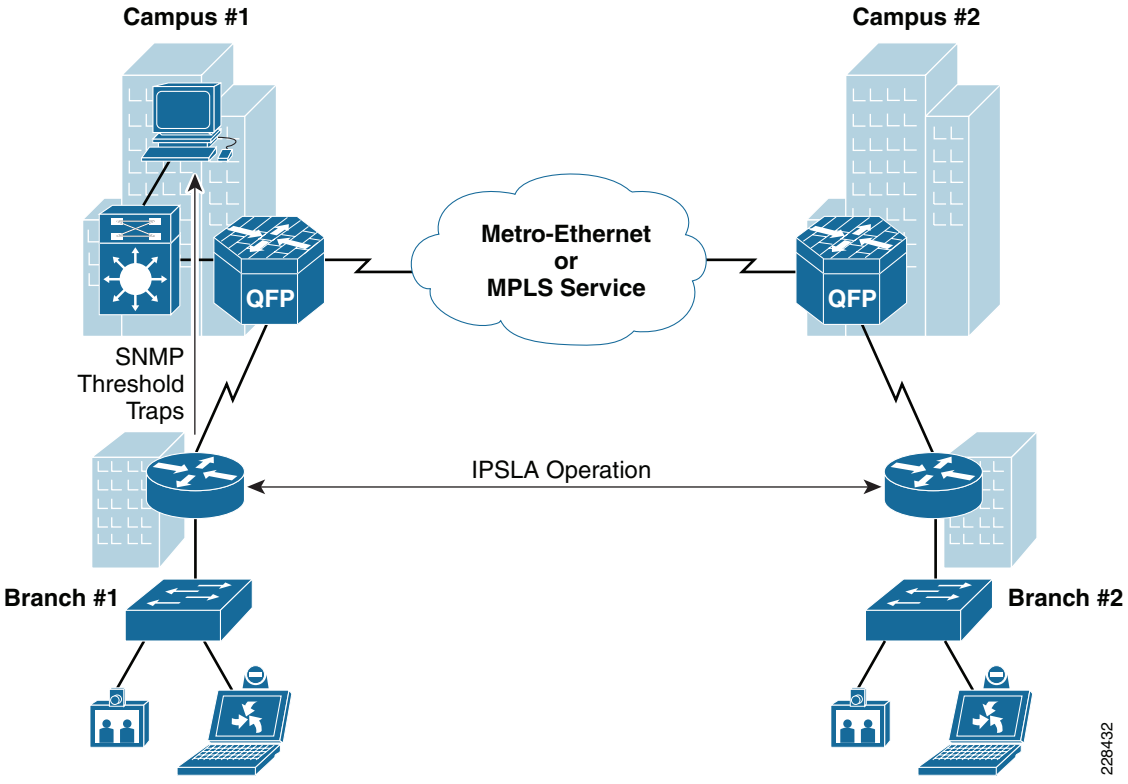
The use of IPSLA as a pre-assessment tool can be disruptive to existing traffic on the IP network infrastructure. After all, the objective of the pre-assessment test is to see whether the network infrastructure can support the additional service. For example, if a particular link within the network infrastructure has insufficient bandwidth, or a switch port has insufficient buffering capacity to support existing TelePresence traffic as well as the additional traffic generated from the IPSLA pre-assessment tests, both the existing TelePresence call and the IPSLA operation show degraded quality during the tests. You must therefore balance the possibility of temporarily degrading production services on the network against the value of the information gathered from running an IPSLA pre-assessment test during normal business hours. Running the IPSLA tests after hours may not accurately assess the ability of the network to handle the additional service, because after-hour traffic patterns may vary significantly from traffic patterns during normal business hours. Further, running a successful pre-assessment test after hours may lead to the installation of a production system that then results in degraded quality both for itself and for other production systems during normal business hours.

Finally, when multiple redundant equal-cost paths exist within the medianet infrastructure, Cisco Express Forwarding (formerly known as CEF) load balances the traffic across the equal-cost paths using a hash of the source and destination IP addresses for each session. Each router and switch along the path independently creates its Cisco Express Forwarding table based on IP routing protocols, and load balances sessions across its interfaces that represent equal-cost paths to the next hop along the path to the destination. An IPSLA probe generated by a switch or router has a different IP source address than the actual video device that is being pre-assessed. Therefore, the path taken by the IPSLA probes within a highly redundant network infrastructure may not be exactly the path taken by the actual video traffic from the device. The use of dedicated routers to perform an IPSLA network assessment eases this issue slightly, because the routers can be configured to use the actual IP addresses that the video endpoints will ultimately use. However, any changes to the Cisco Express Forwarding tables, brought about through routing changes or reloading of the switches/routers along the path, may result in a slightly different path established for the traffic when the actual video devices are installed. You should be aware of these limitations of IPSLA within a highly redundant medianet infrastructure.

IPSLA as an Ongoing Performance Monitoring Tool

If configured with careful consideration, IPSLAs can also be used as an ongoing performance monitoring tool. Rather than simulating an actual medianet video flow, IPSLA operations can be used to periodically send small amounts of traffic between two points within the network, per service class, to assess parameters such as packet loss, one-way latency, and jitter. Figure 6-20 shows an example of such a deployment between two branches.

Figure 6-20 Example of IPSLA Used for Ongoing Performance Monitoring



For example, Figure 6-20 shows both TelePresence and desktop video conferencing endpoints. Following the Cisco 12-class QoS model, TelePresence traffic can be marked CS4 and placed within a real-time interactive service class because it traverses both the private WAN links as well as an MPLS service between the branches. Likewise, desktop video conferencing traffic can be marked AF41 and placed within a Multimedia Conferencing service class because it traverses both the private WAN links and MPLS service between the branches. (Note that both traffic types may be remarked as it enters and exits the MPLS network). The configuration snippets in Example 6-6 and Example 6-7 show this type of IPSLA configuration with a pair of Cisco 3845 ISRs, one configured as the IPSLA sender and the other configured as the corresponding IPSLA responder.

Example 6-6 IPSLA Sender Configuration on a Cisco ISR 3845

```
ip sla 10
 udp-jitter 10.31.0.1 32800 source-ip 10.17.255.37 source-port 32800 num-packets 5
interval 200
 request-data-size 958
 tos 128
 frequency 300
!
ip sla 11
 udp-jitter 10.31.0.1 32802 source-ip 10.17.255.37 source-port 32802 num-packets 5
interval 200
 request-data-size 958
 tos 136
 frequency 300
!
ip sla reaction-configuration 10 react jitterDSAvg threshold-value 10 1 threshold-type
immediate action-type trapOnly
ip sla reaction-configuration 10 react rtt threshold-value 300 1 threshold-type immediate
action-type trapOnly
ip sla reaction-configuration 10 react jitterSDAvg threshold-value 10 1 threshold-type
immediate action-type trapOnly
ip sla reaction-configuration 10 react packetLossDS threshold-value 1 1 threshold-type
immediate action-type trapOnly
ip sla reaction-configuration 10 react packetLossSD threshold-value 1 1 threshold-type
immediate action-type trapOnly
ip sla reaction-configuration 10 react connectionLoss threshold-type immediate action-type
trapOnly
ip sla reaction-configuration 10 react timeout threshold-type immediate action-type
trapOnly
!
ip sla reaction-configuration 11 react rtt threshold-value 300 1 threshold-type immediate
action-type trapOnly
ip sla reaction-configuration 11 react jitterDSAvg threshold-value 10 1 threshold-type
immediate action-type trapOnly
ip sla reaction-configuration 11 react jitterSDAvg threshold-value 10 1 threshold-type
immediate action-type trapOnly
ip sla reaction-configuration 11 react packetLossDS threshold-value 1 1 threshold-type
immediate action-type trapOnly
ip sla reaction-configuration 11 react packetLossSD threshold-value 1 1 threshold-type
immediate action-type trapOnly
ip sla reaction-configuration 11 react connectionLoss threshold-type immediate action-type
trapOnly
ip sla reaction-configuration 11 react timeout threshold-type immediate action-type
trapOnly
!
ip sla group schedule 1 10-11 schedule-period 5 frequency 300 start-time now life forever
!
~
```

Example 6-7 IPSLA Responder Configuration on a Cisco ISR 3845

```
ip sla monitor responder
ip sla monitor responder type udpEcho ipaddress 10.31.0.1 port 32800
ip sla monitor responder type udpEcho ipaddress 10.31.0.1 port 32802
!
~
```

In the configuration example above, five 1000-byte packets (probes) with a CS4 DSCP marking, each spaced 200 milliseconds apart, are sent every 300 seconds. Likewise, five 1000-byte packets with an AF41 DSCP marking are sent every 300 seconds.

Note The request-data-size parameter within the UDP jitter IPSLA operation specifies only the UDP payload size. The overall packet size on an Ethernet network can be obtained by adding the IP header (20 bytes), UDP header (8 bytes), and Layer 2 Ethernet header (14 bytes).

This is a relatively small amount of traffic that can be used to measure parameters such as jitter, one-way latency, and packet loss per service class on an ongoing basis. As with the Cisco Catalyst 6500 example above, statistics can be viewed via the **show ip sla monitor statistics aggregated details** command on the Cisco 3845 ISR configured as the IPSLA sender. However, in this example, the IPSLA sender has also been configured to send SNMP traps in response to the IPSLA traffic in the following situations:

- When destination-to-source jitter or source-to-destination jitter is outside the range of 1–10 milliseconds
- When the round trip latency is outside the range of 1–300 milliseconds
- When any packet loss occurs
- When the IPSLA operation times out or the IPSLA control session indicates a connection loss

Note that the jitter, packet loss, and round-trip-time latency parameters for the SNMP traps are configurable. The values used here are examples only. The settings chosen on a real implementation depend entirely on the service level targets for the particular traffic service class. For a discussion of each of the various traps and how to configure them on Cisco IOS router platforms, see *Cisco IOS IPSLAs Configuration Guide, Release 12.4* at the following URL: http://www.cisco.com/en/US/docs/ios/12_4/ip_sla/configuration/guide/hsla_c.html.

Rather than having to periodically log on to the IPSLA sender to view the statistics, you can simply monitor a central SNMP trap collector to determine whether the jitter, packet loss, and latency targets are being met. The usefulness of this approach depends to a large extend on how often the IPSLA traffic is sent and what the network is experiencing in terms of congestion. If a network is experiencing somewhat continuous congestion, resulting in high jitter (because of queueing) and some packet loss, an IPSLA operation that sends a few packets every few minutes is likely to experience some degradation, and therefore generate an SNMP trap to alert the network administrator. However, even under these circumstances, it may be several IPSLA cycles before one of the IPSLA packets is dropped or experiences high jitter. If the network experiences very transient congestion, resulting in brief moments of high jitter and packet loss, possibly periodic in nature (because of some traffic that sends periodic bursts of packets, such as high definition video frames), it may be many cycles before any of the IPSLA packets experience any packet loss or high jitter. Therefore, you must again balance the amount and frequency of traffic sent via the IPSLA operations against the additional overhead and potential degradation of network performance caused by the IPSLA operation itself. However, if implemented carefully, IPSLA operations can be used to proactively monitor service-level parameters such as jitter, packet loss, and latency per service class, on an ongoing basis. As discussed earlier, you may also choose to implement dedicated routers for IPSLA probes used for ongoing performance monitoring, rather than using the existing routers at branch and campus locations.

Router and Switch Command-Line Interface

The following sections present several Cisco router and switch commands that can be run from the CLI, to gain visibility into traffic flows across an enterprise medianet. As with the functionality discussed in previous sections, a complete listing of all possible CLI commands is outside the scope of this document. Instead, this discussion focuses on commands that assist in determining at a high level whether drops are occurring within router and switch interfaces along the path of a video flow; and more specifically, to determine whether drops are occurring within service classes that are mapped to separate queues within the interfaces on the respective platforms. It is assumed that a QoS model has been implemented in which the various video applications are mapped to different service classes within the medianet infrastructure. The mapping of video applications can be accomplished through classification and marking of the application within the Cisco Catalyst switch port at the ingress edge of the network infrastructure; or by trusting an application-specific device, which is then connected to the Cisco Catalyst switch port, to correctly mark its traffic. Figure 6-21 shows an example of such a QoS model. The reader is encouraged to review Chapter 4, "Medianet QoS Design Considerations" before proceeding.

Figure 6-21 Cisco RFC-4594 Based 12-Class QoS Model

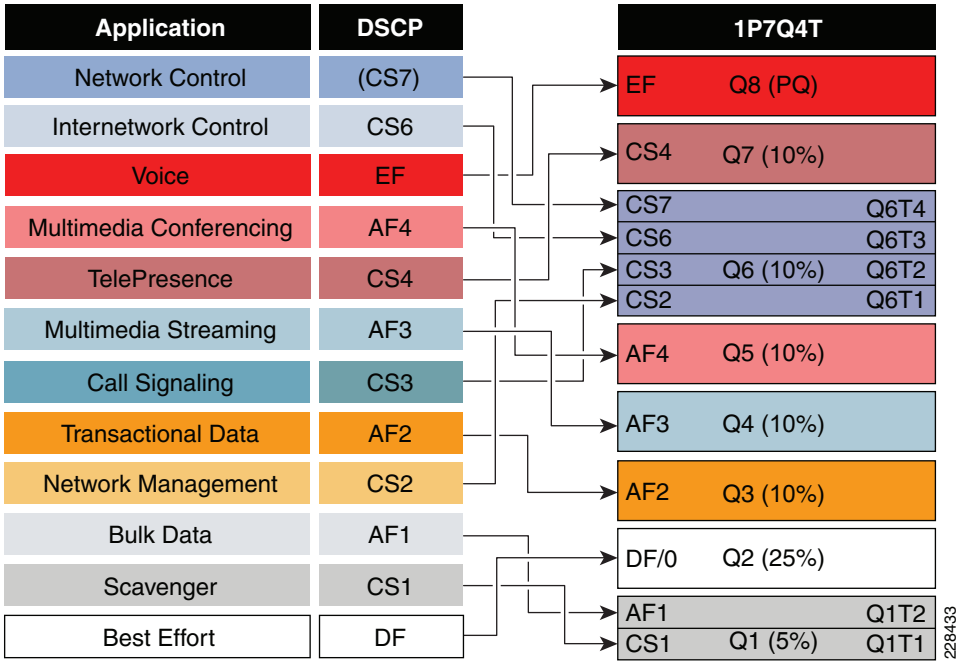
Application Class	PHB	Admission Control	Queueing and Dropping	Application Examples
VoIP Telephony	EF	Required	Priority Queue (PQ)	Cisco IP Phones
Broadcast Video	CS5	Required	Optional (PQ)	Cisco IP Surveillance, Cisco Enterprise TV
Realtime Interactive	CS4	Required	Optional (PQ)	Cisco TelePresence
Multimedia Conferencing	AF4	Required	BW Queue + DSCP WRED	Cisco Unified Personal Communicator
Multimedia Streaming	AF3	Recommended	BW Queue + DSCP WRED	Cisco Digital Media System (VoD)
Network Control	CS6		BW Queue	EIGRP, OSPF, BGP, HSRP, IKE
Call-Signaling	CS3		BW Queue	SCCP, SIP, H.323
OAM	CS2		BW Queue	SNMP, SSH, Syslog
Transactional Data	AF2		BW Queue + DSCP WRED	Cisco WebEx, Cisco MeetingPlace, ERP Apps
Bulk Data	AF1		BW Queue + DSCP WRED	Email, FTP, Backup Apps, Content Distribution
Best Effort	default		Default Queue + RED	Default Class Traffic
Scavenger	CS1		Min BW Queue (Deferential)	YouTube, iTunes, BitTorrent, Xbox Live

227922

As can be seen from Figure 6-21, IP video surveillance traffic is assigned to the Broadcast Video service class with a CS5 marking; TelePresence traffic is assigned to the Real-Time Interactive service class with a CS4 marking; desktop videoconferencing is assigned to the Multimedia Conferencing service class with an AF4x marking; and VoD/enterprise TV is assigned to the Multimedia Streaming service class with an AF3x marking. After the traffic from the various video applications has been classified and marked, it can then be mapped to specific ingress and egress queues and drop thresholds on Cisco router and switch platforms. Each queue can then be allocated a specific percentage of the overall bandwidth of the interface as well as a percentage of the overall buffer space of the particular interface. This provides a level of protection where one particular video and/or data application mapped to a particular service class cannot use all the available bandwidth, resulting in the degradation of all other video and/or data applications mapped to other service classes. The more granular the mapping of the service classes to separate queues (in other words, the more queues implemented on a platform), the more granular the

control and therefore the protection of service classes. When multiple service classes are mapped to a single queue, separate drop thresholds can be implemented (on platforms that support them) to provide differentiation of service classes within the queue. The implementation of queueing and drop thresholds is viewed as necessary to provide the correct per-hop treatment of the video application traffic to meet the overall desired service levels of latency, jitter, and packet loss across the medianet infrastructure. An example of the mapping of service classes to egress queueing on a Cisco Catalyst 6500 WS-X6704-10GE line card, which has a 1P7Q8T egress queueing structure, as shown in Figure 6-22. Note that the percentage of bandwidth allocated per queue depends on the customer environment; Figure 6-22 shows only an example.

Figure 6-22 Example Mapping of Service Classes to Egress Queueing on a Cisco Catalyst 6500 Line Card with 1P7Q8T Structure



This QoS methodology can also provide enhanced visibility into the amount of traffic from individual video application types crossing points within the medianet infrastructure. The more granular the mapping of individual video applications to service classes that are then mapped to ingress and egress queues, the more granular the visibility into the amount of traffic generated by particular video applications. You can also gain additional visibility into troubleshooting video quality issues caused by drops within individual queues on router and switch platforms.

The following high-level methodology can be useful for troubleshooting video performance issues using the router and switch CLI. As with anything, this methodology is not perfect. Some of the shortcomings of the methodology are discussed in the sections that cover the individual CLI commands. However, it can often be used to quickly identify the point within the network infrastructure where video quality issues are occurring. The steps are as follows:

1. Determine the Layer 3 hop-by-hop path of the particular video application across the medianet infrastructure from end-to-end, starting from the Layer 3 device closest to one end of the video session. The traceroute CLI utility can be used for this function.

2. Determine at a high level whether any drops are being seen by interfaces on each of the Layer 3 devices along the path. The **show interface summary** command can be used to provide this function quickly. If drops are being seen on a Layer 3 device, further information can be gained by observing the specific interfaces in which drops are occurring. The **show interface** <*interface*> command can be used for this.
3. To determine whether drops are occurring within the specific queue to which the video application is mapped on the platform, various **show** commands that are specific to a particular platform can be used.

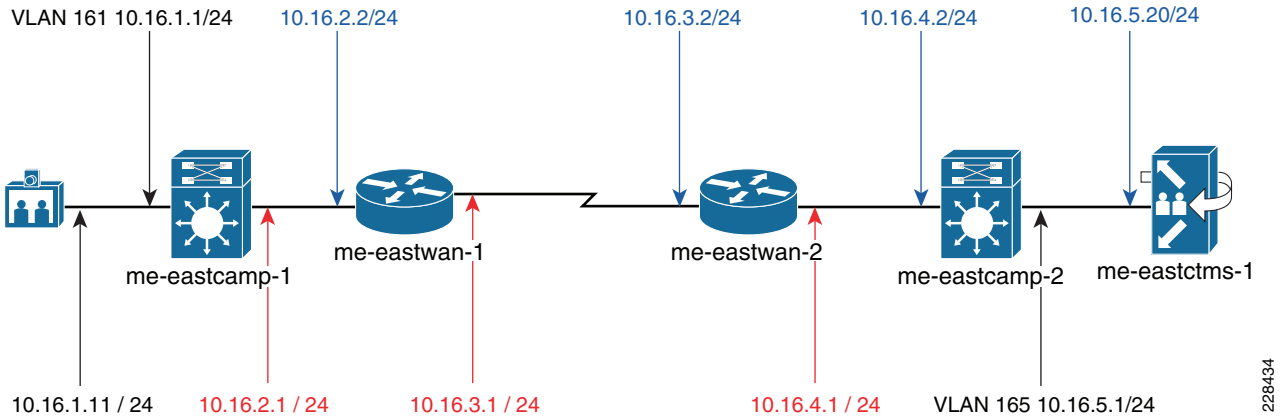
The following sections discuss the commands for each of the steps.

Traceroute

Traceroute is a command-line utility within Cisco router and switch products (and also in Unix and Linux systems) that can be used to produce a list of Layer 3 devices between two points within an IP network infrastructure. The Cisco traceroute utility sends a series of UDP packets with incrementing time-to-live (TTL) values from one IP address configured on the Layer 3 router or switch, to the desired destination IP address. Each Layer 3 device along the path either decrements the TTL value and forwards the UDP packet to the next hop in the path; or, if the TTL value is down to 1, the Layer 3 device discards the packet and sends an ICMP Time Exceeded (Type 11) message back to the source IP address. The ICMP Time Exceeded messages received by the source IP address (in other words, the originating router or switch device) are used to create a list of Layer 3 hops between the source and destination addresses. Note that ICMP Time Exceeded messages need to be allowed within the medianet infrastructure for traceroute to work. Also, if a Layer 3 device along the path does not send ICMP Time Exceeded messages, that device is not included in the list of Layer 3 hops between the source and destination addresses. Further, any Layer 2 devices along the path, which may themselves be the cause of the video quality degradation, are not identified in the traceroute output, because traceroute uses the underlying Layer 3 IP routing infrastructure to operate.

The traceroute utility works best when there are no equal-cost routes between network devices within the IP network infrastructure, and the infrastructure consists of all Layer 3 switches and routers, as shown in the sample network in Figure 6-23.

Figure 6-23 Traceroute in an IP Network with a Single Route Between Endpoints



In this network, if the **traceroute** command is run from *me-eastcamp-1* using the VLAN 161 interface as the source interface, the output looks similar to that shown in Example 6-8.

Example 6-8 Example Output from the Traceroute Utility Over a Non-Redundant Path Network

```
me-eastcamp-1#traceroute
Protocol [ip]:
Target IP address: 10.16.5.20
Source address: 10.16.1.1
Numeric display [n]: yes
Timeout in seconds [3]:
Probe count [3]: 4            ! Sets the number of packets generated with each TTL value.
Minimum Time to Live [1]:
Maximum Time to Live [30]: 6      ! Sets the max TTL value of the UDP packets generated.
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]: v
Loose, Strict, Record, Timestamp, Verbose[V]:
Type escape sequence to abort.
Tracing the route to 10.16.5.20

  1 10.16.2.2 0 msec 4 msec 0 msec 0 msec
  2 10.16.3.2 0 msec 0 msec 4 msec 0 msec
  3 10.16.4.2 0 msec 0 msec 4 msec 4 msec
  4 10.16.5.20 0 msec 0 msec 0 msec 8 msec
```

Traceroute returns the IP addresses of each Layer 3 hop in the route between *me-eastcamp-1* and *me-eastctms-1*. More specifically, because traceroute traces the hop route in one direction only, it returns the IP address of the interface of each router or switch that is closest to the source IP address. These IP addresses are shown in blue in Figure 6-23. Note that because traceroute is initiated by the *me-eastcamp-1* switch, it does not appear within the traceroute output itself.

If the traceroute command is run from *me-eastcamp-2* using the VLAN 165 interface as the source interface, the output looks similar to that shown in Example 6-9.

Example 6-9 Example Output from the Traceroute Utility From the Other Direction

```
me-eastcamp-1#traceroute
Protocol [ip]: ip
Target IP address: 10.16.1.11
Source address: 10.16.5.1
Numeric display [n]: yes
Timeout in seconds [3]:
Probe count [3]: 4
Minimum Time to Live [1]:
Maximum Time to Live [30]: 6
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]: V
Loose, Strict, Record, Timestamp, Verbose[V]:
Type escape sequence to abort.
Tracing the route to 10.16.1.11

  1 10.16.4.1 0 msec 0 msec 0 msec 4 msec
  2 10.16.3.1 0 msec 0 msec 0 msec 0 msec
  3 10.16.2.1 0 msec 0 msec 4 msec 0 msec
  4  *  *  *  *
  5  *  *  *  *
  6  *  *  *  *
Destination not found inside max TTL diameter.   ! Indicates the end device did not return
                                                 ! an ICMP Time Exceeded Pkt.
```

The IP addresses returned from traceroute run in this direction are shown in red in Figure 6-23. Note that because traceroute is initiated by the *me-eastcamp-2* switch, it does not appear within the traceroute output itself. Because a single path exists between the CTS-1000 and the Cisco TelePresence Multipoint Switch, the same Layer 3 hops (routers and switches) are returned regardless of which direction the

traceroute is run, although different IP addresses corresponding to different interfaces are returned, and the list of devices is reversed. Therefore, you need to run the traceroute in only one direction to understand the media flows in both directions. However, note that this may not necessarily be the case in a network with multiple equal-cost paths. Also note that the CTS-1000 does not return ICMP Time Exceeded packets, and therefore the traceroute utility times out. For a TelePresence endpoint, this can be rectified by directing the traceroute to the IP Phone associated with the TelePresence endpoint. However, be aware that some video endpoints may not respond to UDP traceroute packets with ICMP Time Exceeded packets.

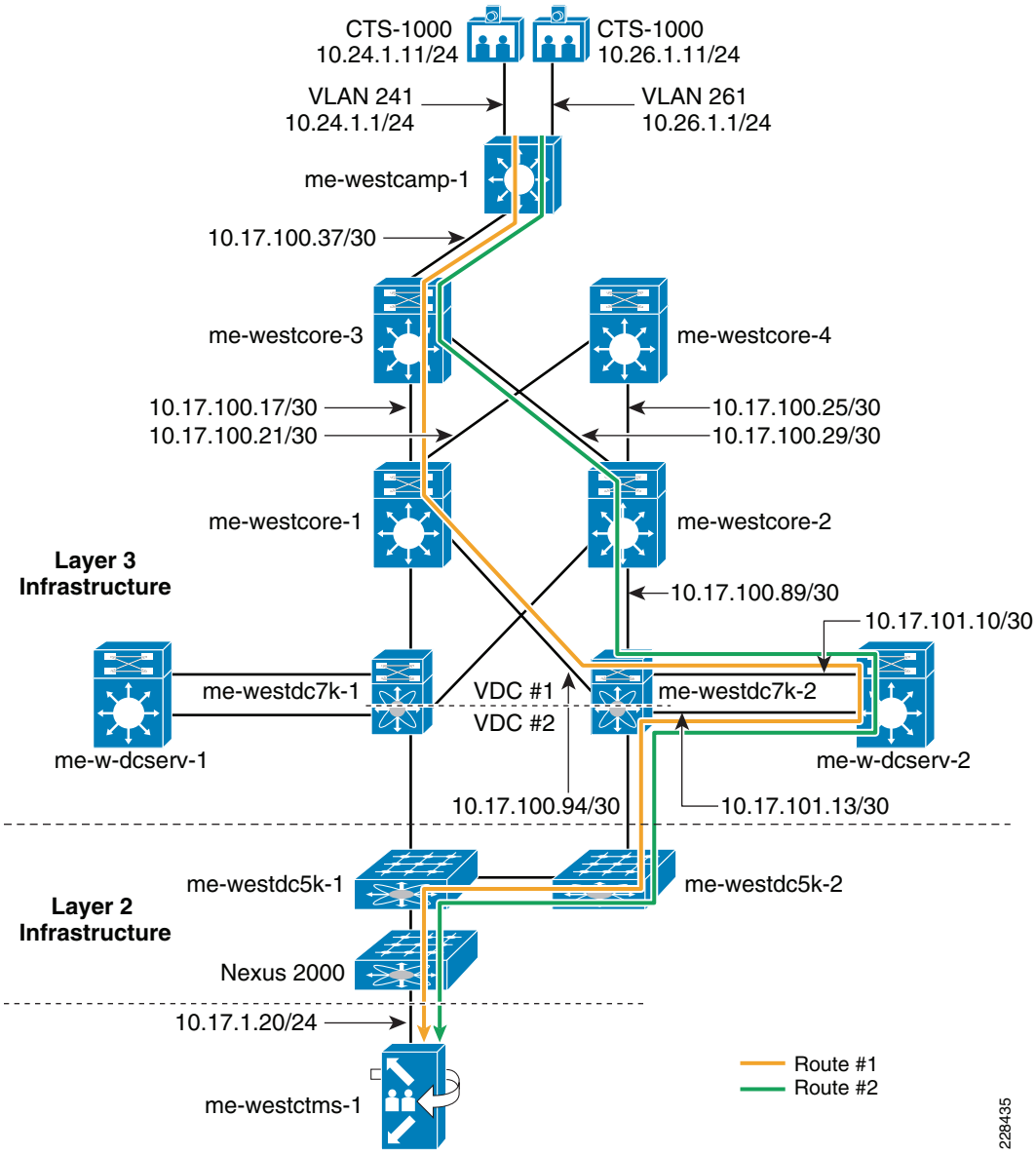
Single-path non-redundant IP network infrastructures are somewhat counter to best practices for network designs with high availability in mind. Unfortunately, the use of traceroute within an equal-cost redundant IP network infrastructure can sometimes return unclear results regarding the path of an actual video flow between two endpoints. An example of why this occurs can be seen with the output of two **traceroute** commands run on a Cisco Catalyst 4500 Series switch to a TelePresence Cisco TelePresence Multipoint Switch (IP address 10.17.1.20), as shown in Example 6-10.

Example 6-10 Example Output from the Traceroute Utility

```
me-westcamp-1#traceroute
Protocol [ip]:
Target IP address: 10.17.1.20
Source address: 10.24.1.1
Numeric display [n]: yes
Timeout in seconds [3]:
Probe count [3]: 4
Minimum Time to Live [1]:
Maximum Time to Live [30]: 10
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]: v
Loose, Strict, Record, Timestamp, Verbose[V]:
Type escape sequence to abort.
Tracing the route to 10.17.1.20

  1 10.17.100.37 8 msec 0 msec 4 msec 0 msec
  2 10.17.100.17 0 msec 0 msec 4 msec 0 msec
  3 10.17.100.94 0 msec 0 msec 0 msec 0 msec
  4 10.17.101.10 4 msec 0 msec 0 msec 0 msec
  5 10.17.101.13 0 msec 4 msec 0 msec 0 msec
  6 10.17.1.20 0 msec 4 msec 0 msec 0 msec

me-westcamp-1#traceroute
Protocol [ip]:
Target IP address: 10.17.1.20
Source address: 10.26.1.1
Numeric display [n]: yes
Timeout in seconds [3]:
Probe count [3]: 4
Minimum Time to Live [1]:
Maximum Time to Live [30]: 10
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]: v
Loose, Strict, Record, Timestamp, Verbose[V]:
Type escape sequence to abort.
Tracing the route to 10.17.1.20

  1 10.17.100.37 0 msec 0 msec 0 msec 0 msec
  2 10.17.100.29 4 msec 0 msec 0 msec 0 msec
  3 10.17.100.89 0 msec 4 msec 0 msec 0 msec
  4 10.17.101.10 0 msec 0 msec 0 msec 4 msec
  5 10.17.101.13 0 msec 0 msec 0 msec 4 msec
  6 10.17.1.20 0 msec 0 msec 0 msec 0 msec
```

The output from this traceroute was obtained from the network shown in Figure 6-24.

Figure 6-24 Test Network Used for Traceroute Example



As can be seen from Example 6-10 and Figure 6-24, the first traceroute is run using the source interface VLAN 241 on switch *me-westcamp-1*, which has IP address 10.24.1.1. The output is Route #1: from *me-westcamp-1* to *me-westdist-3* to *me-westcore-1* to *me-westdc7k-2* (VDC #1) to *me-westdcserv-2* back to *me-westdc7k-2* (VDC #2) and finally to *me-westctms-1*. The second traceroute is run using the source interface VLAN 261 on switch *me-westcamp-1*, which has IP address 10.26.1.11. The output is Route #2: from *me-westcamp-1* to *me-westdist-3* to *me-westcore-2* to *me-westdc7k-2* (VDC #1) to *me-westdcserv-2* back to *me-westdc7k-2* (VDC #2) and finally to *me-westctms-1*. Note that the devices greyed out in Figure 6-22 do not show up at all within the output of either **traceroute** command. These include any Layer 2 devices as well as some Layer 3 devices, and also the actual Cisco TelePresence System endpoints, because the traceroute is initiated from the switches. The two traceroutes follow different routes through the redundant network infrastructure. This is because Cisco Express Forwarding

switching, which itself is based on IP routing protocols, by default load balances sessions based on a hash of the source and destination IP address, when equal-cost paths exist. Therefore, different source and destination address pairs may yield different routes through an equal-cost redundant path network infrastructure. Cisco Express Forwarding switching can be configured for per-packet load balancing. However, this is not recommended because it can result in out-of-order packets for voice and video media. Therefore, you may not be able to tell from the traceroute utility alone whether the route returned through the network is the actual route taken by the video media, because the source IP address of the video endpoint is different than that used for the traceroute utility on the router or switch. Ideally, if traceroute can be run on the video endpoint itself, the actual route followed by the media through the network infrastructure can more easily be determined. However, most video endpoints such as Cisco TelePresence endpoints, Cisco IP video surveillance cameras, and Cisco digital media players (DMPs) do not currently support the traceroute utility.

On some switch platforms, such as Cisco Catalyst 6500 Series platforms, the **show ip cef exact-route <***source ip address***> <***destination ip address***>** command may be used to determine the actual route taken by the media flow of interest. An example of the output using the actual source IP address of a TelePresence CTS-1000, 10.24.1.11, and the destination IP address of the Cisco TelePresence Multipoint Switch, 10.17.1.20, is shown in Example 6-11.

Example 6-11 Example Output from show ip cef exact-route Command

```
me-westdist-3>show ip cef exact-route 10.24.1.11 10.17.1.20
10.24.1.11      -> 10.17.1.20     : GigabitEthernet1/1 (next hop 10.17.100.29)
```

As can be seen, the actual route take by the video and audio streams from the CTS-1000 follows Route #2 from *me-westdist-3* to *me-westcore-2* within this hop, and not Route #1 from *me-westdist-3* to *me-westcore-1*. The same command can be run on all the switches in the path that support the command to determine the actual route of the video flow in question.

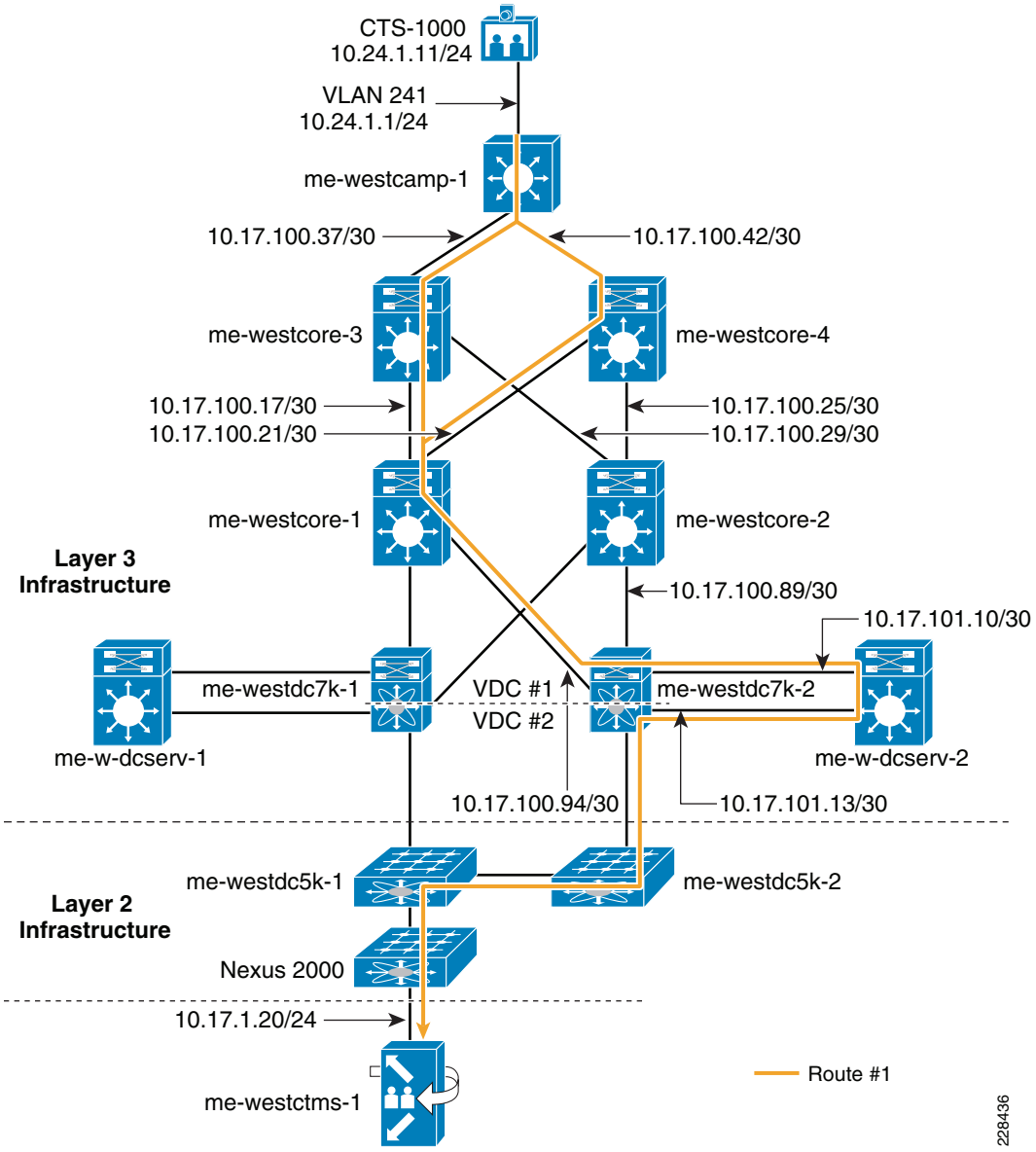
When the initial switch, from which the traceroute utility is run, has equal-cost paths to the first hop along the path to the destination, the output becomes somewhat undeterministic. This is because traceroute packets generated by the switch are CPU-generated, and therefore process-switched packets. These do not follow the Cisco Express Forwarding tables within the switch that generated them. Instead, the switch round-robins the multiple UDP packets generated, each with a given TTL value, out to each next hop with equal cost to the destination. The result is that only some of the hops corresponding to equal-cost paths appear in the traceroute output. However, the list of the actual hops returned by the traceroute depends on the Cisco Express Forwarding tables of the downstream switches and routers. An example of this behavior is shown in Example 6-12 and Figure 6-25.

Example 6-12 Example Output from Traceroute on a Switch with Redundant Paths

```
me-westcamp-1#traceroute
Protocol [ip]:
Target IP address: 10.17.1.20
Source address: 10.24.1.1
Numeric display [n]: yes
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 10.17.1.20

  1 10.17.100.37 0 msec
    10.17.100.42 0 msec
```

```
   10.17.100.37 0 msec
2  10.17.100.21 4 msec
   10.17.100.17 0 msec
   10.17.100.21 0 msec
3  10.17.100.94 0 msec 0 msec 0 msec
4  10.17.101.10 0 msec 0 msec 0 msec
5  10.17.101.13 0 msec 4 msec 0 msec
6  10.17.1.20 0 msec 0 msec 4 msec
```

Figure 6-25 Test Network Used for Redundant Switch Traceroute Example



The traceroute shown in Example 6-12 is again run from source interface VLAN 241 with source IP address 10.24.1.11 to the Cisco TelePresence Multipoint Switch with destination IP address 10.20.1.11. Because the switch from which the **traceroute** command is run has equal-cost paths to the first hop in the path, both switches *me-westdist-3* and *me-westdist-4* appear as the first hop in the path. Both paths then converge at the next switch hop, *me-westcore-1*, with *me-westcore-2* not showing up at all in the

traceroute output. However, note that a video traffic session (consisting of a source IP address and a destination IP address) that is Cisco Express Forwarding-switched through the router follows one or the other first hop through *me-westdist-3* or *me-westdist-4*, and not both hops, as indicated within the traceroute output. Again, the use of the **show ip cef exact-route** command on switches along the path may be necessary to determine the exact route of the video flows.

show interface summary and show interface Commands

After you have discovered the path of the actual video stream, possibly from using a combination of **traceroute** and the **show ip cef exact-route** command on switches along the path, a next logical step in troubleshooting a video quality issue is to see at a very high level whether interfaces are dropping packets. The **show interface summary** command can be used on Cisco Catalyst switch and IOS router platforms for this purpose (note that this command is not supported on Cisco Nexus switch platforms). Example 6-13 shows an example output from this command on a Cisco Catalyst 6500 platform.

Example 6-13 Partial Output from the show interface summary Command on a Cisco Catalyst 6500 Switch

```
me-westcore-1#show interface summary

 *: interface is up
 IHQ: pkts in input hold queue        IQD: pkts dropped from input queue
 OHQ: pkts in output hold queue    OQD: pkts dropped from output queue
 RXBS: rx rate (bits/sec)               RXPS: rx rate (pkts/sec)
 TXBS: tx rate (bits/sec)               TXPS: tx rate (pkts/sec)
 TRTL: throttle count

  Interface                IHQ IQD OHQ OQD RXBS   RXPS    TXBS    TXPS    TRTL
-------------------------------------------------------------------------------
  Vlan1                    0   0   0   0   0      0       0       0       0
* GigabitEthernet1/1       0   0   0   0   1000   1       0       0       0
  GigabitEthernet1/2       0   0   0   0   0      0       0       0       0
...

* TenGigabitEthernet3/1    0   0   0   0   1000   1       2000    1       0
* TenGigabitEthernet3/2    0   0   0   0   1000   1       1000    1       0
  TenGigabitEthernet3/3    0   0   0   0   0      0       0       0       0
  TenGigabitEthernet3/4    0   0   0   0   0      0       0       0       0
* GigabitEthernet5/1          00   0   0   1000   1       2000    3       0
* GigabitEthernet5/2          00   0   0   2000   2       0       0       0
* Loopback0                0   0   0   0   0      0       0       0       0
```

The **show interface summary** command can be used to quickly identify the following:

- Which interfaces are up on the switch or router, as indicated by the asterisk next to the interface
- Whether any interfaces are experiencing any input queue drops (IQD) or output queue drops (OQD)
- The amount of traffic transmitted by the interface in terms of bits/second (TXBS) or packets/second (TXPS)
- The amount of traffic received by the interface in terms of bits/second (RXBS) or packets/second (RXPS)

The **show interface summary** command may need to be run multiple times over a short time interval to determine whether drops are currently occurring, rather than having occurred previously. Alternatively, the **clear counters** command can typically be used to clear all the counters on all the interfaces.

However, simply because an interface is determined to be experiencing drops does not necessarily mean that the interface is relevant to the path of the video flow in question. You may still need to run the **show ip cef exact-route** command, or consult the IP routing tables via the **show ip route** command to determine whether the particular interface experiencing drops is along the path of the video flow. Example 6-14 shows an example output from both of these commands.

Example 6-14 Example Output from the show ip route and show ip cef exact-route Commands

```
me-westdist-3#show ip route 10.17.1.0
Routing entry for 10.17.1.0/24
  Known via "eigrp 111", distance 90, metric 6144, type internal
  Redistributing via eigrp 111
  Last update from 10.17.100.29 on GigabitEthernet1/1, 2w2d ago
  Routing Descriptor Blocks:
  * 10.17.100.17, from 10.17.100.17, 2w2d ago, via GigabitEthernet5/3
      Route metric is 6144, traffic share count is 1
      Total delay is 140 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 4
    10.17.100.29, from 10.17.100.29, 2w2d ago, via GigabitEthernet1/1
      Route metric is 6144, traffic share count is 1
      Total delay is 140 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 4

me-westdist-3#show ip cef exact-route 10.24.1.11 10.17.1.20
10.24.1.11      -> 10.17.1.20     : GigabitEthernet1/1 (next hop 10.17.100.29)
```

Example 6-14 shows that the IP routing tables indicate that there are equal-cost paths to IP subnet 10.17.1.20 through next hops 10.17.100.17 and 10.17.100.29, via interfaces GigabitEthernet5/3 and GigabitEthernet1/1, respectively. The asterisk next to the 10.17.100.17 route indicates that the next session will follow that route. However, the output from the **show ip cef exact-route** command shows that the Cisco Express Forwarding table has already been populated with a session from source IP address 10.24.1.11, corresponding to the CTS-1000, to destination IP address 10.17.1.20, corresponding to the Cisco TelePresence Multipoint Switch, via interface GigabitEthernet1/1. Therefore, when troubleshooting drops along the path for this particular video flow, you should be concerned with drops shown on interface GigabitEthernet1/1.

Having determined which relevant interfaces are currently experiencing drops, you can drill down further into the interface via the **show interface <***interface***>** command. Example 6-15 shows an example output from this command on a Cisco Catalyst 6500 platform.

Example 6-15 Example Output from the show interface Command

```
me-westdist-3#show interface gigabitethernet1/1
GigabitEthernet1/1 is up, line protocol is up (connected)
  Hardware is C6k 1000Mb 802.3, address is 0018.74e2.7dc0 (bia 0018.74e2.7dc0)
  Description: CONNECTION TO ME-WESTCORE-2 GIG1/25
  Internet address is 10.17.100.30/30
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 1000Mb/s
  input flow-control is off, output flow-control is off
  Clock mode is auto
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:04, output 00:00:00, output hang never
  Last clearing of "show interface" counters 00:13:53
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0! Input & output
```

```
                                                                    ! queue drops.
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
  L2 Switched: ucast: 117 pkt, 22493 bytes - mcast: 184 pkt, 14316 bytes
  L3 in Switched: ucast: 14 pkt, 7159 bytes - mcast: 0 pkt, 0 bytes mcast
  L3 out Switched: ucast: 0 pkt, 0 bytes mcast: 0 pkt, 0 bytes
     374 packets input, 53264 bytes, 0 no buffer
     Received 250 broadcasts (183 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored ! May indicate link-level errors
    0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     282 packets output, 32205 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets  ! May indicate link-level errors.
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

The **show interface** command can provide an instantaneous display of the current depth of the input and output queues, as well as a running total of input and output drops seen by the interface. This can be used to detect possible congestion issues occurring within the switch interface. It also provides additional detail in terms of the type of traffic: unicast versus multicast switched by the interface. More importantly, the **show interface** command provides additional detail regarding potential link level errors, such as CRCs, collisions, and so on. These can be the result of cabling issues or even duplex mismatches between switch interfaces that are difficult to detect, but can be the cause of degraded video quality as well. Note that changing the load interval from the default of 5 minutes to a lower value, such as 60 seconds, can provide increased visibility, so that the statistics are then more up-to-date.

Platform Specific Queue-Level Commands

Because a relevant interface along the path of the video flow in question is experiencing drops does not necessarily mean that the drops are occurring within the queue that holds the particular video application traffic. You may need to run additional platform-specific commands to display drops down to the queue level to determine whether video degradation is occurring on a particular switch or router. The following sections discuss some of these platform-specific commands.

Cisco Catalyst 6500 Series Commands

When QoS is enabled on Cisco Catalyst 6500 Series switches, the **show queueing interface** command allows you to view interface drops per queue on the switch port. Example 6-16 shows the output from a Cisco Catalyst 6500 WS-X6708-10GE line card. Selected areas for discussion have been highlighted in bold.

Example 6-16 Output from Cisco Catalyst 6500 show queueing interface Command

```
me-eastcore-1#show queueing interface tenGigabitEthernet 1/1

Interface TenGigabitEthernet1/1 queueing strategy:  Weighted Round-Robin
  Port QoS is enabled
  Trust boundary disabled

  Trust state: trust DSCP
  Extend trust state: not trusted [COS = 0]
  Default COS is 0
    Queueing Mode In Tx direction: mode-dscp    Transmit queues [type = 1p7q4t]:
    Queue Id    Scheduling  Num of thresholds
```

```
    ----------------------------------------
       01         WRR                 04
       02         WRR                 04
       03         WRR                 04
       04         WRR                 04
       05         WRR                 04
       06         WRR                 04
       07         WRR                 04
       08         Priority              01

    WRR bandwidth ratios:    1[queue 1]  25[queue 2]   4[queue 3]  10[queue 4]  10[queue
5]  10[queue 6]  10[queue 7]
    queue-limit ratios:           1[queue 1]  25[queue 2]   4[queue 3]  10[queue 4]
10[queue 5]  10[queue 6]  10[queue 7]  30[Pri Queue]

    queue tail-drop-thresholds
    --------------------------
    1     70[1] 100[2] 100[3] 100[4]
    2     70[1] 100[2] 100[3] 100[4]
    3     100[1] 100[2] 100[3] 100[4]
    4     100[1] 100[2] 100[3] 100[4]
    5     100[1] 100[2] 100[3] 100[4]
    6     100[1] 100[2] 100[3] 100[4]
    7     100[1] 100[2] 100[3] 100[4]

    queue random-detect-min-thresholds
    ----------------------------------
      1    80[1] 100[2] 100[3] 100[4]
      2    80[1] 100[2] 100[3] 100[4]
      3    70[1] 80[2] 90[3] 100[4]
      4    70[1] 80[2] 90[3] 100[4]
      5    70[1] 80[2] 90[3] 100[4]
      6    70[1] 80[2] 90[3] 100[4]
      7    60[1] 70[2] 80[3] 90[4]

    queue random-detect-max-thresholds
    ----------------------------------
      1    100[1] 100[2] 100[3] 100[4]
      2    100[1] 100[2] 100[3] 100[4]
      3    80[1] 90[2] 100[3] 100[4]
      4    80[1] 90[2] 100[3] 100[4]
      5    80[1] 90[2] 100[3] 100[4]
      6    80[1] 90[2] 100[3] 100[4]
      7    70[1] 80[2] 90[3] 100[4]

    WRED disabled queues:

    queue thresh cos-map
    ---------------------------------------
    1     1      0
    1     2      1
    1     3
    1     4
    2     1      2
    2     2      3 4
    2     3
    2     4
    3     1      6 7
    3     2
    3     3
    3     4
    4     1
    4     2
    4     3
```

```
    4     4
    5     1
    5     2
    5     3
    5     4
    6     1
    6     2
    6     3
    6     4
    7     1
    7     2
    7     3
    7     4
    8     1       5

    queue thresh dscp-map
    ---------------------------------------
    1     1       1 2 3 4 5 6 7 8 9 11 13 15 17 19 21 23 25 27 29 31 33 39 41 42 43 44 45
47
    1     2
    1     3
    1     4
    2     1       0
    2     2
    2     3
    2     4
    3     1       14
    3     2       12
    3     3       10
    3     4
    4     1       22
    4     2       20
    4     3       18
    4     4
    5     1       30 35 37
    5     2       28
    5     3       26
    5     4
    6     1       38 49 50 51 52 53 54 55 57 58 59 60 61 62 63
    6     2       36
    6     3       34
    6     4
    7     1       16
    7     2       24
    7     3       48
    7     4       56
    8     1       32 40 46

    Queueing Mode In Rx direction: mode-dscp
    Receive queues [type = 8q4t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       01         WRR                 04
       02         WRR                 04
       03         WRR                 04
       04         WRR                 04
       05         WRR                 04
       06         WRR                 04
       07         WRR                 04
       08         WRR                 04

    WRR bandwidth ratios:   10[queue 1]   0[queue 2]   0[queue 3]   0[queue 4]   0[queue
5]   0[queue 6]   0[queue 7]  90[queue 8]
```

```
    queue-limit ratios:     80[queue 1]   0[queue 2]   0[queue 3]   0[queue 4]   0[queue
5]   0[queue 6]   0[queue 7]  20[queue 8]

    queue tail-drop-thresholds
    --------------------------
    1     70[1] 80[2] 90[3] 100[4]
    2     100[1] 100[2] 100[3] 100[4]
    3     100[1] 100[2] 100[3] 100[4]
    4     100[1] 100[2] 100[3] 100[4]
    5     100[1] 100[2] 100[3] 100[4]
    6     100[1] 100[2] 100[3] 100[4]
    7     100[1] 100[2] 100[3] 100[4]
    8     100[1] 100[2] 100[3] 100[4]

    queue random-detect-min-thresholds
    ----------------------------------
      1    40[1] 40[2] 50[3] 50[4]
      2    100[1] 100[2] 100[3] 100[4]
      3    100[1] 100[2] 100[3] 100[4]
      4    100[1] 100[2] 100[3] 100[4]
      5    100[1] 100[2] 100[3] 100[4]
      6    100[1] 100[2] 100[3] 100[4]
      7    100[1] 100[2] 100[3] 100[4]
      8    100[1] 100[2] 100[3] 100[4]

    queue random-detect-max-thresholds
    ----------------------------------
      1    70[1] 80[2] 90[3] 100[4]
      2    100[1] 100[2] 100[3] 100[4]
      3    100[1] 100[2] 100[3] 100[4]
      4    100[1] 100[2] 100[3] 100[4]
      5    100[1] 100[2] 100[3] 100[4]
      6    100[1] 100[2] 100[3] 100[4]
      7    100[1] 100[2] 100[3] 100[4]
      8    100[1] 100[2] 100[3] 100[4]

    WRED disabled queues:      2  3  4  5  6  7  8

    queue thresh cos-map
    ---------------------------------------
    1     1      0 1
    1     2      2 3
    1     3      4
    1     4      6 7
    2     1
    2     2
    2     3
    2     4
    3     1
    3     2
    3     3
    3     4
    4     1
    4     2
    4     3
    4     4
    5     1
    5     2
    5     3
    5     4
    6     1
    6     2
    6     3
    6     4
```

```
    7     1
    7     2
    7     3
    7     4
    8     1       5
    8     2
    8     3
    8     4

    queue thresh dscp-map
    ---------------------------------------
    1     1       0 1 2 3 4 5 6 7 8 9 11 13 15 16 17 19 21 23 25 27 29 31 33 39 41 42 43 44
45 47
    1     2
    1     3
    1     4
    2     1       14
    2     2       12
    2     3       10
    2     4
    3     1       22
    3     2       20
    3     3       18
    3     4
    4     1       24 30
    4     2       28
    4     3       26
    4     4
    5     1       32 34 35 36 37 38
    5     2
    5     3
    5     4
    6     1       48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
    6     2
    6     3
    6     4
    7     1
    7     2
    7     3
    7     4
    8     1       40 46
    8     2
    8     3
    8     4

  Packets dropped on Transmit:
    BPDU packets:  0

    queue               dropped  [dscp-map]
    ---------------------------------------------

    1                       0  [1 2 3 4 5 6 7 8 9 11 13 15 17 19 21 23 25 27 29 31 33 39
41 42 43 44 45 47 ]
    2                       0  [0 ]
    3                       0  [14 12 10 ]
    4                       0  [22 20 18 ]
    5                       0  [30 35 37 28 26 ]
    6                       0  [38 49 50 51 52 53 54 55 57 58 59 60 61 62 63 36 34 ]
    7                       0  [16 24 48 56 ]
    8                       0  [32 40 46 ]

  Packets dropped on Receive:
    BPDU packets:  0
```

```
    queue               dropped  [dscp-map]
    ---------------------------------------------
    1                         0  [0 1 2 3 4 5 6 7 8 9 11 13 15 16 17 19 21 23 25 27 29 31
33 39 41 42 43 44 45 47 ]
    2                         0  [14 12 10 ]
    3                         0  [22 20 18 ]
    4                         0  [24 30 28 26 ]
    5                         0  [32 34 35 36 37 38 ]
    6                         0  [48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 ]
    8                         0  [40 46 ]
```

The information within the first highlighted section can be used to quickly verify that the queueing and bandwidth ratios have been set correctly for the traffic service class of interest that is crossing the particular interface. As can be seen, the line card has a 1p7q4t egress queueing structure, meaning one priority queue and seven additional queues, each with four different drop thresholds. Egress queueing is configured to use a weighted-round robin (WRR) algorithm. The WRR bandwidth ratios are used by the scheduler to service the queues, which effectively allocates bandwidth across the seven non-priority queues based on the weight ratios. Note that the priority queue is always serviced first, and therefore has no weight. The queue-limit ratios allocate available egress queue space based on the ratios as well. Note that egress queueing space for the priority queue is included.

The second highlighted section can be used to quickly verify that a particular traffic service class is mapped to the correct egress queue on the line card. It provides a quick view of the mapping of DSCP values to egress queues and drop thresholds. Further, this can then be used to identify which video applications are mapped to which queues, based on DSCP values. This assumes specific video applications have been mapped to service classes with separate DSCP values. Note that in older Cisco Catalyst 6500 line cards, egress queues may be mapped to internal switch class of service (CoS) values that are then mapped to DSCP values. In such cases, you may need to use the **show mls qos maps dscp-cos** command to display the mapping of DSCP values to internal CoS values within the Cisco Catalyst switch.

Finally, the third highlighted block shows the number of packets dropped by the interface, per transmit queue. This can be used for either performance management, in the case where a particular video application mapped to the queue is experiencing degraded service because of packet loss; or for fault isolation, in the case where a particular video application is dropping the connection because of packet loss.

The same information is also provided for ingress queueing with this particular line card. Note, however, that the various Cisco Catalyst 6500 line cards support different ingress and egress queueing structures, as well as modes of operations. Older Cisco Catalyst 6500 line cards support ingress queuing based on Layer 2 CoS marking only. Ingress queueing may not be used within a routed (non-trunked) infrastructure on Cisco Catalyst 6500 line cards.

Cisco Catalyst 4500/4900 Series Commands

Visibility into traffic flows down at the queue level within a Cisco Catalyst 4500 Series switch depends on the supervisor line card within the switch. For Cisco Catalyst 4500 Series switches with a Supervisor-II-Plus, Supervisor-IV, or Supervisor-V (also referred to as classic supervisors), and for Cisco Catalyst 4900 Series switches, the **show interface counters** command provides a similar ability to view interface drops per queue on the switch port. Example 6-17 shows a partial output from a Cisco Catalyst 4948 switch. For brevity, output from only the first two interfaces and the last interface on the switch are shown. Selected areas for discussion have been highlighted in bold.

Example 6-17 Output from Cisco Catalyst 4948 show interface counters detail Command

```
tp-c2-4948-1#show interface counters detail
```

```
Port InBytesIn UcastPkts InMcastPkts InBcastPkts ! Provides info on ingress multicast packets
Gi1/1   0          0          0       0
Gi1/2   0          0          0       0
...
Gi1/48 500745084 946163       4778144 892284

Port OutBytes OutUcastPkts OutMcastPkts OutBcastPkts ! Provides info on egress multicast
                                                       packets
Gi1/1   0       0          0          0
Gi1/2   0       0          0          0
...
Gi1/48  18267775 20009     190696     2

Port    InPkts 64 OutPkts 64InPkts 65-127 OutPkts 65-127
Gi1/1   0          0          0          0
Gi1/2   0          0          0          0
...
Gi1/48  5676114    107817     705522     97227

Port    InPkts 128-255   OutPkts 128-255   InPkts 256-511   OutPkts 256-511
Gi1/1      0              0                  0          0
Gi1/2      0              0                  0          0
...
Gi1/48     58703          1700               169614     2283

Port        InPkts 512-1023  OutPkts 512-1023
Gi1/1      0                 0
Gi1/2      0                 0
...
Gi1/48     5859           1461

Port InPkts 1024-1518 OutPkts 1024-1518InPkts 1519-1548 OutPkts 1519-1548
Gi1/1   0              0                  0              0
Gi1/2   0              0                  0              0
...
Gi1/48 779             219                0              0

Port InPkts 1549-9216OutPkts 1549-9216
Gi1/1   0                 0
Gi1/2   0                 0
...
Gi1/48  0                 0

Port Tx-Bytes-Queue-1Tx-Bytes-Queue-2Tx-Bytes-Queue-3Tx-Bytes-Queue-4 ! Provides
                                                     ! transmitted byte count per queue
Gi1/1   0              0              0                 0
Gi1/2   0              0              0                 0
...
Gi1/48 67644           1749266        181312            16271855

Port    Tx-Drops-Queue-1Tx-Drops-Queue-2Tx-Drops-Queue-3 Tx-Drops-Queue-4 ! Provides
                                                          ! packet drop count per queue
Gi1/1   0                 0              0                 0
Gi1/2   0                 0              0                 0
...
Gi1/48  0                 0              0                 0

Port Dbl-Drops-Queue-1Dbl-Drops-Queue-2Dbl-Drops-Queue-3Dbl-Drops-Queue-4 ! Provides DBL
                                                          ! packet drop count per queue
Gi1/1   0                 0              0                 0
Gi1/2   0                 0              0                 0
...
Gi1/48  0                 0              0                 0
```

```
Port          Rx-No-Pkt-Buff     RxPauseFrames     TxPauseFrames    PauseFramesDrop
Gi1/1       0                  0                 0                0
Gi1/2       0                  0                 0                0
...
Gi1/48      0                  0                 0                0

Port        UnsupOpcodePause
Gi1/1       0
Gi1/2       0
...
Gi1/48      0
```

The first two highlighted sections can provide information regarding how many unicast and multicast packets have crossed the interface in the inbound or outbound direction. Multicast traffic is often used to support real-time and VoD broadcasts. The multicast packet count within the switch interface increments from when the switch was reloaded or the counters were manually cleared. Because of this, and because the information does not include the byte count, you cannot use the statistics alone to determine the data rate of multicast traffic across the interface. However, you may be able to gain some useful information regarding the percentage of multicast traffic on the interface based on the ratio of the unicast to multicast packets seen.

The third highlighted section provides two additional pieces of information. First, it indicates the number of queues per interface. Because the output above is from a Cisco Catalyst 4948 switch, four transmit queues per interface are supported. Second, the output indicates the amount of traffic, in bytes, that has been transmitted per queue per interface. Because this is a summation of bytes since the counters were last cleared or the switch reloaded, you must run the command multiple times over a time interval to get a rough estimate of the byte rate over that time period. This can be used to gain an idea of the current data rate of a particular traffic service class across the switch interface.

The final two highlighted sections indicate the number of packets dropped in the egress direction, per transmit queue. You can use this information to assist in troubleshooting a video application performance issue or fault condition caused by packet loss. Note that *Dbl-Drops* are drops that are the result of the dynamic buffer limiting (DBL) algorithm, which attempts to fairly allocate buffer usage per flow through the Cisco Catalyst 4500 switch. You have the option of enabling or disabling DBL per service class on the switch.

To make use of information regarding transmit queues drops shown in Example 6-17, you must understand which traffic classes are assigned to which transmit queues. For Cisco Catalyst 4500 Series switches with classic supervisors as well as Cisco Catalyst 4900 Series switches, the **show qos maps** command can be used to display which DSCP values are mapped to which transmit queues on the switch, as shown in Example 6-18.

Example 6-18 Output from Cisco Catalyst 4948 show qos maps Command

```
tp-c2-4948-1#show qos maps
DSCP-TxQueue Mapping Table (dscp = d1d2) ! Provides mapping of DSCP value to transmit
                                         ! queue on the switch
d1 : d2  0  1  2  3  4  5  6  7  8  9
---------------------------------------
 0 :    02 01 01 01 01 01 01 01 01 01
 1 :    01 01 01 01 01 01 04 02 04 02
 2 :    04 02 04 02 04 02 04 02 04 02
 3 :    04 02 03 03 04 03 04 03 04 03
 4 :    03 03 03 03 03 03 03 03 04 04
 5 :    04 04 04 04 04 04 04 04 04 04
 6 :    04 04 04 04

Policed DSCP Mapping Table (dscp = d1d2)
d1 : d2  0  1  2  3  4  5  6  7  8  9
```

```
-----------------------------------
 0 :    00 01 02 03 04 05 06 07 08 09
 1 :    10 11 12 13 14 15 16 17 18 19
 2 :    20 21 22 23 24 25 26 27 28 29
 3 :    30 31 32 33 34 35 36 37 38 39
 4 :    40 41 42 43 44 45 46 47 48 49
 5 :    50 51 52 53 54 55 56 57 58 59
 6 :    60 61 62 63

DSCP-CoS Mapping Table (dscp = d1d2)
d1 : d2  0  1  2  3  4  5  6  7  8  9
-----------------------------------
 0 :    00 00 00 00 00 00 00 00 01 01
 1 :    01 01 01 01 01 01 02 02 02 02
 2 :    02 02 02 02 03 03 03 03 03 03
 3 :    03 03 04 04 04 04 04 04 04 04
 4 :    05 05 05 05 05 05 05 05 06 06
 5 :    06 06 06 06 06 06 07 07 07 07
 6 :    07 07 07 07

CoS-DSCP Mapping Table
   CoS:   0  1  2  3  4  5  6  7
--------------------------------
  DSCP:   0  8 16 24 32 46 48 56
```

The highlighted section in the example above shows the mapping of the DSCP values to transmit queues. The vertical column, marked *d1*, represents the first decimal number of the DSCP value, while the horizontal column, marked *d2*, represents the second decimal number of the DSCP value. For example, a d1 value of 3 and a d2 value of 2 yields a DSCP decimal value of 32, which corresponds to the CS4 service class. You still need to separately understand the mapping of specific video applications to service classes that are then marked with a particular DSCP value. However, combined with the knowledge of which traffic classes are mapped to which transmit queue, you can use this information to troubleshoot video application performance issues across the Cisco Catalyst 4500/Cisco Catalyst 4900 switch platform.

Note DSCP markings are represented by 6-bit values within the ToS byte of the IP packet. The DSCP values are the upper 6 bits of the ToS byte. Therefore, a DSCP decimal value of 32 represents a binary value of 100000, or the CS4 service class. The full ToS byte would have a value of 10000000 or a hexidecimal value of 0x80.

For the Cisco Catalyst 4500 with a Sup-6E supervisor line card, the mapping of traffic classes to egress queues is accomplished via an egress policy map applied to the interface. The policy map can be viewed through the **show policy-map interface** command. Example 6-19 shows the output from a GigabitEthernet interface. Selected areas for discussion have been highlighted in bold.

Example 6-19 Output from Cisco Catalyst 4500 Sup-6E show policy-map interface Command

```
me-westcamp-1#show policy-map int gig 3/3
 GigabitEthernet3/3

  Service-policy output: 1P7Q1T ! Name and direction of the policy map applied to the
                                ! interface.

    Class-map: PRIORITY-QUEUE (match-any)!  Packet counters increment across all
     22709 packets                       ! interfaces to which the policy map is applied.

      Match:  dscp ef (46)
        0 packets
```

```
  Match:  dscp cs5 (40)
    0 packets
  Match:  dscp cs4 (32)
    22709 packets

  police:                 ! Byte counters under 'police' line increment per interface.
      cir 300000000 bps, bc 12375000 bytes, be 12375000 bytes
    conformed Packet count - n/a, 10957239 bytes; actions:
      transmit
    exceeded Packet count - n/a, 0 bytes; actions:
      drop
    violated Packet count - n/a, 0 bytes; actions:
      drop
    conformed 2131000 bps, exceed 0 bps, violate 0 bps
  priority queue:         ! Byte counters and packet drops under 'priority queue' line
      Transmit: 9877576 Bytes, Queue Full Drops: 0 Packets ! increment per interface.

Class-map: CONTROL-MGMT-QUEUE (match-any)
  17 packets
  Match:  dscp cs7 (56)
    0 packets
  Match:  dscp cs6 (48)
    8 packets
  Match:  dscp cs3 (24)
    9 packets
  Match:  dscp cs2 (16)
    0 packets
  bandwidth: 10 (%) ! Byte counters and packet drops under 'bandwidth'  line
      Transmit: 1616 Bytes, Queue Full Drops: 0 Packets ! increment per interface.

Class-map: MULTIMEDIA-CONFERENCING-QUEUE (match-all)
  0 packets
  Match:  dscp af41 (34) af42 (36) af43 (38)
  bandwidth: 10 (%)
      Transmit: 0 Bytes, Queue Full Drops: 0 Packets

Class-map: MULTIMEDIA-STREAMING-QUEUE (match-all)
  0 packets
  Match:  dscp af31 (26) af32 (28) af33 (30)
  bandwidth: 10 (%)
      Transmit: 0 Bytes, Queue Full Drops: 0 Packets

Class-map: TRANSACTIONAL-DATA-QUEUE (match-all)
  0 packets
  Match:  dscp af21 (18) af22 (20) af23 (22)
  bandwidth: 10 (%)
      Transmit: 0 Bytes, Queue Full Drops: 0 Packets
  dbl
      Probabilistic Drops: 0 Packets
      Belligerent Flow Drops: 0 Packets

Class-map: BULK-DATA-QUEUE (match-all)
  0 packets
  Match:  dscp af11 (10) af12 (12) af13 (14)
  bandwidth: 4 (%)
      Transmit: 0 Bytes, Queue Full Drops: 0 Packets
  dbl
      Probabilistic Drops: 0 Packets
      Belligerent Flow Drops: 0 Packets

Class-map: SCAVENGER-QUEUE (match-all)
  0 packets
  Match:  dscp cs1 (8)
  bandwidth: 1 (%)
```

```
        Transmit: 0 Bytes, Queue Full Drops: 0 Packets

    Class-map: class-default (match-any)
      6 packets
      Match: any
        6 packets
      bandwidth: 25 (%)
          Transmit: 436 Bytes, Queue Full Drops: 0 Packets
      dbl
          Probabilistic Drops: 0 Packets
          Belligerent Flow Drops: 0 Packets
```

In Example 6-19, the first highlighted line shows the name of the service policy and direction (outbound or inbound) applied to the interface. The second highlighted section shows the mapping of DSCP markings to each queue defined within the policy map. Directly under that, the number of packets that matched the service class are displayed. Take special note that if a policy map is shared among multiple interfaces, these packet counters increment for all interfaces that have traffic that matches the particular class-map entry. For example, if the policy map named *1P7Q1T* shown in the example above were applied across two uplink interfaces, the packet counters would show the total packets that matched each class-map entry for both interfaces. This can lead to some confusion, as shown in Example 6-20. Selected areas for discussion have been highlighted in bold.

Example 6-20 Second Example Output from Cisco Catalyst 4500 Sup-6E show policy-map interface Command

```
me-westcamp-1#show policy-map int gig 3/1
 GigabitEthernet3/1

  Service-policy output: 1P7Q1T

    Class-map: PRIORITY-QUEUE (match-any)
      15360 packets
      Match:  dscp ef (46)
        0 packets
      Match:  dscp cs5 (40)
        0 packets
      Match:  dscp cs4 (32)
        15360 packets
      police:
          cir 300000000 bps, bc 12375000 bytes, be 12375000 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        violated 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps, violate 0 bps
      priority queue:
          Transmit: 0 Bytes, Queue Full Drops: 0 Packets
```

Notice in Example 6-20 that interface GigabitEthernet3/1 appears to have seen 15,360 packets that match the PRIORITY-QUEUE class-map entry. Yet, both the policer and the priority queue statistics indicate that no packets that match the PRIORITY-QUEUE class-map entry have been sent by this interface. In this scenario, the 15,360 packets were sent by the other interface, GigabitEthernet3/3, which shared the policy map named *1P7Q1T*. To prevent this type of confusion when viewing statistics from the **show policy-map interface** command on the Cisco Catalyst 4500 with Sup6E, you can simply define a different policy map name for each interface. Example 6-21 shows an example of this type of configuration.

Example 6-21 Partial Configuration Example Showing Separate Policy Map Per Interface

```
class-map match-all MULTIMEDIA-STREAMING-QUEUE
 match  dscp af31  af32  af33
class-map match-any CONTROL-MGMT-QUEUE
 match  dscp cs7
 match  dscp cs6
 match  dscp cs3
 match  dscp cs2
class-map match-all TRANSACTIONAL-DATA-QUEUE
 match  dscp af21  af22  af23
class-map match-all SCAVENGER-QUEUE
 match  dscp cs1
class-map match-all MULTIMEDIA-CONFERENCING-QUEUE
 match  dscp af41  af42  af43
class-map match-all BULK-DATA-QUEUE
 match  dscp af11  af12  af13
class-map match-any PRIORITY-QUEUE
 match  dscp ef
 match  dscp cs5
 match  dscp cs4
!
!
policy-map 1P7Q1T-GIG3/3
 class PRIORITY-QUEUE
    police cir percent 30 bc 33 ms
      conform-action transmit
      exceed-action drop
      violate-action drop
    priority
 class CONTROL-MGMT-QUEUE
    bandwidth percent 10
 class MULTIMEDIA-CONFERENCING-QUEUE
    bandwidth percent 10
 class MULTIMEDIA-STREAMING-QUEUE
    bandwidth percent 10
 class TRANSACTIONAL-DATA-QUEUE
    bandwidth percent 10
    dbl
 class BULK-DATA-QUEUE
    bandwidth percent 4
    dbl
 class SCAVENGER-QUEUE
    bandwidth percent 1
 class class-default
    bandwidth percent 25
    dbl
policy-map 1P7Q1T-GIG3/1
 class PRIORITY-QUEUE
    police cir percent 30 bc 33 ms
      conform-action transmit
      exceed-action drop
      violate-action drop
    priority
 class CONTROL-MGMT-QUEUE
    bandwidth percent 10
 class MULTIMEDIA-CONFERENCING-QUEUE
    bandwidth percent 10
 class MULTIMEDIA-STREAMING-QUEUE
    bandwidth percent 10
 class TRANSACTIONAL-DATA-QUEUE
    bandwidth percent 10
    dbl
 class BULK-DATA-QUEUE
```

```
    bandwidth percent 4
    dbl
 class SCAVENGER-QUEUE
    bandwidth percent 1
 class class-default
    bandwidth percent 25
    dbl
!
~
!
interface GigabitEthernet3/1
 description CONNECTION TO ME-WESTDIST-3 GIG1/13
 no switchport
 ip address 10.17.100.38 255.255.255.252
 ip pim sparse-mode
 load-interval 30
 service-policy output 1P7Q1T-GIG3/1
!
~
!
interface GigabitEthernet3/3
 description CONNECTION TO ME-WESTDIST-4 GIG1/2
 no switchport
 ip address 10.17.100.41 255.255.255.252
 ip pim sparse-mode
 load-interval 30
 service-policy output 1P7Q1T-GIG3/3
!
~
```

Notice that the class-map definitions shown at the top of the configuration example are shared between the policy maps. However, a unique policy map name is applied to each of the GigabitEthernet uplink interfaces.

Referring back to Example 6-20, when a policer is applied to a queue, the bit rates of the data that conform, exceed, and violate the policer committed information rate (CIR) are also displayed within the **show policy-map interface** command. This information can provide a view of how much traffic is currently being handled by a policed queue, and whether sufficient bandwidth has been provisioned on the policer for the service classes handled by the queue. The final two highlighted sections in Example 6-20 provide an aggregate byte count of the packets handled by the particular queue, as well as the number of packets dropped because of insufficient buffer space on the queue. This holds for either the priority queue defined via the priority command, or a class-based weighted fair queueing (CBWFQ) defined via the bandwidth command. You can get an estimate of the overall data rate through a particular queue by running the **show policy-map interface** command several times over fixed time intervals and dividing the difference in byte count by the time interval.

Cisco Catalyst 3750G/3750E Series Commands

When QoS is enabled on the Cisco Catalyst 3750G/3750E Series switches with the **mls qos global** command, egress queueing consists of four queues; one of which can be a priority queue, each with three thresholds (1P3Q3T). The third threshold on each queue is pre-defined for the queue-full state (100 percent). Queue settings such as buffer allocation ratios and drop threshold minimum and maximum settings are defined based on queue-sets applied across a range of interfaces; not defined per interface. The Cisco Catalyst 3750G/3750E Series switches support two queue sets. Ports are mapped to one of the two queue-sets. By default, ports are mapped to queue-set 1. The **show platform port-asic stats drop** command allows you to view interface drops per queue on the switch port. Example 6-22 shows the output from a NME-XD-24ES-1S-P switch module within a Cisco 3845 ISR, which runs the same code base as the Cisco Catalyst 3750G.

Example 6-22 Output from Cisco Catalyst 3750G/3750E show platform port-asic stats drop Command

```
me-eastny-3#show platform port-asic stats drop fast 1/0/1

  Interface Fa1/0/1 TxQueue Drop Statistics
    Queue 0
      Weight 0 Frames 0
      Weight 1 Frames 0
      Weight 2 Frames 0
    Queue 1
      Weight 0 Frames 0
      Weight 1 Frames 0
      Weight 2 Frames 0
    Queue 2
      Weight 0 Frames 0
      Weight 1 Frames 0
      Weight 2 Frames 0
    Queue 3
      Weight 0 Frames 0
      Weight 1 Frames 0
      Weight 2 Frames 0
```

To make use of information regarding transmit queues drops shown in Example 6-22, you must understand which traffic classes are assigned to which transmit queues and which drop thresholds within those queues. For Cisco Catalyst 3750G or 3750E Series switches, the **show mls qos maps dscp-output-q** command can be used to display which DSCP values are mapped to which transmit queues and drop thresholds on the switch, as shown in Example 6-23.

Example 6-23 Output from Cisco Catalyst 3750G or 3750E Series show mls qos maps dscp-output-q Command

```
me-eastny-3#show mls qos maps dscp-output-q
   Dscp-outputq-threshold map:
     d1 :d2    0      1        2       3       4       5       6       7       8       9

------------------------------------------------------------------------------------------
      0 :    03-03 02-01 02-01 02-01 02-01 02-01 02-01 02-01 04-01 02-01
      1 :    04-02 02-01 04-02 02-01 04-02 02-01 02-01 03-01 02-01 03-01
      2 :    02-01 03-01 02-01 03-01 02-03 03-01 02-02 03-01 02-02 03-01
      3 :    02-02 03-01 01-03 04-01 02-02 04-01 02-02 04-01 02-02 04-01
      4 :    01-01 01-01 01-01 01-01 01-01 01-01 01-03 01-01 02-03 04-01
      5 :    04-01 04-01 04-01 04-01 04-01 04-01 02-03 04-01 04-01 04-01
      6 :    04-01 04-01 04-01 04-01
```

The vertical column, marked *d1*, represents the first decimal number of the DSCP value, while the horizontal column, marked *d2*, represents the second decimal number of the DSCP value. For example, a d1 value of 3 and a d2 value of 2 yields a DSCP decimal value of 32, which corresponds to the CS4 service class. This is mapped to queue 1, drop threshold 3 in Example 6-23 (highlighted in bold). Again, you still need to separately understand the mapping of specific video applications to service classes that are then marked with a particular DSCP value. However, combined with the knowledge of which traffic classes and are mapped to which transmit queue and drop threshold, you can use this information to troubleshoot video application performance issues across the Cisco Catalyst 3750G/3750E Series platforms.

To see the particular values of the buffer allocation and drop thresholds, you can issue the **show mls qos queue-set** command. An example of the output is shown in Example 6-24.

Example 6-24 Example Output From Cisco Catalyst 3750G or 3750E Switch Stack show mls qoe queue-set Command

```
me-eastny-3#show mls qos queue-set
Queueset: 1
Queue       :  1        2        3    4
----------------------------------------------------------
buffers :      30      30      35      5
threshold1 :   100     70      100     40
threshold2 :   100     80      100     100
reserved    :  50      100     50      100
maximum  :     400     100     400     100

Queueset: 2
Queue :        1        2       3     4
----------------------------------------------------------
buffers :      25      25      25      25
threshold1 :   100     200     100     100
threshold2 :   100     200     100     100
reserved :     50      50      50      50
maximum  :     400     400     400     400
```

In Example 6-24, buffers are allocated according to weight ratios across the four egress queues. *Threshold1* and *threshold2* correspond to the two configurable thresholds per queue, with the third non-configurable threshold being at 100 percent queue depth. The Cisco Catalyst 3750G and 3750E Series switches dynamically share buffer space across an ASIC that may support more than one physical interface. The reserved and maximum settings are used to control the minimum reserved buffer percentage size guaranteed per queue per port, and the maximum buffer percentage size a particular port and queue can dynamically allocate when it needs additional capacity. The combination of drop statistics per queue, mapping of DSCP value to output queue, and the buffer allocations per queue-set, can be used to determine whether sufficient bandwidth has been allocated per service class (and per application if individual video applications are mapped to separate service classes corresponding to different DSCP values) on the Cisco Catalyst 3750G/3750E Series platforms.

When configured in a switch stack, statistics such as those found within the **show platform port-asic stats drop** command are not directly accessible on member switches from the master switch. To determine which switch is the master switch, and which switch you are currently logged into within the switch stack, you can run the **show switch** command. An example of this output is shown in Example 6-25.

Example 6-25 Sample Output From Cisco Catalyst 3750G or 3750E Switch Stack show switch Command

```
me-eastny-3#show switch
Switch/Stack Mac Address : 0015.2b6c.1680
                                                     H/W   Current
Switch#  Role   Mac Address     Priority Version  State
-----------------------------------------------------------------------------
*1       Master   0015.2b6c.1680     15       0       Ready
 2       Member  001c.b0ae.bf00      1        0       Ready
```

The output from Example 6-25 shows that Switch 1 is the Master switch, and the asterisk next to Switch 1 indicates that the output was taken from a session off this switch. To access the statistics from the **show platform port-asic stats drop** command on member switches of the stack, you must first establish a session to the member switch via the **session** command. This is shown in Example 6-26.

Example 6-26 Example Output From Member Cisco Catalyst 3750G or 3750E Switch

```
me-eastny-3#session 2
```

```
me-eastny-3-2#show platform port-asic stats drop gig 2/0/24

  Interface Gi2/0/24 TxQueue Drop Statistics
    Queue 0
      Weight 0 Frames 0
      Weight 1 Frames 0
      Weight 2 Frames 0
    Queue 1
      Weight 0 Frames 0
      Weight 1 Frames 0
      Weight 2 Frames 0
    Queue 2
      Weight 0 Frames 0
      Weight 1 Frames 0
      Weight 2 Frames 0
    Queue 3
      Weight 0 Frames 0
      Weight 1 Frames 0
      Weight 2 Frames 0
```

Note that when the **session 2** command is run, the command prompt changed from *me-eastny-3* to *me-eastny-3-2*, indicating that a session to member switch #2 has been established. After the session is established to the remote switch, the **show platform port-asic stats drop** command can be run on an interface, such as GigabitEthernet 2/0/24 shown in the example above, to obtain the drop statistics per queue on the port.

Router Show Policy Map Commands

For Cisco routers, the mapping of traffic classes to egress queues over WAN interfaces is accomplished via an egress policy map applied to the interface, in the same manner as the Cisco Catalyst 4500 with a Sup-6E supervisor. Again, the policy map can be viewed through the **show policy-map interface** command. Example 6-27 shows the output from a Cisco ASR 1000 Series router with a OC-48 packet-over-SONET (POS) interface. Selected areas for discussion have been highlighted in bold.

Example 6-27 Output from Cisco 1000 Series ASR show policy-map interface Command

```
me-westwan-1#show policy-map int pos 1/1/0
 POS1/1/0
  Service-policy output: OC-48-WAN-EDGE
    queue stats for all priority classes:
      Queueing
      queue limit 512 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 18577357/16278388540
    Class-map: VOIP-TELEPHONY (match-all)
      3347 packets, 682788 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match:  ip dscp ef (46)
      police:
          cir 49760000 bps, bc 1555000 bytes, be 1555000 bytes
        conformed 3347 packets, 682788 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        violated 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps, violate 0000 bps
      Priority: Strict, b/w exceed drops: 0
```

```
Class-map: REAL-TIME-INTERACTIVE (match-all)
  18574010 packets, 16277705752 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  ip dscp cs4 (32)
  police:
      cir 821040000 bps, bc 12315600 bytes, be 12315600 bytes
    conformed 18574010 packets, 16277705752 bytes; actions:
      transmit
    exceeded 0 packets, 0 bytes; actions:
      drop
    violated 0 packets, 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps, violate 0000 bps
  Priority: Strict, b/w exceed drops: 0

Class-map: NETWORK-CONTROL (match-all)
  1697395 packets, 449505030 bytes
  30 second offered rate 1000 bps, drop rate 0000 bps
  Match:  ip dscp cs6 (48)
  Queueing
  queue limit 173 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 1644399/446219278
  bandwidth 5% (124400 kbps)
Class-map: CALL-SIGNALING (match-any)
  455516 packets, 157208585 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  ip dscp cs3 (24)
  Queueing
  queue limit 173 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 455516/157208585
  bandwidth 5% (124400 kbps)
Class-map: OAM (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  ip dscp cs2 (16)
  Queueing
  queue limit 173 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth 5% (124400 kbps)
Class-map: MULTIMEDIA-CONFERENCING (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  ip dscp af41 (34) af42 (36) af43 (38)
  Queueing
  queue limit 347 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth 10% (248800 kbps)
Class-map: MULTIMEDIA-STREAMING (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  ip dscp af31 (26) af32 (28) af33 (30)
  Queueing
  queue limit 173 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth 5% (124400 kbps)
    Exp-weight-constant: 4 (1/16)
    Mean queue depth: 0 packets
    class       Transmitted       Random drop   Tail drop   Minimum   Maximum   Mark
                    pkts/bytes pkts/bytes    pkts/bytesthresh   thresh   prob
```

```
Class-map: BROADCAST-VIDEO (match-all)
  771327514 packets, 1039749488872 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  ip dscp cs5 (40)
  Queueing
  queue limit 173 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 771327514/1039749488872
  bandwidth 5% (124400 kbps)
Class-map: TRANSACTIONAL-DATA (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  ip dscp af21 (18) af22 (20) af23 (22)
  Queueing
  queue limit 173 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth 5% (124400 kbps)
Class-map: BULK-DATA (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  ip dscp af11 (10) af12 (12) af13 (14)
  Queueing
  queue limit 139 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth 4% (99520 kbps)
Class-map: SCAVENGER (match-all)
  79 packets, 6880 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  ip dscp cs1 (8)
  Queueing
  queue limit 64 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 79/6880
  bandwidth 1% (24880 kbps)
Class-map: class-default (match-any)
  3209439 packets, 908940688 bytes
  30 second offered rate 1000 bps, drop rate 0000 bps
  Match:  any
  Queueing
  queue limit 695 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 3052981/905185696
  bandwidth 20% (497600 kbps)
    Exp-weight-constant: 4 (1/16)
    Mean queue depth: 1 packets
    class Transmitted     Random dropTail drop   Minimum   Maximum   Mark
           pkts/bytes     pkts/bytes pkts/bytes thresh     thresh  prob
     0   3052981/905185696 0/0        0/0        173         347     1/10
     1      0/0            0/0        0/0        194         347     1/10
     2      0/0            0/0        0/0        216         347     1/10
     3      0/0            0/0        0/0        237         347     1/10
     4      0/0            0/0        0/0        259         347     1/10
     5      0/0            0/0        0/0        281         347     1/10
     6      0/0            0/0        0/0        302         347     1/10
     7      0/0            0/0        0/0        324         347     1/10
```

The main difference between the router and the Cisco Catalyst 4500 switch with Sup6E is that the router implements queues in software. It is therefore not limited to eight egress queues as is the Cisco Catalyst 4500 with Sup6E. Example 6-27 shows the 12-class QoS model implemented with 12 separate egress queues over the OC-48 POS interface. Each class-map entry highlighted in bold corresponds to a queue. With this model, traffic from multiple service classes do not have to share a

single queue. This can provide a higher level of granularity into the visibility of various video applications, if separate applications are mapped to separate service classes. The traffic rate and drop rate, as well as counts of total packets and bytes outbound, and also counts of total drops for each queue can be seen from the **show policy-map interface** command when such a policy map is applied to the interface.

Simple Network Management Protocol

The Simple Network Management Protocol (SNMP) refers both to a specific protocol used to collect information and configure devices over an IP network, as well as an overall Internet-standard network management framework. The SNMP network management framework consists of the following components:

- Network management stations (NMSs)—Typically a server that runs network management applications, which in turn uses the SNMP protocol to monitor and control network elements.
- Network elements—The actual managed devices (routers, switches, TelePresence codecs, and so on) on the IP network.
- Agents—Software components running within network elements that collect and store management information.
- Managed objects—Specific characteristics of network elements that can be managed. Objects can be single entities or entire tables. Specific instances of managed objects are often referred to as variables.
- Management information bases (MIBs)—Collections of related management objects. MIBs define the structure of the management data through a hierarchical namespace using object identifiers (OIDs). Each OID describes a particular variable that can either be read from a managed object or set on a managed object. MIBs can be standards-based or proprietary. Because SNMP management information uses a hierarchical namespace, individual vendors can extend the management capabilities of their products through proprietary MIBs, which are typically published.

Currently, three versions of SNMP are commonly deployed:

- SNMPv1—The initial version introduced in the late 1980s. The security model used by SNMPv1 consists of authentication only, using community strings (read-only and read/write) that are sent in clear text within SNMP messages. Because of this, SNMPv1 is considered inherently insecure, and read/write capability should be used with caution, even over private networks.
- SNMPv2c—Proposed in the mid 1990s. The “c” in SNMPv2c indicates a simplified version of SNMPv2 that also uses a security model based on community strings. SNMPv2 improved the performance of SNMPv1 by introducing features such as the get-bulk-request protocol data unit (PDU) and notifications, both listed in Table 6-3. However, because SNMPv2c still uses the same security model as SNMPv1, read/write capability should be used with caution.
- SNMPv3—Introduced in the early 2000s, and is currently defined primarily under IETF RFCs 3411-3418. A primary benefit of SNMPv3 is its security model, which eliminates the community strings of SNMPv1 and SNMPv2. SNMPv3 supports message integrity, authentication, and encryption of messages; allowing both read and read/write operation over both public and private networks.

As mentioned above, the SNMP protocol defines a number of PDUs, some of which are shown in Table 6-3, along with the particular version of SNMP that supports them. These PDUs are essentially the commands for managing objects through SNMP.

Table 6-3 SNMP Versions and PDUs

Version	PDU	Description
SNMPv1	get-request	Command/response mechanism by which an NMS queries a network element for a particular variable
SNMPv1	response	Command/response mechanism by which an NMS receives information about a particular variable from a network element, based on a previously issued SNMP request message
SNMPv1	get-next-request	Command/response mechanism that can be used iteratively by an NMS to retrieve sequences of variables from a network element
SNMPv1	set-request	Issued by an NMS to change the value of a variable on a network element, or to initialize SNMP traps or notifications to be sent from a network element
SNMPv1	trap	Asynchronous mechanism by which a network elements issues alerts or information about an event to an NMS
SNMPv2	get-bulk-request	Improved command/response mechanism that can be used by an NMS to retrieve sequences of variables from a network element with a single command
SNMPv2	inform-request	Provides similar functionality as the trap PDU, but the receiver acknowledges the receipt with a response PDU

SNMP traps and/or informs (generically referred to a notifications) can be used to send critical fault management information, such as cold start events, link up or down events, and so on, from a medianet infrastructure device back to an NMS. This may be helpful in troubleshooting issues in which a video session has failed. SNMP GET commands can be used to pull statistics medianet infrastructure devices, which may then be used for assessing performance.

Example 6-28 shows basic configuration commands for enabling SNMP on a Cisco Catalyst 6500 Switch.

Example 6-28 Sample SNMP Configuration on a Cisco Catalyst 6500 Switch

```
me-westcore-1(config)#snmp-server group group1 v3 priv access 10
me-westcore-1(config)#snmp-server user trapuser group1 v3 auth sha trappassword priv des
privacypassword
me-westcore-1(config)#snmp-server trap-source Loopback0
me-westcore-1(config)#snmp-server ip dscp 16
```

```
me-westcore-1(config)#snmp-server host 10.17.2.10 version 3 priv trapuser
me-westcore-1(config)#snmp-server enable traps
me-westcore-1(config)#access-list 10 permit 10.17.2.10
```

This configuration creates an SNMP group called *group1* that uses SNMPv3 and *access-list 10* to limit access to only the NMS workstation at IP address 10.17.2.10. A userid called *trapuser* is associated with the SNMP group. The userid uses Secure Hash Algorithm (SHA) for authentication with password *trappassword*, and DES for encryption with password *privacypassword*.

The commands **snmp-server enable traps** and **snmp-server host 10.17.2.10 version 3 priv trapuser** cause the switch to send SNMP traps to the NMS workstation. Note that this enables all traps available on the Cisco Catalyst switch to be enabled. The network administrator may desire to pare this down to traps applicable to the configuration of the Cisco Catalyst switch. Finally, the switch is configured to send traps using the Loopback0 interface with the DSCP marking of CS2 (note that not all platforms support the ability to set the DSCP marking of SNMP data).

The SNMP group information can be displayed with the **show snmp group** command shown in Example 6-29.

Example 6-29 Sample Output From show snmp group Command on a Cisco Catalyst 6500 Switch

```
me-westcore-1#show snmp group

groupname: group1                          security model:v3 priv
readview : v1default                       writeview: <no writeview specified>

notifyview: *tv.FFFFFFFF.FFFFFFFF.FFFFFFFF.F
row status: active      access-list: 10
```

Similarly, the SNMP user information can be displayed with the **show snmp user** command shown in Example 6-30.

Example 6-30 Sample Output From show snmp user Command on a Cisco Catalyst 6500 Switch

```
me-westcore-1#show snmp user

User name: trapuser
Engine ID: 800000090300001874E18540
storage-type: nonvolatile        active
Authentication Protocol: SHA
Privacy Protocol: DES
Group-name: group1
```

Note that the specific management objects that can be accessed via SNMP depend on the platform and software version of the platform. The Cisco MIB Locator, at the following URL, can be helpful in determining supported MIBS: http://tools.cisco.com/ITDIT/MIBS/servlet/index.

Application-Specific Management Functionality

The following sections summarize the components that provide application-specific management functionality for each of the four major video application solutions that co-exist over a converged medianet infrastructure: Cisco TelePresence, Cisco Digital Media Suite, Cisco IP Video Surveillance, and Cisco Desktop Video Collaboration.

Cisco TelePresence

Within Cisco TelePresence, application-specific management functionality is distributed among the following four major components of the deployment:

- Cisco TelePresence System Manager
- Cisco TelePresence Multipoint Switch
- Cisco Unified Communications Manager
- Cisco TelePresence System endpoints

Figure 6-26 provides a high-level summary of the main management roles of each of the components of a TelePresence deployment, each of which is discussed in the following sections.

Figure 6-26 Summary of the Management Roles of the Components of a TelePresence Deployment

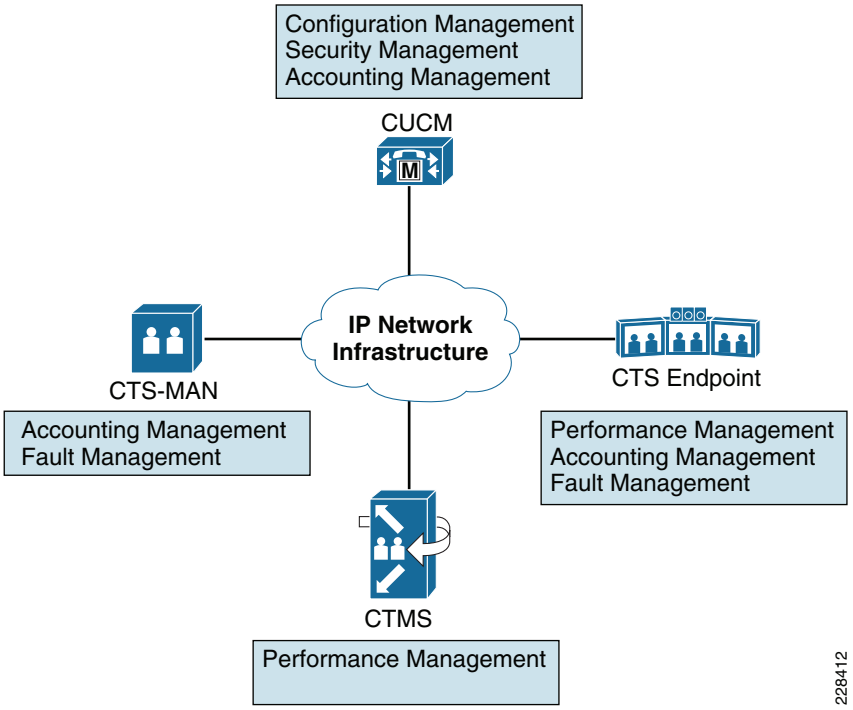


Table 6-4 highlights the application-specific management functionality of each component.

Table 6-4 Cisco TelePresence Application-Specific Management Functionality

Management Product /Tool	Management Functionality	Description
Cisco TelePresence Manager	Fault management	• The Cisco TelePresence Manager web-based GUI provides a centralized view of the status of Cisco TelePresence Multipoint Switch devices and Cisco TelePresence System endpoints; including the status of the connectivity between Cisco TelePresence System endpoints and the Cisco Unified Communications Manager, the status of connectivity between Cisco TelePresence System endpoints and the Cisco TelePresence System Manager, and the synchronization of Cisco TelePresence System rooms with the e-mail/calendaring system used for scheduling meetings. • The Cisco TelePresence Manager web-based GUI also provides a centralized view of scheduled meetings, including those that have error conditions.
	Configuration management	• The Cisco TelePresence Manager web-based GUI provides device/element management capabilities, in that the configuration of the Cisco TelePresence Manager itself is accomplished through the GUI. Limited configuration support of the Cisco TelePresence Manager is available via a Secure Shell (SSH) command-line interface (CLI) as well. • The Cisco TelePresence Manager web-based GUI also provides a centralized view of the configuration capabilities of individual Cisco TelePresence System endpoints; including features such as high-speed auxiliary codec support, document camera support, interoperability support, and so on.
	Accounting management	• The Cisco TelePresence Manager interoperates with an e-mail/calendaring system to retrieve information for meetings scheduled by end users, and update individual Cisco TelePresence System endpoints regarding upcoming meetings. • The Cisco TelePresence Manager interoperates with one or more Cisco TelePresence Multipoint Switch devices to allocate segment resources for multipoint meetings scheduled by end users. • The Cisco TelePresence Manager web-based GUI provides a centralized view of ongoing and scheduled meetings for the entire TelePresence deployment, and per individual Cisco TelePresence System endpoint.
	Security management	• The Cisco TelePresence Manager web-based GUI provides a centralized view of the web services security settings of each Cisco TelePresence System endpoint, as well as a centralized view of the security settings of scheduled and ongoing meetings. • The Cisco TelePresence Manager currently provides administrative access via the local user database only.

Table 6-4 Cisco TelePresence Application-Specific Management Functionality (continued)

Cisco Unified Communications Manager	Fault management	• The Cisco Unified Communications Manager provides limited fault management capability for Cisco TelePresence deployments. The Session Initiation Protocol (SIP) registration status of the Cisco TelePresence System endpoints to the Cisco Unified Communications Manager can be centrally viewed from the Cisco Unified Communications Manager Administration web-based GUI.
	Configuration management	• The Cisco Unified Communications Manager centrally controls the configuration of Cisco TelePresence System endpoints via the Cisco Unified Communications Manager Administration web-based GUI. • The Cisco Unified Communications Manager centrally controls the provisioning (that is, downloading of system load and device configuration) for Cisco TelePresence System endpoints via TFTP/HTTP server functionality.
	Accounting management	• Call detail records (CDRs) captured by the Cisco Unified Communications Manager can be used to determine start and stop times for Cisco TelePresence meetings. These may be used to bill back individual departments based on TelePresence room resource usage.
	Performance management	• The Cisco Unified Communications Manager Administration web-based GUI provides the ability to statically limit the amount of network bandwidth resources used for audio and video per TelePresence meeting and per overall location. **Note** Note that Cisco Unified Communications Manager location-based admission control has no knowledge of network topology.
	Security management	• The Cisco Unified Communications Manager centrally controls the security configuration of Cisco TelePresence System endpoints via the Cisco Unified Communications Manager Administration web-based GUI. • In combination with the Certificate Authority Proxy Function (CAPF) and Certificate Trust List (CTL) Provider functionality, Cisco Unified Communications Manager provides the framework for enabling secure communications (media) and signaling (call signaling, and web services) for TelePresence deployments.

Table 6-4 Cisco TelePresence Application-Specific Management Functionality (continued)

Cisco TelePresence Multipoint Switch	Fault management	• The Cisco TelePresence Multipoint Switch provides limited fault management capabilities. The web-based GUI interface can display errors and warnings for scheduled and non-scheduled meetings, as well as system errors.
	Configuration management	• The Cisco TelePresence Multipoint Switch web-based GUI provides device/element management capabilities, in that the configuration of the Cisco TelePresence Multipoint Switch itself is accomplished through the GUI. Limited configuration support of the Cisco TelePresence Multipoint Switch is available via an SSH CLI as well. • The Cisco TelePresence Multipoint Switch web-based GUI also provides the interface for administrators and meeting schedulers to configure static and ad hoc TelePresence meetings.
	Performance management	• The Cisco TelePresence Multipoint Switch web-based GUI provides centralized call statistics for multipoint calls, including SLA parameters such as bit rates, latency, drops, jitter, and so on, per Cisco TelePresence System endpoint. • The Cisco TelePresence Multipoint Switch web-based GUI also provides historical statistics for Cisco TelePresence Multipoint Switch resources including CPU utilization, traffic load per interface, packet discards, TCP connections, memory, and disk usage.
	Security management	• The Cisco TelePresence Multipoint Switch web-based GUI provides the interface for configuration of the security requirements for static and ad hoc TelePresence meetings. • Access control to the Cisco TelePresence Multipoint Switch is via the local database with three roles: administrator, meeting scheduler, or diagnostic technician.
Cisco TelePresence System Endpoint	Fault management	• The Cisco TelePresence System web-based GUI and SSH interfaces both provide device/element management capabilities, including a view of the system status, as well as diagnostics that can be used to troubleshoot the camera, microphone, and display components of the Cisco TelePresence System endpoint. • SIP Message log files accessed through the Cisco TelePresence System web-based GUI can be used to troubleshoot SIP signaling between the Cisco TelePresence System endpoint and Cisco Unified Communications Manager. • Additional Cisco TelePresence System log files can be collected and downloaded via the web-based GUI to provide system-level troubleshooting capabilities. • Status of peripheral devices (cameras, displays, microphones, and so on) can be accessed centrally via SNMP through the CISCO-TELEPRESENCE-MIB.
	Configuration management	• The Cisco TelePresence System web-based GUI and SSH interfaces both provide information regarding current hardware and software versions and current configuration of the Cisco TelePresence System endpoint. Limited configuration is done on the Cisco TelePresence System endpoint itself. Most of the configuration is done via the Cisco Unified Communications Manager Administrator web-based GUI.

Table 6-4 Cisco TelePresence Application-Specific Management Functionality (continued)

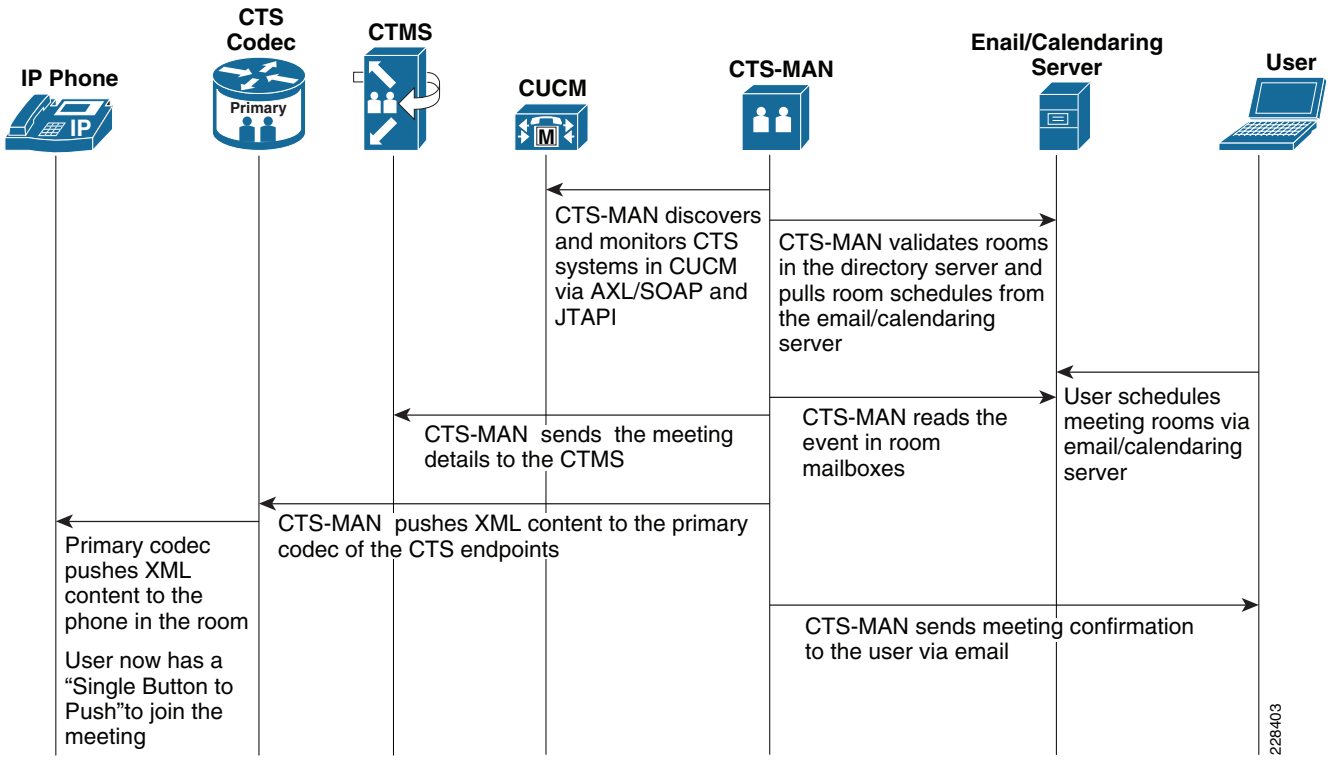
	Accounting management	• The Cisco TelePresence System web-based GUI provides access to statistics for ongoing calls, or the previous call if the Cisco TelePresence System endpoint is currently not in a call. Accounting management statistics include the call start time, duration of the call, remote number, bit rate, and the number of packets and bytes transmitted and received during the call. These statistics are also available via an SSH CLI as well as through SNMP.
	Performance management	• The Cisco TelePresence System web-based GUI provides access to statistics for ongoing calls, or the previous call if the Cisco TelePresence System endpoint is currently not in a call. Performance management statistics include parameters such as packet loss, latency, jitter, and out-of-order packets for audio and video media streams. These can be used to assess the performance of the network infrastructure in meeting service level agreements. These statistics are also available via SNMP through the CISCO-TELEPRESENCE-CALL-MIB. • An IP service level agreements (IPSLA) responder within the Cisco TelePresence System endpoint can be enabled, allowing the Cisco TelePresence System endpoint to respond to packets sent by an IPSLA initiator. IPSLA can be used to pre-assess network performance before commissioning the Cisco TelePresence System endpoint onto a production network, or used to assess ongoing network performance or when troubleshooting.
	Security management	• Access control to the individual Cisco TelePresence System endpoints is currently handled via a local database, although the userid and password used for access control are centrally managed via the configuration within the Cisco Unified Communications Manager Administration web-based GUI. • SNMP notifications can be set on the Cisco TelePresence System endpoint to alert after failed access control attempts.

Note Both static location-based admission control and RSVP are considered part of performance management within this document, because the scheduling of resources is not done per end user, but to ensure that necessary resources are allocated to meet service level requirements.

Cisco TelePresence Manager

From a management perspective, the primary functions of Cisco TelePresence Manager are resource allocation, which is part of accounting management; and fault detection, which is part of fault management. Cisco TelePresence Manager allocates Cisco TelePresence System endpoints (meeting rooms) and Cisco TelePresence Multipoint Switch segment resources based on meetings scheduled by end users through an e-mail/calendaring system such as Microsoft Exchange or IBM Lotus Domino. Note that the Cisco TelePresence Manager has no knowledge of the underlying IP network infrastructure, and therefore has no ability to schedule any network resources or provide Call Admission Control (CAC) to ensure that the TelePresence call goes through during the scheduled time. Figure 6-27 shows an example of the resource scheduling functionality of Cisco TelePresence Manager.

Figure 6-27 Cisco TelePresence Manager Resource Scheduling

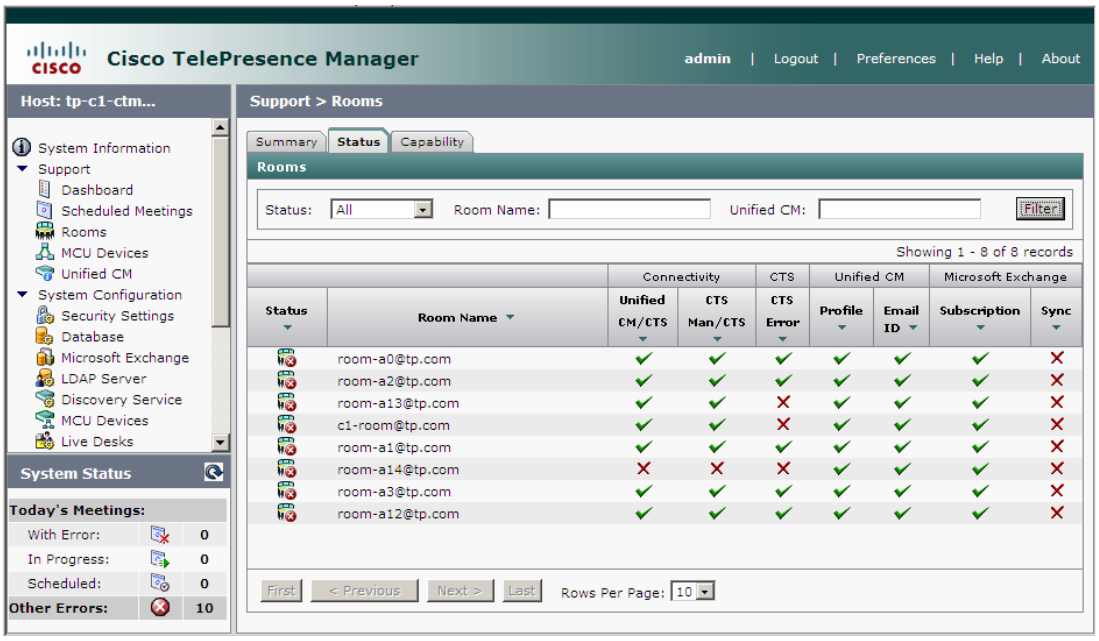


Cisco TelePresence Manager periodically queries the e-mail/calendaring system to determine whether an end user has scheduled TelePresence rooms for an upcoming meeting. Having previously synchronized the TelePresence rooms defined within the Cisco Unified Communications Manager database with the TelePresence rooms defined within the e-mail/calendaring system database, the Cisco TelePresence Manager then pushes the meeting schedule the IP Phone associated with each TelePresence room. If a multipoint meeting has been scheduled by the end user, the Cisco TelePresence Manager selects an appropriate Cisco TelePresence Multipoint Switch for the meeting, and schedules the necessary resources for the meeting. The Cisco TelePresence Multipoint Switch then updates the end user via an e-mail confirmation.

Note In Figure 6-27 and throughout this chapter, the CTS codec and associated IP phone together are considered the CTS endpoint. The IP phone shares the same dial extension as the CTS codec, is directly connected to it, and is used to control TelePresence meetings.

The other primary management function of the Cisco TelePresence Manager is fault detection. Cisco TelePresence Manager includes the ability to centrally view error conditions in the various components of a Cisco TelePresence deployment. It also allows you to view error conditions that resulted in the failure of scheduled meetings. Figure 6-28 shows an example of the Cisco TelePresence Manager screen used to view the status of Cisco TelePresence System endpoints.

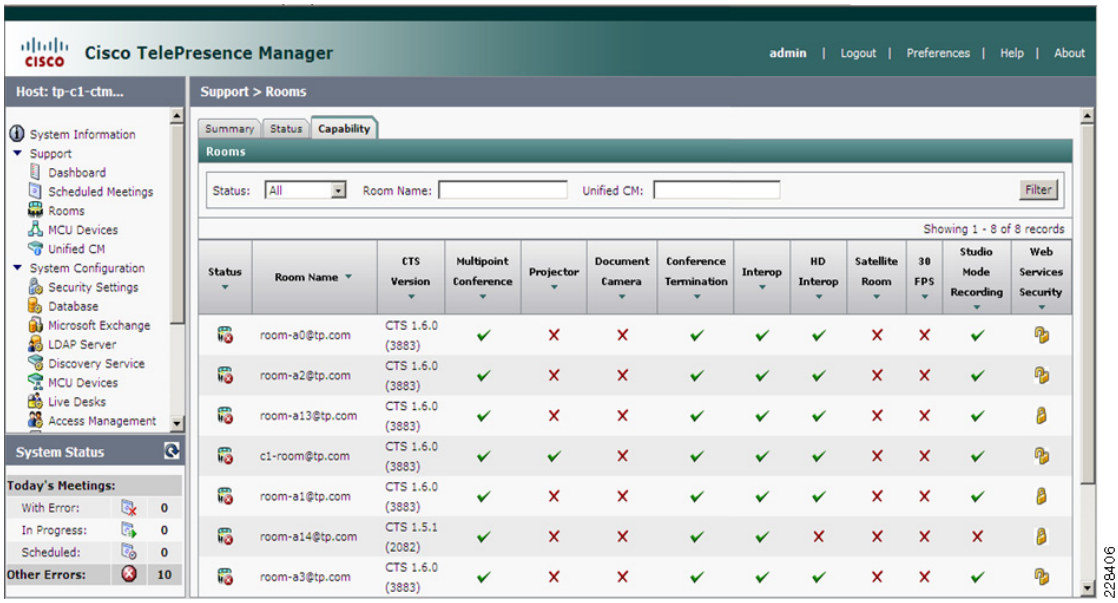
Figure 6-28 Fault Detection Using the Cisco TelePresence Manager



As shown in Figure 6-28, a red X indicates some type of error condition that may need to be further investigated. These error conditions can include communication problems between the Cisco TelePresence System endpoint and the Cisco Unified Communications Manager, or between the Cisco TelePresence System endpoint and the Cisco TelePresence Manager itself. Other error conditions include problems within the Cisco TelePresence System endpoint itself, such as an issue with one of the peripherals (cameras, displays, and so on). Still other error conditions include synchronizing the TelePresence room defined within the Cisco Unified Communications Manager with the room definition within the e-mail/calendaring system. The System Status panel in the lower left corner of Figure 6-28 provides information regarding whether any meetings scheduled for the current day had errors. By clicking on the icons within the panel, you can gain additional details about the scheduled meetings and error conditions.

The Cisco TelePresence Manager also plays a minor role in both configuration management and security management. Cisco TelePresence Manager allows central viewing of specific configured features supported by a particular Cisco TelePresence System endpoint, such as a projector, document camera, or high-speed auxiliary codec support. It also allows you to centrally view the web service security settings for particular Cisco TelePresence System endpoints. Both of these functions are illustrated in Figure 6-29.

Figure 6-29 Cisco TelePresence Manager View of Configuration and Security Options for Cisco TelePresence System Endpoints

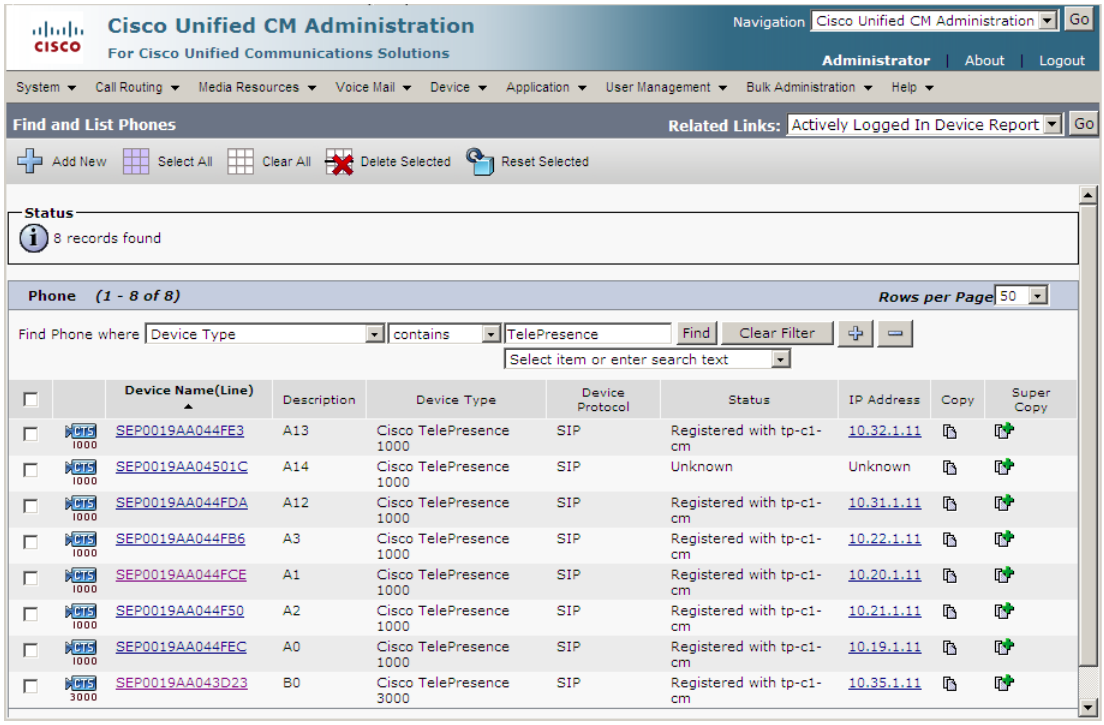


A red X indicates that the particular feature is either not configured or currently unavailable on the Cisco TelePresence System endpoint. A locked padlock indicates that web services communications from the Cisco TelePresence System endpoint are secured via Transport Layer Security (TLS). An open padlock indicates that web services communications from the Cisco TelePresence System endpoint are in clear text. Note that this functionality allows the viewing of only certain capabilities configured on the Cisco TelePresence System endpoint. All changes to the Cisco TelePresence System endpoint configuration are handled through the Cisco Unified Communications Manager, which is discussed next.

Cisco Unified Communications Manager

From an overall TelePresence deployment perspective, the primary function of the Cisco Unified Communications Manager is a SIP back-to-back user agent for session signaling. However, the Cisco Unified Communications Manager also plays a central management role for TelePresence deployments. From an FCAPS perspective, the primary roles of the Cisco Unified Communications Manager are in configuration management and security management. The device configuration and software image version for each of the Cisco TelePresence System endpoints is centrally managed through the Cisco Unified Communications Manager Administration web-based GUI, and downloaded to each Cisco TelePresence System endpoint when it boots up. The Cisco Unified Communications Manager therefore plays a central role in the initial provisioning of Cisco TelePresence System endpoints onto the network infrastructure, as well as any ongoing changes to the configuration of the Cisco TelePresence System endpoints. Figure 6-30 provides an example of the Cisco Unified Communications Manager Administrator web page, showing the TelePresence endpoints configured for the particular Cisco Unified Communications Manager.

Figure 6-30 Centralized Configuration Management via the Cisco Unified Communications Manager



The detailed configuration for each Cisco TelePresence System endpoint can be viewed and modified by clicking on each device listed under the Device Name column shown in Figure 6-30. Included within the configuration of each Cisco TelePresence System endpoint is the security configuration. TelePresence security includes the use of Secure Real-time Transport Protocol (SRTP) for confidentiality and data authentication of the audio and video media streams; as well as TLS for confidentiality and data authentication of the SIP signaling and web services signaling between the various TelePresence components. For a thorough discussion of Cisco TelePresence security, see *Cisco TelePresence Secure Communications and Signaling* at the following URL: http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/telepresence.html.

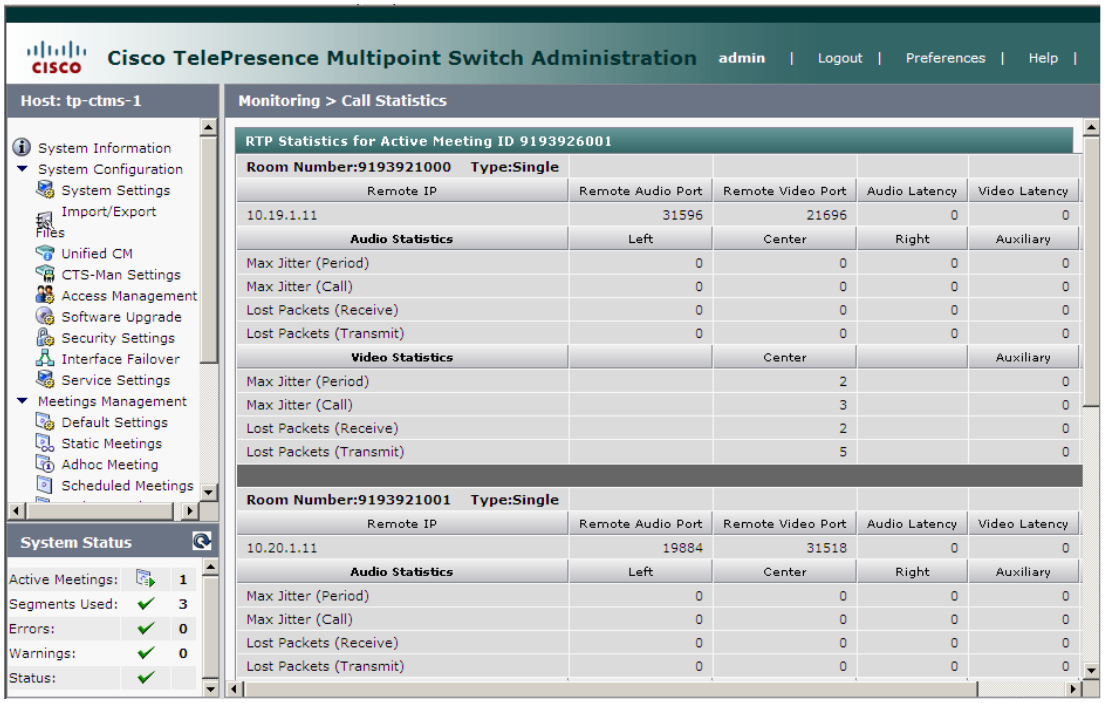
Cisco Unified Communications Manager also plays a role in accounting management, in that call detail records (CDRs) can be captured and used to bill back end users for TelePresence room usage. Cisco Unified Communications Manager can also play a role in performance management, in terms of bandwidth allocation, using static location-based CAC, although it is not in widespread use today for TelePresence deployments. The amount of bandwidth used for the audio and video components of an individual TelePresence call can be centrally controlled per zone via Cisco Unified Communications Manager. Also the total amount of bandwidth allocated for aggregate audio and video traffic to and from a location can be centrally controlled, via Cisco Unified Communications Manager. When a new TelePresence call requested via SIP signaling results in the amount of bandwidth allocated either for the individual call or aggregated for the entire location exceeding the configured zone or location bandwidth, the new call does not proceed. This helps maintain the overall quality of ongoing TelePresence calls. Because static location-based CAC has no knowledge of the underlying network infrastructure, it is typically effective only in hub-and-spoke network designs. Cisco offers location-based CAC integrated with Resource Reservation Protocol (RSVP), using an RSVP agent device, for VoIP and Cisco Unified Communications Manager-based Desktop Video Conferencing. However, this is currently not supported for Cisco TelePresence deployments.

Finally, the Cisco Unified Communications Manager plays a minor role in fault management. The SIP registration state of Cisco TelePresence System endpoints can be centrally viewed, and faults detected, from the Cisco Unified Communications Manager Administration web-based GUI interface, as shown in the Status column of Figure 6-30.

Cisco TelePresence Multipoint Switch

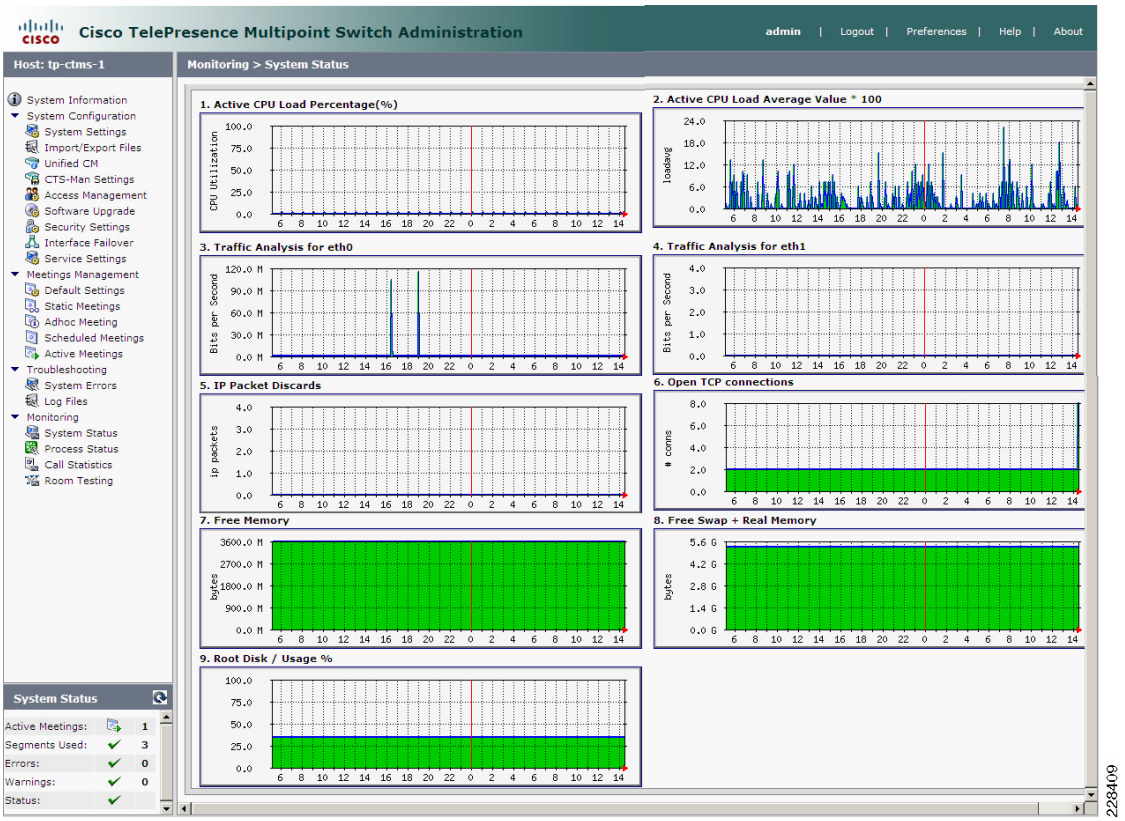
From an overall TelePresence deployment perspective, the primary function of the Cisco TelePresence Multipoint Switch is to provide switching of the video and audio media for multipoint TelePresence calls. However, as with the Cisco Unified Communications Manager, the Cisco TelePresence Multipoint Switch also plays a management role for TelePresence deployments. From an FCAPS perspective, the primary function of Cisco TelePresence Multipoint Switch is in performance management. The Cisco TelePresence Multipoint Switch can collect performance data regarding the Cisco TelePresence System endpoints in an ongoing multipoint meeting. Figure 6-31 shows an example of the call statistics collected by the Cisco TelePresence Multipoint Switch for one of the Cisco TelePresence System endpoints within a three-party multipoint call.

Figure 6-31 Cisco TelePresence Multipoint Switch Performance Statistics for Ongoing Meetings



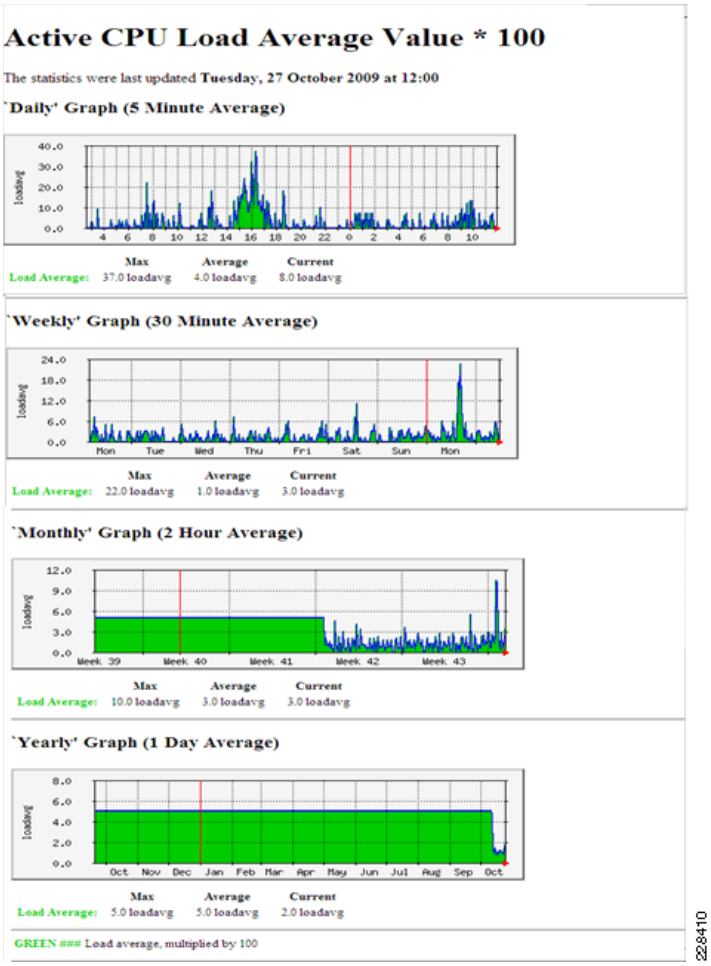
Call statistics include the maximum jitter seen for the last period (ten seconds), the maximum jitter seen for the duration of the call, latency, and lost packets in both the transmit and receive directions. These statistics are collected by the Cisco TelePresence Multipoint Switch for both the audio and video channels for each of the endpoints. Cisco TelePresence Multipoint Switch call statistics can be used to quickly view whether any leg of a multipoint call is outside the required service level agreement (SLA) parameters of jitter, packet loss, and latency. Statistics regarding the overall status of the Cisco TelePresence Multipoint Switch are also collected, as shown in Figure 6-32. These statistics include CPU loading of the Cisco TelePresence Multipoint Switch, traffic loading for the FastEthernet interfaces, Cisco TelePresence Multipoint Switch memory and disk utilization, open TCP connections, and Cisco TelePresence Multipoint Switch packet discards.

Figure 6-32 Cisco TelePresence Multipoint Switch Statistics for Overall Status



Each of the categories shown in Figure 6-32 can be expanded by clicking on it. For example, the Active CPU Load Average Value * 100 statistics can be expanded, as shown in Figure 6-33. This provides detail regarding CPU utilization on a daily, weekly, monthly, and yearly basis.

*Figure 6-33 Expanded Statistics for Active CPU Load Average Value * 100*



The statistics collected by the Cisco TelePresence Multipoint Switch can be used to perform long-term trend analysis, allowing you to plan the deployment of additional Cisco TelePresence Multipoint Switch resources before capacity limits are reached and service is degraded.

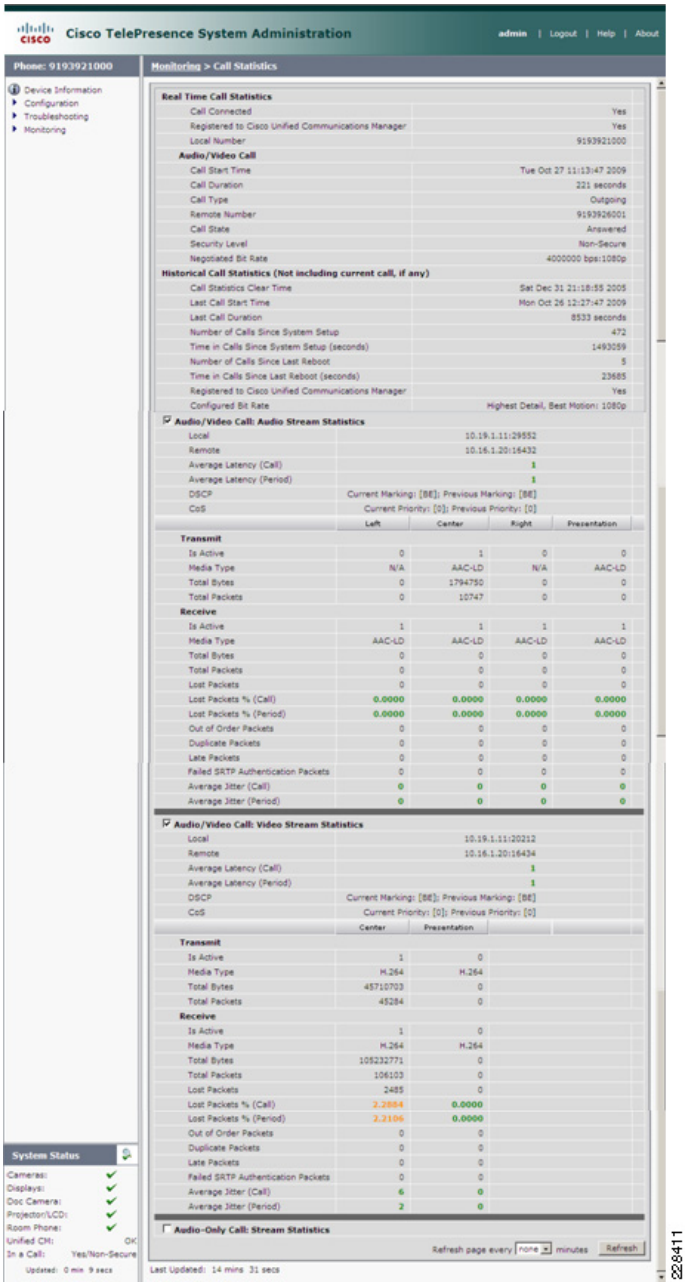
The Cisco TelePresence Multipoint Switch also plays a role in both configuration management and security management. Static and ad hoc meetings, as well as the security requirements for those meetings, are configured directly on the Cisco TelePresence Multipoint Switch by network administrators or meeting schedulers. Meetings can be configured as non-secured, secured, or best effort. Best effort means that if all endpoints support encryption, the call goes through as secured. However, if any endpoint does not support encryption, the call falls back to an unencrypted or non-secured call. Access control to the Cisco TelePresence Multipoint Switch is controlled through its local database, with the capability of defining three roles: administrators, who have full access to the system; meeting schedulers, who can only schedule static or ad hoc meetings; and diagnostic technicians, who can perform diagnostics on the Cisco TelePresence Multipoint Switch.

Finally, the Cisco TelePresence Multipoint Switch plays a minor role in fault management. The Cisco TelePresence Multipoint Switch logs system errors as well as error or warning conditions regarding meetings. For example, a error message might indicate that a Cisco TelePresence System endpoint cannot join a secure multipoint meeting because it is not configured to support encryption. The error messages can be viewed via the web-based GUI interface of the Cisco TelePresence Multipoint Switch.

Cisco TelePresence System Endpoint

From an overall TelePresence deployment perspective, the primary function of the Cisco TelePresence System endpoint is to transmit and receive the audio and video media for TelePresence calls. However, from an FCAPS management perspective, the Cisco TelePresence System endpoint also plays a role in performance management. The Cisco TelePresence System endpoint collects statistics regarding ongoing TelePresence meetings, or the previous meeting if the device is not in a current call. These can be viewed through the Cisco TelePresence System web-based GUI interface, as shown in the example in Figure 6-34.

Figure 6-34 Cisco TelePresence System Endpoint Call Statistics



As with the Cisco TelePresence Multipoint Switch, statistics are collected for both the audio and video channels for each of the endpoints. The statistics include SLA parameters such as average latency for the period (ten seconds) and for the call; average jitter for the period and the call; percentage of lost packets for the period and the call; as well as total lost packets, out of order packets, late packets, or duplicate packets. They also include some accounting management information such as the call start time, call duration, and the remote phone number; as well as the bandwidth of the call, and the number of bytes or packets sent and received. These statistics can also be collected and stored centrally via SNMP through the CISCO-TELEPRESENCE-CALL-MIB supported by Cisco TelePresence System endpoints running Cisco TelePresence System version 1.5 or higher software. These statistics can then be used for performance analysis and/or billing purposes. A more limited set of call statistics, primarily the accounting management statistics, is available through the SSH CLI, as shown in Example 6-31.

Example 6-31 Call Statistics Available via the SSH Command-Line Interface

```
admin:show call statistics all

       Call Statistics
Registered to Cisco Unified Communications Manager : Yes
Call Connected: Yes

Call type      : Audio/Video Call Call Start Time: Oct 27 11:48:29 2009
Duration (sec) : 2119             Direction: Outgoing
Local Number   : 9193921003       Remote Number: 9193926001
State          : Answered         Bit Rate: 4000000 bps,1080p
Security Level : Non-Secure

    -- Audio --
IP Addr Src: 10.22.1.11:25202     Dst : 10.16.1.20:16444
Latency Avg: 1                    Period: 1

Statistics     Left       Center     Right      Aux
Tx Media Type  N/A        AAC-LD     N/A        AAC-LD
Tx Bytes       0          17690311   0          0
Tx Packets     0          105930     0          0
Rx Media Type  AAC-LD     AAC-LD     AAC-LD     AAC-LD
Rx Bytes       0          0          0          0
Rx Packets     0          0          0          0
Rx Packets Lost0          0          0          0

    -- Video --
IP Addr Src:    10.22.1.11:20722   Dst : 10.16.1.20:16446
Latency Avg:    1                  Period: 1

Statistics                Center         Aux
Tx Media Type  H.264      H.264
Tx Bytes                  1068119107     0
Tx Packets                1087322        0
Rx Media Type  H.264      H.264
Rx Bytes                  1067246669     0
Rx Packets                1055453        0
Rx Packets Lost           1876           0

    -- Audio Add-in --
IP Addr Src:    10.22.1.11:0   Dst : 0.0.0.0:0
Latency Avg:    N/A            Period: N/A

Statistics                Center
Tx Media Type  N/A
Tx Bytes                  0
Tx Packets                0
Rx Media Type  N/A
```

```
Rx Bytes                  0
Rx Packets                0
Rx Packets Lost           0
```

In addition to passive collection of statistics during calls, Cisco TelePresence System endpoints can also function as IPSLA responders, as of Cisco TelePresence System version 1.4 or higher. IPSLA can be used to pre-assess network performance before commissioning the Cisco TelePresence System endpoint onto a production network. Optionally, IPSLA can be used to assess network performance when troubleshooting a performance issue of a production device. See Network-Embedded Management Functionality, page 6-2 for more information regarding the use of IPSLA for performance management.

The Cisco TelePresence System endpoint also supports extensive fault management capabilities through diagnostics that can be used to troubleshoot the camera, microphone, and display components of the Cisco TelePresence System endpoint. These diagnostics can be accessed through either the web-based GUI interface of the Cisco TelePresence System endpoint, or through the SSH CLI. Additionally, SIP log files stored within the Cisco TelePresence System endpoint can be accessed through the web-based GUI to troubleshoot call signaling between the Cisco TelePresence System endpoint and the Cisco Unified Communications Manager. Finally, the status of each component (displays, microphones, speakers, and so on) of the Cisco TelePresence System endpoint can be accessed centrally via SNMP through the CISCO-TELEPRESENCE-MIB. This management information base (MIB) is supported on Cisco TelePresence System endpoints running software version 1.5 and higher.

The Cisco TelePresence System endpoint itself also plays a minor role in configuration management and security management. In terms of configuration management, the configuration of the Cisco TelePresence System endpoint, including specific hardware and software levels of each component (displays, microphones, speakers, and so on), can be viewed through the web-based GUI interface, or accessed through the SSH CLI. However, modifications to the configuration of the Cisco TelePresence System endpoint is primarily controlled centrally by the Cisco Unified Communications Manager. In terms of security management, access to the Cisco TelePresence System endpoint is via its local database. However, the userid and passwords are configured centrally within the Cisco Unified Communications Manager and downloaded to the Cisco TelePresence System endpoint.
Cisco TelePresence System 1.6 introduces password aging for the SSH and web-based GUI interface of the Cisco TelePresence System endpoints. The security settings of the Cisco TelePresence System endpoint are controlled via the Cisco Unified Communications Manager centrally, as discussed previously. Finally, the Cisco TelePresence System endpoint also supports the ability to generate SNMP traps for authentication failures when attempting to access the system. This can be used to monitor the Cisco TelePresence System endpoints against brute-force password attacks.

Cisco TelePresence SNMP Support

As of this writing (CTS version 1.6), CTS, CTMS, and CTS Manager support the MIBs listed in Table 6-5. Future versions of Cisco TelePresence may add additional SNMP MIB support.

Table 6-5 MIB Support in TelePresence Endpoints (CTS, CTMS, and CTS-MAN)

MIB Name	Description
CISCO-SYSLOG-MIB	Provides an SNMP interface into syslog messages
CISCO-CDP-MIB	Provides Ethernet neighbor information, such as the attached IP phone and upstream switch
HOST-RESOURCES-MIB	Provides system operating system information such as system CPU, memory, disk, clock, and individual process information

Table 6-5 MIB Support in TelePresence Endpoints (CTS, CTMS, and CTS-MAN) (continued)

RFC-1213-MIB	Provides basic MIB2 structure/information such as system uptime, system description, SNMP location, and SNMP contact
IF-MIB	Provides Ethernet interface statistics, such as bytes and packets transmitted and received, as well as interface errors
UDP-MIB	Provides the number of inbound and outbound UDP packets, as well as drops
TCP-MIB	Provides the number of inbound and outbound TCP packets, connections, and number of TCP retransmissions
CISCO-TELEPRESENCE-MIB	Provides notification on peripheral and user authentication failures; also allows for the remote restart of the CTS device
CISCO-TELEPRESENCE-CALL-MIB	Provides detailed call statistics for TelePresence meetings
CISCO-ENVMON-MIB	Provides system temperature
SNMP protocol-specific MIBs: • SNMP-FRAMEWORK-MIB • SNMP-MPD-MIB • SNMP-NOTIFICATION-MIB • SNMP-TARGET-MIB • SNMP-USM-MIB • SNMP-VACM-MIB	Provides information relating to the SNMP daemon configuration and current state

IP Video Surveillance

For information regarding the medianet management functionality of the Cisco IP Video Surveillance solution, see the *Cisco IP Video Surveillance Design Guide* at the following URL: http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/IPVS/IPVS_DG/IPVS_DG.pdf.

Digital Media Systems

For information regarding the medianet management functionality of the Cisco Digital Media Systems solution, see the *Cisco Digital Media System 5.1 Design Guide for Enterprise Medianet* at the following URL: http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/DMS_DG/DMS_DG.html.

Desktop Video Collaboration

Future revisions of this document will include discussion regarding medianet management functionality for Cisco Desktop Video Collaboration solutions.

Summary

This design chapter has focused on functionality that can be used to provide increased visibility and management of video flows within an enterprise medianet. From a high-level perspective, the functionality can be separated into two broad categories: application-specific management functionality and network-embedded management functionality. Application-specific management refers to functionality within the components of a particular video solution: Cisco TelePresence, Cisco IP Video Surveillance, Cisco Digital Media Systems, and Cisco Desktop Video Collaboration. Network-embedded management refers to functionality embedded within the medianet infrastructure itself, which allows both visibility and management of video flows. These include specific embedded software features such as NetFlow and IPSLA, the Cisco router and Cisco Catalyst switch CLI itself, and also hardware modules such as the Cisco NAM embedded within Cisco Catalyst 6500 Series Switches. By implementing a QoS model that separates the various video applications into different service classes, which are then mapped to separate queues and drop thresholds within Cisco router and switch platforms, you can gain additional visibility into the video applications themselves by collecting flow information based on DSCP aggregation, as well as monitoring the router and switch queues. Typically, the more granular the QoS model (that is, up to 12 service classes) and the more queues and drop thresholds deployed throughout medianet infrastructure devices, the greater the visibility and ability to manage the flows.