

# Video Quality of Service (QoS) Tutorial

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Objective](#)

[What this doesn't cover](#)

[Network traffic characteristics of video](#)

[Video quality measurement](#)

[Controls at endpoints](#)

[Visible artifacts](#)

–

[Transport network SLAs for video quality](#)

[Controls in the transport network](#)

[Video varieties](#)

[Video traffic codecs](#)

[QoS mechanisms for Video](#)

[Bandwidth guarantee](#)

[Queuing](#)

[Header Compression](#)

[Link fragmentation and interleaving](#)

[Congestion avoidance](#)

[Burst](#)

[How much Bandwidth?](#)

[Resolution](#)

[Frame rate](#)

[Bandwidth calculation](#)

[Classifying/marketing video traffic](#)

[Configuration](#)

[CUBE bandwidth handling](#)

[Video codec Payload types](#)

[Monitoring/measuring](#)

[IP SLAs Video](#)

[CUBE VQM](#)

[Reference](#)

## Introduction

This document reviews the subject of video call quality and provides a tutorial on things to keep in mind while Quality of Service (QoS) is configured on a Cisco Unified Border Element(CUBE) or a

Time-Division Multiplexing (TDM) gateway.

Contributed by Baktha Muralidharan, Cisco TAC Engineer, edited by Anoop Kumar.

## Prerequisites

### Requirements

This document is most beneficial for the engineers familiar with voice over IP (VoIP), although others might find it useful.

### Components Used

There is no specific hardware or software used to write this document.

## Background Information

Digitized audio in its simplest form is a set of audio samples, each sample describing the sound pressure during that period. Conversational audio can be captured and reproduced to a high degree of accuracy, with just 8000 samples per second[1]. This then means that as long as the network is able to transport the samples without excessive delay, jitter and packet loss, audio can be faithfully reproduced at the other end.

In contrast presentation, processing and transport of video is much more complex. Brightness, contrast, color saturation, responsiveness (to motion) and lip-sync are just some of the attributes that determine the quality of the video. Video samples generally require much larger space. Not surprisingly, video places a much larger demand on network bandwidth, on the transport network. Audio quality is determined by :Microphone Speaker in the headset Codec - compression transport network video call quality is affected by: Camera Display device Video codec Transport network Compatibility/Interoperability

**Note:** It is important to understand that unlike audio, quite a bit goes on at video endpoints, when it comes to tuning quality.

### Objective

QoS in general is a vast and complex subject requiring consideration of overall traffic requirements (rather than just the traffic you wish to improve the quality of) and needs to be checked on every network component along the path of the media flow. Achieving video quality in a video conference is even more complex as it involves in addition to the *network* components, review and examination of configuration and tuning at the endpoints. Broadly, video quality entails this:

- End-point tuning- Optimizing the configuration of endpoints (e.g. resolution, frame-per-second)
- Transport optimization- Optimizing the network to transport the video traffic per network SLA.
- Interoperability considerations- Quite often video calls involve endpoints of varied capabilities.

Designing and configuring the systems to maximize *interoperability* can impact video quality.

The specific focus in this document will be the QoS considerations on the IOS gateway or CUBE when handling video calls.

Tuning at the endpoints would involve adjust a set of parameters on the video endpoints. This of course depends on the product but here are a few general “knobs” :

- Resolution (i.e. picture size)
- Frame rate (i.e. motion sensitivity/reality)
- Tagging (i.e. ToS marking)

Tuning the network for video generally involves the following :

- Understanding the composition of traffic flowing through the CUBE- e.g. peak [call] volume etc.
- Reviewing network link/pipe capacity
- Designing appropriate QoS policies, to ensure SLA is met for each traffic class

Interoperability comes into play when heterogeneous (video telephony as well as telepresence (TP)) systems participate in a conference call. The experience provided by a TP and video phone system are fundamentally different. Interoperability between them is generally achieved by bridging them using a process known as *cascading*.

## What this doesn't cover

This is not a design document and not a comprehensive video QoS document either. Specifically this document does not cover these topics:

- Signaling [protocols] of video calls, beyond what is required to illustrate qos-related aspects.
- Video Endpoint setup/configuration
- Comprehensive review of QoS mechanisms including Policing, Queuing, Shaping and Burst
- Review of QoS config on Layer 2 switches or trust boundary considerations.

## Network traffic characteristics of video

Video, like audio is real-time. Audio transmissions are constant-bit-rate (CBR). In contrast, video traffic tends to be *bursty* and is referred to as being variable-bit-rate(VBR.) Consequently bit rate for video transmission will necessarily not be constant, if we need to maintain a certain quality<sup>[2]</sup>.

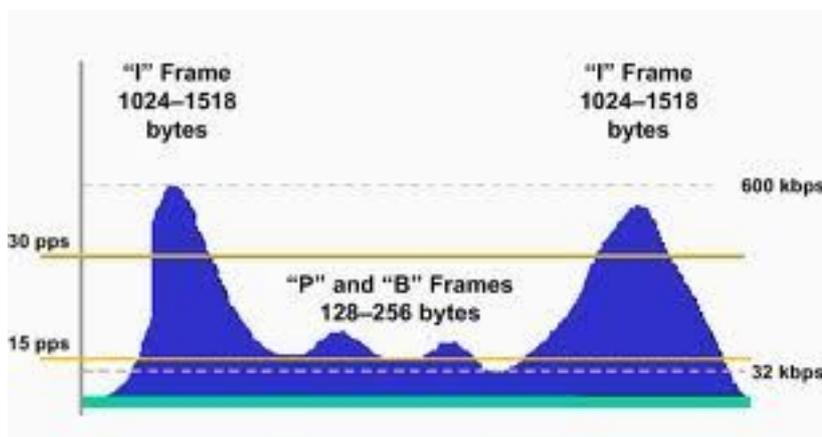


Image 1

Determination of bandwidth and bursting required for video is also more involved. This is discussed later in this document.

- Video traffic is bursty.
- Video packets can be quite large.
- Audio is always CBR. Video is typically VBR.

### Why is video bursty?

The answer lies in the way video is compressed. Remember that video is a sequence of images (frames) played to provide a visual motion effect. Compression techniques used by video codecs use an approach called Delta encoding<sup>[3]</sup>, which works by storing values of bytes as differences (deltas) between sequential (samples) values rather than the values themselves. Accordingly video is encoded (and transmitted) as consecutive frames carrying just the “moving parts” rather than whole frames.

You are probably wondering *Why, the audio changes incrementally too ?* Well, true enough, but “motion” (or dynamics) doesn’t impact audio nearly as much as it does video. The 8-bit audio [samples](#) do not compress better when delta encoded, video samples (frames) do. The *relative change* from sample (frame to frame) to sample is video is much smaller than that in audio. Depending on the nature and degree of *motion*, video samples can greatly vary in size. Image 2 illustrates video compression-

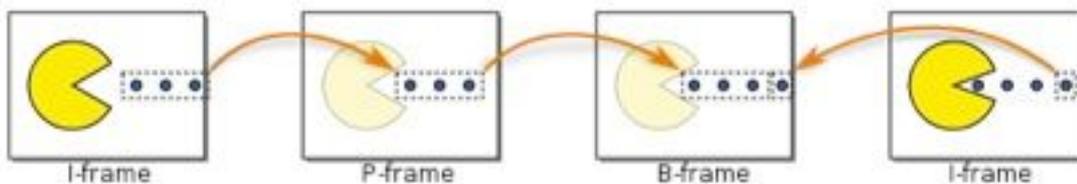


Image 2

An **Iframe** is an Intra-coded picture, in effect a fully specified picture, like a conventional static image file.

A **Pframe** (Predicted picture) holds only the changes in the image from the previous frame. The encoder does not need to store the unchanging background pixels in the Pframe, thus saving space. Pframes are also known as *deltaframes*.

A **Bframe** (Bi-predictive picture) saves even more space by using differences between the current frame and both the preceding and following frames to specify its content.

## Video quality measurement

Cisco video gear do not measure or report on video quality as such, so video quality is *perceived* rather than measured. There are standardized algorithms that measure quality by means of a MOS (Mean Opinion Score). However, if issues reported on audio-quality are any indication, video quality (TAC) cases are more likely to be opened because user perceived quality issues rather than reports by a tool.

## Controls at endpoints

Factors that affect video quality include:

- the video codec (MPEG4, H261, H263, H264 & H265)

- size (1/8th screen, 1/4 screen, full screen)
- frame rate (1 to 30 frames per second, 6 default)
- the compression quality setting (low, medium, high)

Generally each of the above is selectable/controllable at endpoints.

## Visible artifacts

**Quilting, Combing & Banding** get used to these terms, part of video impairment taxonomy. Refer to this document for details on the common video impairments:

Ref:

[http://www.estadistica.cl/public\\_disk/Programas/DeTodo/Manejo\\_y\\_pegado\\_de\\_subtitulos/VirtualDub-1.6.14/help/p-artifacts.html](http://www.estadistica.cl/public_disk/Programas/DeTodo/Manejo_y_pegado_de_subtitulos/VirtualDub-1.6.14/help/p-artifacts.html)

## Transport network SLAs for video quality

Recommended network SLA for video<sup>[4]</sup> is as follows:

- Latency 150–300ms
- Jitter 10 ms–50ms
- Loss 0.5%

Incidentally the recommended network SLA for transporting audio are:

- Latency 150–300ms
- Jitter 20 ms–50ms
- Loss 1%

**Note:** Clearly Video is more sensitive to packet loss than voice. This should be expected once you understand that interframes require information from previous frames, which means that loss of interframes can be devastating to the process of reconstructing the video image.

## Controls in the transport network

Generally the SLA for video transport can be delivered using QoS policies that are very similar to those used for audio transport. There are some differences however owing to the nature of video traffic.

**Note:** Although the scope of this document is limited to the CUBE component, remember QoS is end-to-end.

## Video varieties

Are all video same? Well, not quite. The variations of video as a medium include:

- Video telephony/Video conferencing
  - Real-time interactive
  - Relatively lower bandwidth. Up to approx. 1Mbps
- Telepresence
  - Real-Time interactive
  - *Immersive* experience
  - Requires very high bandwidth
- Streaming
  - Real-time, Unidirectional
  - Can be unicast or multicast
  - High bandwidth
  - Not delay sensitive (the video can take several seconds to queue up)
  - Largely insensitive to jitter (because of application buffering)
  - Loss should be no more than 5 percent.
  - Latency should be no more than 4 to 5 seconds (depending on the video application's buffering capabilities)
  - Some video (e.g. entertainment) might be considered for Scavenger service.

**Note:** In the interest of brevity, illustrations are not extensively provided for each type of video listed above.

## Video traffic codecs

- **H.261**- Codec was originally designed for transmission over [ISDN](#) lines. With the use of *video bonding*, video bit rates are multiples of 64 Kbit/s.**H.263** - Codec is used in IP-based videoconferencing as well as in [ISDN](#) networks. H.263 requires half the bandwidth to achieve the same video quality as in H.261. As a result, H.263 has largely replaced H.261. H.263 has been optimized for a large range of bit rates and not just 64K bits/s like with H.261.**H.264/MPEG-4 - Cs** currently one of the most commonly used formats and uses half or less the bit rate of [MPEG-2](#), [H.263](#), or [MPEG-4 Part 2](#).**H265** - One of several potential successors to the widely used H.264 and based on extension of same concepts. It supports resolutions up to 8192x4320, including 8K UHD.

**Note:** Video, like audio, is carried in Realtime Protocol (RTP)

## QoS mechanisms for Video

In principle the QoS mechanisms employed to deliver the SLAs for a video transport network are *mostly* the same as those for audio. There are some differences however, mostly due to the bursty nature of video and VBR transmission.

There are two approaches to QoS, namely *Interated Services(intserv)* and *differentiated Services(diffserv)*.

Think of Intserv as operating at *signaling level* and diffserv at media-level. In other words, the intserv model ensures quality by operating at control plane; diffserv aims to ensure quality by

operating at data plane level.

In IntServ architecture network devices make requests for static bandwidth reservations and maintain the state of all reserved flows while performing classification, marking and queuing services these flows; the IntServ architecture operates-and integrates-both the control plane and the data plane, and as such has been largely abandoned due to inherent scaling limitations. The protocol used to make the bandwidth reservations is RSVP (Resource reSerVation Protocol).

There is also IntServ/DiffServ Model, which is sort of a mix. This model separates control plane operations from data plane operations. RSVP operation is limited to admission control only; with DiffServ mechanisms handling classification, marking, policing and scheduling operations. As such, the IntServ/DiffServ model is highly scalable and flexible.

**Note:** This document only focuses on diffserv (viz-a-viz prioritization scheme, LLQ) approach.

## Bandwidth guarantee

Bandwidth is obviously the most fundamental qos parameter. This depends on several parameters, most notably:

- Codec used
- Frame rate
- Image size
- Call volume (peak and average)

The old trick of throwing bandwidth at the problem is not always the solution. This is especially true for video quality. For example, with CUVA (Cisco Unified Video Advantage) there is no synchronization mechanism between the two devices (phone and PC) involved. Thus QoS should be configured to minimize jitter, latency, fragmented packets, and out-of-order packets.

**Note:** Interactive Video has the same service level requirements as VoIP because a voice call is embedded within the video stream. Streaming Video has much laxer requirements, because of the high amount of buffering that has been built into the applications.

Finally it is important to understand that unlike VoIP there are no clean formulas for calculating the required incremental bandwidth. This is because video packet sizes and packet rates vary significantly and are largely a function of the degree of motion within the video images being transmitted. More on this later.

## Queuing

Low Latency Queuing (LLQ) is the preferred queuing policy for VoIP audio. Given the stringent delay/jitter sensitive requirements of TP and the need to synchronize audio and video for CUVA, priority (LLQ) queuing is the recommended for all video traffic as well. Note that, for video, priority bandwidth is generally fudged up by 20% to account for the overhead.

## Header Compression

Not recommended for video.

## Link fragmentation and interleaving

LFI is a popular mechanism to ensure jitter doesn't get out of control on slow links, where serialization delays can be high.

But then again Interactive-Video is not recommended for slow links. This is because LLQ which the video traffic is assigned to, are not subject to fragmentation. This means that the large Interactive-Video packets (such as 1500-byte full-motion I-Frames) could cause serialization delays for smaller Interactive-Video packets.

## Congestion avoidance

Selective discard based on RTCP

## Burst

This QoS mechanism is an important for video traffic, which, as mentioned earlier, is bursty.

The optional burst parameter can be configured as part of the *priority* command[\[6\]](#).

With H.264, the worst-case burst would be the full screen of (spatially-compressed) video. Based on extensive testing on TP systems, this is found to be 64 KB. Therefore the LLQ burst parameter should be configured to permit up to 64 KB of burst per frame per screen. Thus the CTS-1000 system running at 1080p-Best (with the optional support of an auxiliary video stream[\[7\]](#)) would be configured with an LLQ with an optimal burst parameter of 128 (2x64) KB.

## How much Bandwidth?

So, how much bandwidth is required to transport a video call faithfully? Before we get down to the calculations, it is important to understand the following concepts, which are unique to video.

## Resolution

This basically refers to the size of the image. Other commonly used terms for this include *video format* and *screen size*. Commonly used video formats are shown below.

| Format | Video Resolution<br>(pixels) |
|--------|------------------------------|
| SQCIF  | 128x96                       |
| QCIF   | 176x144                      |
| SCIF   | 256x192                      |
| SIF    | 352x240                      |
| CIF    | 352x288                      |
| DCIF   | 528x384                      |
| 4CIF   | 704x576                      |
| 16CIF  | 1408x1152                    |

Vast majority of video conferencing equipment run at CIF or 4CIF formats.

Ref: [http://en.wikipedia.org/wiki/Common\\_Intermediate\\_Format](http://en.wikipedia.org/wiki/Common_Intermediate_Format)

**Note:** There is no equivalence for (video) *resolution* in the audio world

## Frame rate

This refers to the rate at which an imaging device produces unique consecutive images called [frames](#). Frame rate is expressed as frames per second (fps).

**Note:** The equivalent metric in audio world is sampling time. E.g. 8000 for g.711ulaw.

## Bandwidth calculation

Bandwidth calculations for video telephony systems and other traditional video conference systems tend to be simpler.

As an example, consider a TP call with resolution of 1080 x1920. The bandwidth required is calculated as follows-

2,073,600 pixels per frame

x3 colors per pixel

x1 Byte (8 bits) per color

x 30 frames per second

= 1.5Gbps per screen. Uncompressed!

With compression, a bandwidth of 4Mbps per screen ( > 99% compressed) is enough to transport the above frame!

The following table lists some of the combinations-

| Picture format | Luminance pixels | Luminance lines | Uncompressed bit rate (Mbit/s) |       |             |       |
|----------------|------------------|-----------------|--------------------------------|-------|-------------|-------|
|                |                  |                 | 10 frames/s                    |       | 30 frames/s |       |
|                |                  |                 | Grey                           | Color | Grey        | Color |
| SQCIF          | 128              | 96              | 1.0                            | 1.5   | 3.0         | 4.4   |
| QCIF           | 176              | 144             | 2.0                            | 3.0   | 6.1         | 9.1   |
| CIF            | 352              | 288             | 8.1                            | 12.2  | 24.3        | 36.5  |
| 4CIF           | 704              | 576             | 32.4                           | 48.7  | 97.3        | 146.0 |
| 16CIF          | 1408             | 1152            | 129.8                          | 194.6 | 389.3       | 583.9 |

Note that above calculations are for a single screen. A TP call could involve multiple screens and so, total bandwidth for the call would be a multiple of the per-screen bandwidth.

Refer to <https://supportforums.cisco.com/thread/311604> for a good bandwidth calculator for Cisco

TP systems.

## Classifying/marketing video traffic

How is video traffic identified/distinguished? One way to classify packets on CUBE is using DSCP markings.

The following table illustrates DSCP markings per Cisco QoS baseline as well as RFC 4594.

| Traffic                 | Layer 3<br>PHB | Layer 3<br>DSCP |
|-------------------------|----------------|-----------------|
| Call Signaling          | CS3            | 24              |
| Voice                   | EF             | 46              |
| Video<br>conference     | AF41           | 34              |
| TelePresence            | CS4            | 32              |
| Multimedia<br>Streaming | AF31           | 26              |
| Broadcast<br>video      | CS5            | 40              |

PHB - Per Hop Behavior. Refers to what the router does as far as packet classification and traffic conditioning functions, such as metering, marking, shaping, and policing.

By default, prior to version 9.0 CUCM (Cisco Unified Call Manager) marked any and all video traffic (including TelePresence) to AF41. Starting from version 9.0, CUCM preconfigures the following DSCP values:

- TelePresence (immersive video) calls at CS4 and
- Video (IP video telephony) calls at AF41

## Configuration

Configuring to tune for audio quality entails calculating priority bandwidth and implementing LLQ policy on a WAN link. This is generally based on the anticipated call volume and audio codec used.

While the principles are same, video bandwidth through a CUBE is not so easily calculable. This is due to a number of factors, including:

- How does one calculate the total bandwidth required given different TP calls (flowing through the CUBE) might involve different number of screens and of different resolutions?
- The bursty nature and VBR
- Another dimension of complexity [in bandwidth computation] has to do with “interoperability calls”? Interoperability calls use TIP. TIP stands for Telepresence Interoperability Protocol. TIP is used to multiplex multiple screens, multiple audio streams, as well as an auxiliary-data screen into two RTP flows, one each for video and audio. It enables point-to-point and multipoint sessions as well as a mix of multi-screen and single-screen endpoints. TIP is a Cisco proprietary protocol. TIP is based on RTCP.

Therefore, the bandwidth provisioning for video systems sometimes happens in the reverse order-

i.e. amount of bandwidth that a transport network can deliver, with LLQ policy, is determined first and based on that, the endpoint is configured. Endpoint video systems are smart enough to adjust the various video parameters for the pipe size! Accordingly, the endpoints signal the call.

## CUBE bandwidth handling

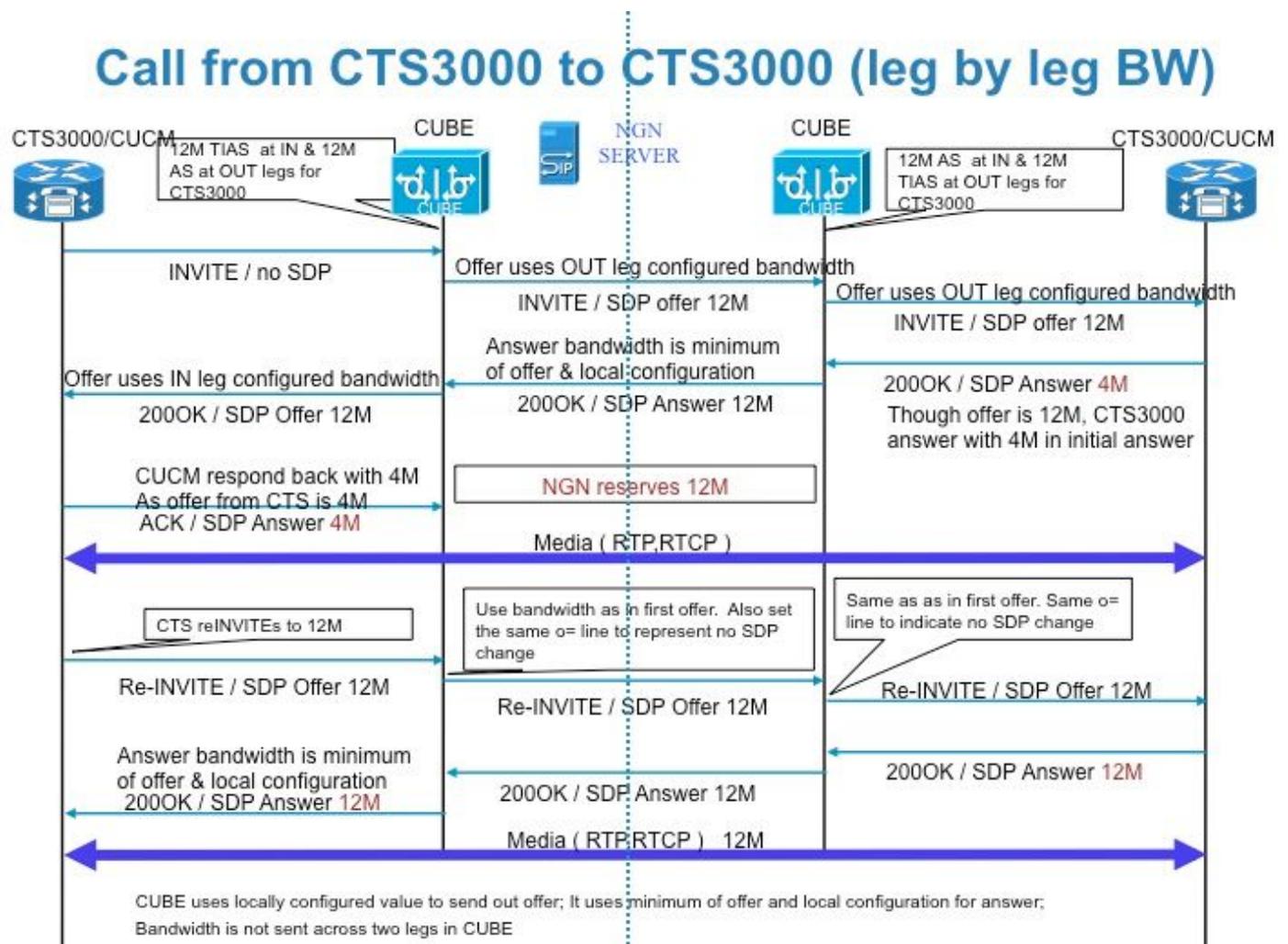
So, how does CUBE handle Bandwidth in its (SIP) offer/answers when signaling video calls? CUBE populates the video bandwidth fields in SDP as follows-

1. From *bandwidth attribute in the incoming SDP*. In SDP, there exists a bandwidth attribute, which has a modifier used to specify what type of bit-rate the value refers to. The attribute has the following form: `b=<modifier>:<value>`

2. From *video bandwidth configured on the CUBE*. For example, the estimated maximum bandwidth is calculated based on the features used by CTS user and the estimated bandwidth is pre-configured on CUBE, using the CLI-

- `<bandwidth video tias-modifier>` or
  - `<bandwidth video as-modifier>`
3. Default video bandwidth (384 Kbps)

The call flow shown below illustrates how CUBE populates bandwidth in call signaling messages-



Specifically, the CUBE uses the following logic:

- On offers (to DO calls), CUBE uses the configured bandwidth.
- On (answers to EOs), CUBE sends bandwidth whose value is the minimum of offer & local configuration.

At the SDP session level, the TIAS value is the maximal amount of bandwidth needed when all declared media streams are used[8].

## Video codec Payload types

This is another area in which video differs from audio. Audio codecs use static payload types. Video codecs, in contrast, use dynamic RTP payload types, which use the range 96 to 127.

The reason for the use of dynamic payload type has to do with the wide applicability of video codecs. Video codecs have parameters that provide a receiver with the properties of the stream that will be sent. Video payload types are defined in SDP, using the `a=rtpmap` parameter. Additionally, the `"a=fmtp:"` attribute MAY be used to specify format parameters. The `fmtp` string is opaque and is just passed to the other side.

Here is an example-

```
m=video 2338 RTP/AVP 97 98 99 100
c=IN IP4 192.168.90.237
b=TIAS:768000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42800d;max-mbps=40500;max-fs=1344;max-smbps=40500
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42800d;max-mbps=40500;max-fs=1344;max-smbps=40500;packetization-mode=1
a=rtpmap:99 H263-1998/90000
a=fmtp:99 custom=1024,768,4;custom=1024,576,4;custom=800,600,4;cif4=2;custom=720,480,2;custom=640,480,2;custom=512,288,1;cif=1;custom=352,240,1;qcif=1;maxbr=7680
a=rtpmap:100 H263/90000
a=fmtp:100 cif=1;qcif=1;maxbr=7680
```

Note that the two endpoints involved in a call might use different payload type for the same codec. CUBE responds to each side with `a=rtpmap` line received on the other leg. This means that the config "asymmetric payload full" is needed for video calls to work.

## L2 bandwidth

Unlike voice, real-time IP video traffic in general is a somewhat bursty, variable bit rate stream. Therefore video, unlike voice, does not have clear formulas for calculating network overhead because video packet sizes and rates vary proportionally to the degree of motion within the video image itself. From a network administrator's point of view, bandwidth is always provisioned at Layer 2, but the variability in the packet sizes and the variety of Layer 2 media that the packets may traverse from end-to-end make it difficult to calculate the real bandwidth that should be provisioned at Layer 2. However, the conservative rule that has been thoroughly tested and widely used is to over-provision video bandwidth by 20%. This accommodates the 10% burst and the network overhead from Layer 2 to Layer 4.

## Monitoring/measuring

As mentioned earlier video endpoints do not report a MOS as such. However the following tools could be used to measure/monitor the transport network performance, and to monitor video quality.

## IP SLAs Video

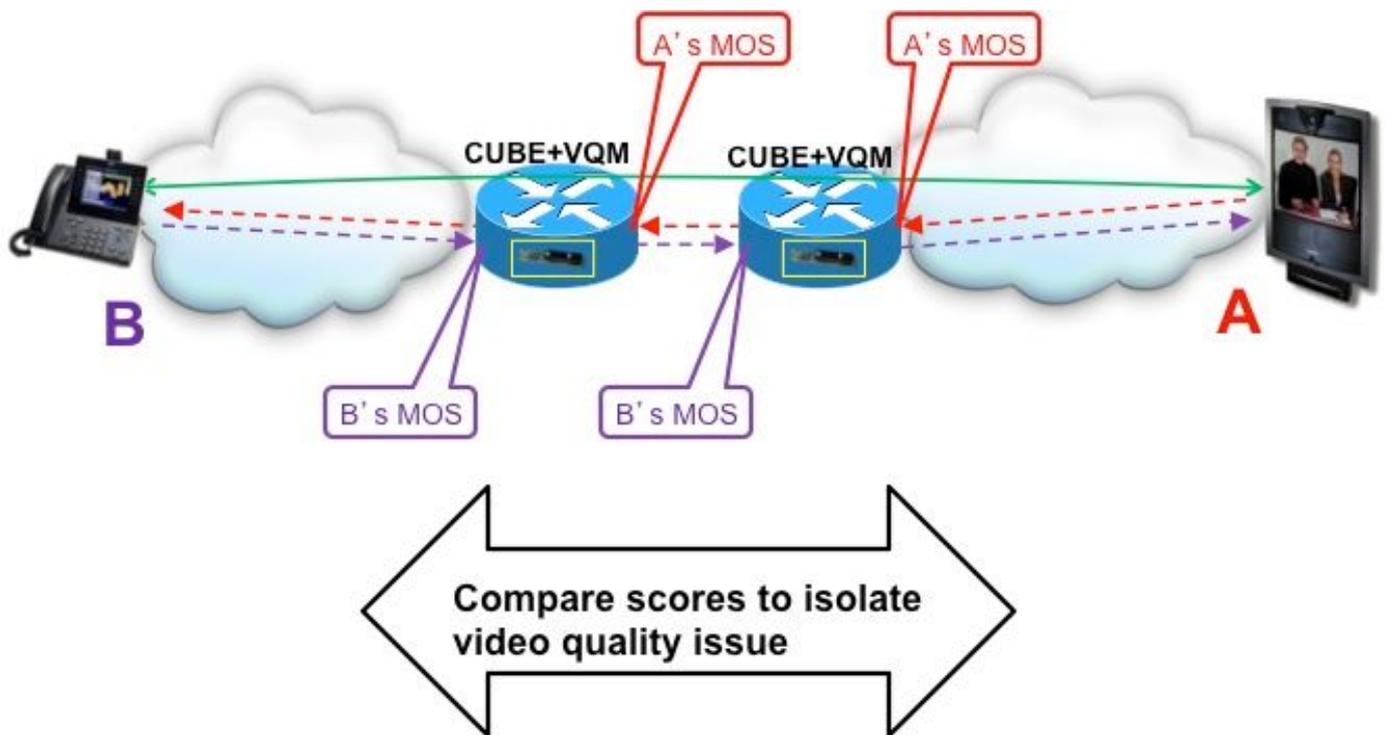
A feature embedded in IOS, IP SLAs (Service Level Agreements) performs the active monitoring of the network performance. IP SLAs video operation differs from other IP SLA operations in that all traffic is one way only, with a responder required to process the sequence numbers and time stamps locally and to wait for a request from the source before sending the calculated data back.

The source sends a request to the responder when the current video operation is done. This request signals the responder that no more packets will arrive, and that the video sink function in the video operation can be turned off. When the response from the responder arrives at the source, the statistics are read from the message, and the relevant fields in the operation are updated.

CiscoWorks IPM (IOS Performance Monitor) uses IP SLA probe and MediaTrace<sup>[9]</sup> to measure user traffic performance and reports.

## CUBE VQM

The VQM (Video Quality Monitor) feature, available on the CUBE, is a great tool to monitor video quality between two points of interest. Results are presented as MOS.



This is available from IOS 15.2(1)T and above. Note that VQM uses DSP resources.

## Reference

- [http://www.cisco.com/en/US/partner/tech/tk1077/technologies\\_configuration\\_example09186a00807ca099.shtml](http://www.cisco.com/en/US/partner/tech/tk1077/technologies_configuration_example09186a00807ca099.shtml)
- <http://www.cisco.com/en/US/partner/docs/video/milticomm/h320Bonding.html>
- [http://www.cisco.com/en/US/docs/solutions/TelePresence\\_Network\\_Systems\\_1.1\\_DG.pdf](http://www.cisco.com/en/US/docs/solutions/TelePresence_Network_Systems_1.1_DG.pdf)
- [http://www.cisco.com/en/US/docs/voice\\_ip\\_comm/uc\\_system/design/guides/videodg/gos.html](http://www.cisco.com/en/US/docs/voice_ip_comm/uc_system/design/guides/videodg/gos.html)
- [http://www.cisco.com/en/US/docs/voice\\_ip\\_comm/uc\\_system/design/guides/videodg/basics.html](http://www.cisco.com/en/US/docs/voice_ip_comm/uc_system/design/guides/videodg/basics.html)
- IP video-telephony SRND -

- [http://www.cisco.com/en/US/tech/tk1077/technologies\\_configuration\\_example09186a0080111c1b.shtml](http://www.cisco.com/en/US/tech/tk1077/technologies_configuration_example09186a0080111c1b.shtml)
- [http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/TP\\_InterOp\\_v2.html](http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/TP_InterOp_v2.html)
- [http://www.encoding.com/do\\_you\\_have\\_any\\_information\\_on\\_h.264\\_levels](http://www.encoding.com/do_you_have_any_information_on_h.264_levels)
- [http://www.cisco.com/en/US/partner/tech/tk1077/technologies\\_configuration\\_example09186a0080111c1b.shtml](http://www.cisco.com/en/US/partner/tech/tk1077/technologies_configuration_example09186a0080111c1b.shtml)
- <http://www.watchpointvideo.com/pdf/Measuring%20Video%20Quality%20in%20Videoconferencing%20Systems.pdf>
- [http://www.broadcastpapers.com/whitepapers/Why\\_IPTV\\_is\\_different\\_from\\_IP\\_data\\_and\\_Vol\\_P.pdf?CFID=25762102&CFTOKEN=60dc627518f1a19b-3F4F563D-FC97-8A61-6169D8F641750255](http://www.broadcastpapers.com/whitepapers/Why_IPTV_is_different_from_IP_data_and_Vol_P.pdf?CFID=25762102&CFTOKEN=60dc627518f1a19b-3F4F563D-FC97-8A61-6169D8F641750255)
- [http://en.wikipedia.org/wiki/Video\\_compression\\_picture\\_types](http://en.wikipedia.org/wiki/Video_compression_picture_types)
- [http://inst.eecs.berkeley.edu/~ee290t/sp04/lectures/coding\\_standards.pdf](http://inst.eecs.berkeley.edu/~ee290t/sp04/lectures/coding_standards.pdf)
- <http://www.cs.jhu.edu/~yairamir/cs667/Multimedia/compress.gif>
- <http://www.wireshark.org/lists/wireshark-users/201003/msg00125.html>
- <http://www.networkworld.com/news/tech/2002/0923tech.html>
- <http://www.javvin.com/protocolH263.html>
- [http://www.cisco.com/en/US/docs/ios-xml/ios/ipsla/configuration/12-2se/sla\\_video.html#GUID-29B155B6-AFC3-4F8B-AC1D-C127C9D797F0](http://www.cisco.com/en/US/docs/ios-xml/ios/ipsla/configuration/12-2se/sla_video.html#GUID-29B155B6-AFC3-4F8B-AC1D-C127C9D797F0)

[1] Based on highest audio human-audible frequency of approx. 4000Hz. Ref: Nyquist theorem.

[2] Constant Bit Rate (CBR) transmission schemes are *possible* with video, but they trade off quality to maintain CBR.

[3] For Inter-frame compressions

[4] Note that SLA is more stringent for TP.

[5] Life-size images and high quality audio

[6] The default value for this parameter is 200ms of traffic at priority bandwidth. The Cisco LLQ algorithm has been implemented to include a default burst parameter equivalent to 200 ms worth of traffic. Testing has shown that this burst parameter does not require additional tuning for a single IP Videoconferencing (IP/VC) stream. For multiple streams, this burst parameter may be increased as required.

[7] An auxiliary video stream is a 5 fps video channel for sharing presentations or other collateral vial the data projector.

[8] Note that some systems use the "AS" (Application Specific) modifier to convey maximum bandwidth. Interpretation of this attribute is dependent on the application's notion of maximum bandwidth.

CUBE is agnostic as to the specific bandwidth modifier (TIAS or AS).

[9] Mediatrace is an IOS Software feature that discovers the routers and switches along the path of an IP flow.

StartSelection:0000000199 EndSelection:0000000538