CHAPTER 2

Medianet Bandwidth and Scalability

This chapter discusses the bandwidth requirements for different types of video on the network, as well as scalability techniques that allow additional capacity to be added to the network.

Bandwidth Requirements

Video is the ultimate communications tool. People naturally use visual clues to help interpret the spoken word. Facial expression, hand gestures, and other clues form a large portion of the messaging that is normal conversation. This information is lost on traditional voice-only networks. If enough of this visual information can be effectively transported across the network, potential productivity gains can be realized. However, if the video is restricted by bandwidth constraints, much of the visual information is lost. In the case of video conferencing, the user community does not significantly reduce travel. In the case of video surveillance, small distinguishing features may be lost. Digital media systems do not produce engaging content that draws in viewers. In each case, the objectives that motivated the video deployment cannot be met if the video is restricted by bandwidth limitations.

Quantifying the amount of bandwidth that a video stream consumes is a bit more difficult than other applications. Specifying an attribute in terms of bits per second is not sufficient. The per-second requirements result from other more stringent requirements. To fully understand the bandwidth requirements, the packet distribution must be fully understood. This is covered in Chapter 1, "Medianet Architecture Overview,"and briefly revisited here.

The following video attributes affect how much bandwidth is consumed:

- Resolution—The number of rows and columns in a given frame of video in terms of pixel count. Often resolution is specified as the number of rows. Row counts of 720 or greater are generally accepted as high definition (HD) video. The number of columns can be derived from the number of rows by using the aspect ratio of the video. Most often HD uses an aspect ratio of 16:9, meaning 16 columns for every 9 rows. As an example, a resolution of 720 and an aspect ratio of 16:9 gives a screen dimension of 1280 x 720 pixels. The same 720 resolution at a common 4:3 aspect ratio gives a screen dimension of 960 x 720 pixels. Resolution has a significant effect on bandwidth requirements as well as the productivity gains of video on the network. Resolution is a second-degree term when considering network load. If the aspect ratio is held at 16:9 and the resolution is increased from 720 to 1080, the number of pixels per frame jumps from 921,600 to 2,073,600, which is significant. If the change is in terms of percent, a 50 percent increase in resolution results in a 125 percent increase in pixel count. Resolution is also a key factor influencing the microburst characteristics of video. A microburst results when an encoded frame of video is sliced into packets and placed on the outbound queue of the encoder network interface card (NIC). This is discussed in more detail later in this chapter.

- Encoding implementation—Encoding is the process of taking the visual image and representing it in terms of bytes. Encoders can be distinguished by how well they compress the information. Two factors are at work. One is the algorithm that is used. Popular encoding algorithms are H.264 and MPEG-4. Other older encoders may use H.263 or MPEG-2. The second factor is how well these algorithms are implemented. Multiple hardware digital signal processors (DSPs) are generally able to encode the same video in less bytes than a battery operated camera using a low power CPU. For example, a flip camera uses approximately 8 Mbps to encode H.264 at a 720 resolution. A Cisco TelePresence CTS-1000 can encode the same resolution at 2 Mbps. The algorithm provides the encoder flexibility to determine how much effort is used to optimize the compression. This in turn gives vendors some latitude when trying to meet other considerations, such as cost and power.
- Quality—Encoding video uses a poor compression. This means that some amount of negligible visual information can be discarded without having a detrimental impact of the viewer experience. Examples are small variations in color at the outer edges of the visual spectrum of red and violet. As more detail is omitted from the encoded data, small defects in the rendered video begin to become apparent. The first noticeable impact is color banding. This is when small color differences are noticed in an area of common color. This is often most pronounced at the edge of the visible spectrum, such as a blue sky.
- Frame rate—This is the number of frames per second (fps) used to capture the motion. The higher the frame rate, the smoother and more life-like is the resulting video. At frame rates less than 5 fps, the motion becomes noticeably jittery. Typically, 30 fps is used, although motion pictures are shot at 24 fps. Video sent at more than 30 fps offers no substantial gain in realism. Frame rates have a linear impact on bandwidth. A video stream of 15 fps generates approximately half as much network traffic as a stream of 30 fps.
- Picture complexity—Encoders must take a picture and encode it in as few bytes as possible without noticeably impacting the quality. As the image becomes more complex, it takes more bytes to describe the scene. Video of a blank wall does not consume as much bandwidth as a scene with a complex image, such as a large room of people. The impact on bandwidth is not substantial but does have some influence.
- Motion—Just like picture complexity, the amount of motion in a video has some influence over how much bandwidth is required. The exception is Motion JPEG (M-JPEG). The reason is that all other encoding techniques involve temporal compression, which capitalizes on savings that can be made by sending only the changes from one frame to the next. As a result, video with little motion compresses better than video with a great deal of motion. Usually, this means that video shot outside, where a breeze may be moving grass or leaves, often requires more network bandwidth than video shot indoors. Temporal compression is discussed in more detail in Chapter 1, "Medianet Architecture Overview."

It is possible to have some influence on the bandwidth requirements by changing the attributes of the video stream. A 320x240 video at 5 fps shot in a dark closet requires less bandwidth than a 1080x1920 video at 30 fps shot outside on a sunny, breezy day. The attributes that have the most influence on network bandwidth are often fully configurable. These are resolution, frame rate, and quality settings. The remaining attributes are not directly controlled by the administrator.

Measuring Bandwidth

Network bandwidth is often measured in terms of bits per second. This is adequate for capacity planning. If a video stream is expected to run at 4 megabits per second (Mbps), a 45 Mbps circuit can theoretically carry 11 of these video streams. The number is actually less, because of the sub-second bandwidth requirements. This is referred to as the *microburst* requirements, which are always greater than the one second smoothed average.

Consider the packet distribution of video. First remember that frames are periodic around the frame rate. At 30 fps, there is a frame every 33 msec. The size of this frame can vary. Video is composed of two basic frame types, I-frames and P-frames. I-frames are also referred to as full reference frames. They are larger than P-frame but occur much less frequently. It is not uncommon to see 128 P-frames for every 1 I-frame. Some teleconference solutions send out even fewer I-frames. When they are sent, they look like a small burst on the network when compared to the adjacent P-frames. It may take as many as 80 packets or more to carry an I-frame of high definition 1080 video. These 80+ packets show up on the outbound NIC of the encoder in one chunk. The NIC begins to serialize the packet onto the Ethernet wire. During this time, the network media is being essentially used at 100 percent; the traffic bursts to line rate for the duration necessary to serialize an I-frame. If the interface is a Gigabit interface, the duration of this burst is one-tenth as long as the same burst on a 100 Mbs interface. A microburst entails the concept that the NIC is 100 percent used during the time it takes to serialize all the packets that compose the entire frame. The more packets, the longer duration required to serialize them.

It is best to conceive of a microburst as either the serialization delay of the I-frame or the total size of the frame. It is not very useful to characterize an I-frame in terms of a rate such as Kbps, although this is fairly common. On closer examination, all bursts, and all packets, are sent at line rate. Interfaces operate only at a single speed. The practice of averaging all bits sent over a one-second interval is somewhat arbitrary. At issue is the network ability to buffer packets, because multiple inbound streams are in contention for the same outbound interface. A one-second measurement interval is too long to describe bandwidth requirements because very few devices can buffer one second worth of line rate data. A better interval is 33 msec, because this is the common frame rate.

There are two ways to consider this time interval. First, the serialization delay of any frame should be less than 33 msec. Second, any interface in the network should be able to buffer the difference in serialization delay between the ingress and egress interface over a 33-msec window. During congestion, the effective outbound serialization delay for a given stream may fall to zero. In this case, the interface may have to queue the entire frame. If queue delays of above 33 msec are being experienced, the video packets are likely to arrive late. Network shapers and policers are typical points of concern when talking about transporting I-frames. These are discussed in more detail in Chapter 4, "Medianet QoS Design Considerations,"and highlighted later in this chapter.

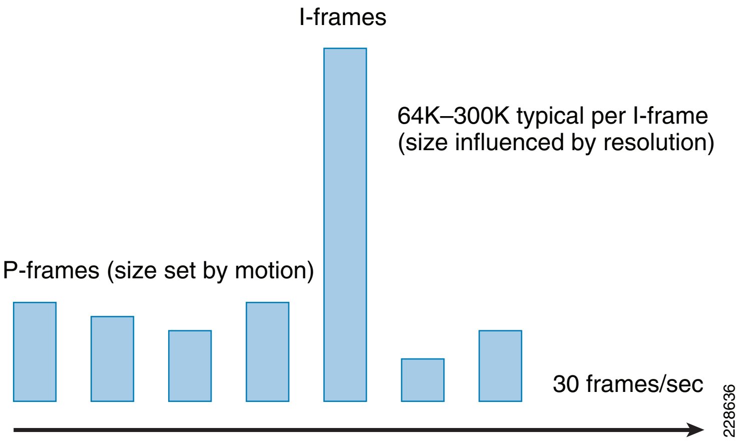
Video Transports

Several classifications can be used to describe video, from real-time interactive streaming video on the high end to prerecorded video on the low end. Real-time video is being viewed and responded to as it is occurring. This type of stream has the highest network requirements. Remote medicine is an example of an application that uses this type of video. TelePresence is a more common application. In all cases, packets that are dropped by the network cannot be re-sent because of the time-sensitive nature. Real-time decoders are built with the smallest de-jitter buffers possible. On the other extreme is rebroadcasting previously recorded video. This is usually done over TCP and results in a network load similar to large FTP file transfers. Dropped packets are easily retransmitted. Chapter 4, "Medianet QoS Design Considerations" expands on this concept and discusses the various types of video and the service levels required of the network.

Packet Flow Malleability

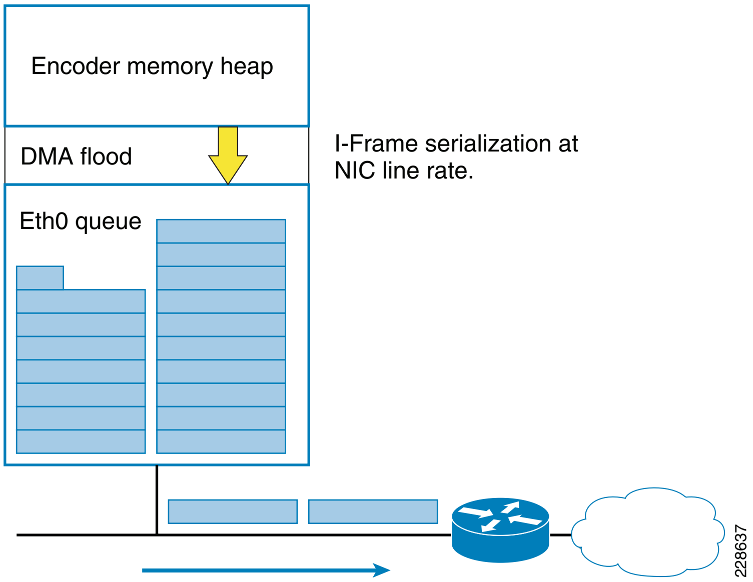
Video packets are constrained by the frame rate. Each frame consists of multiple packets, which should arrive within the same frame window. There are I-frames and P-frames. The network is not aware of what type of frame has been sent, or that a group of packets are traveling together as a frame. The network considers each packet only as a member of a flow, without regard to packet distribution. When tools such as policers and shapers are deployed, some care is required to accommodate the grouping of packets into frames, and the frame rate. The primary concern is the I-frame, because it can be many times larger than a P-frame, because of the way video encoders typically place I-frames onto the network. (See Figure 2-1.)

Figure 2-1 P-frames and I-frames



When an I-frame is generated, the entire frame is handed to the network abstraction layer (NAL). This layer breaks the frame into packets and sends them on to the IP stack for headers. The processor on the encoder can slice the frame into packets much faster than the Ethernet interface can serialize packets onto the wire. As a result, video frames generate a large number of packets that are transmitted back-to-back with only the minimum interpacket gap (IPG). (See Figure 2-2.)

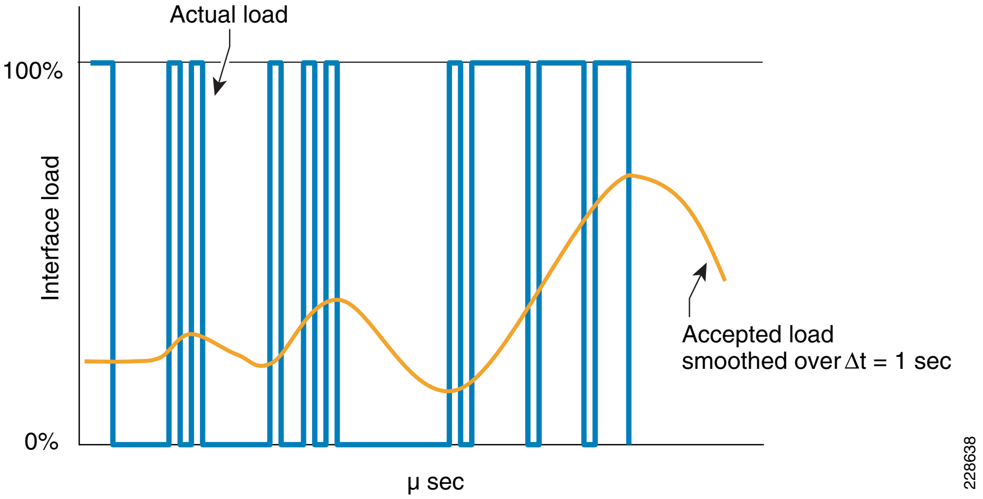
Figure 2-2 I-frame Serialization



The service provider transport and video bandwidth requirements set the limit to which video streams can be shaped and recast. Natural network video is sent at a variable bit rate. However, many transports have little tolerance for traffic flows that exceed a predetermined contract committed information rate (CIR). Although Chapter 4, "Medianet QoS Design Considerations" discusses this in further details, some overview of shapers and policers is warranted as part of the discussion of bandwidth requirements.

Any interface can transmit only at line rate. Interfaces use a line encoding scheme to ensure that both the receiver and transmitter are bit synchronized. When a user states that an interface is running at *x* Mbps, that is an average rate over 1 second of time. The interface was actually running at 100 percent utilization while those packets were being transmitted, and idle at all the other times. Figure 2-3 illustrates this concept:

Figure 2-3 Interface Load/Actual Load



Microbursts

In video, frames are sent as a group of packets. These packets are packed tightly because they are generated at the same time. The larger the frame, the longer the duration of the microburst that results when the frame is serialized. It is not uncommon to find microbursts measured in terms of bits per second. Typically the rate is normalized over a frame. For example, if an I-frame is 80 Kb, and must be sent within a 33 msec window, it is tempting to say the interface is running at 4 Mbps but bursting to (80x1000x8)/0.033 = 19.3 Mbps. In actuality, the interface is running at line rate long enough to serialize the entire frame. The interface speed and buffers are important in determining whether there will be drops. The normalized 33 msec rate gives some useful information when setting shapers. If the line rate in the example above is 100 Mbps, you know that the interface was idle for 80.7 percent of the time during the 33 msec window. Shapers can help distribute idle time. However, this does not tell us whether the packets were evenly distributed over the 33 msec window, or whether they arrived in sequence during the first 6.2 msec. The encoders used by TelePresence do some level of self shaping so that packets are better distributed over a 33 msec window, while the encoders used by the IP video surveillance cameras do not.

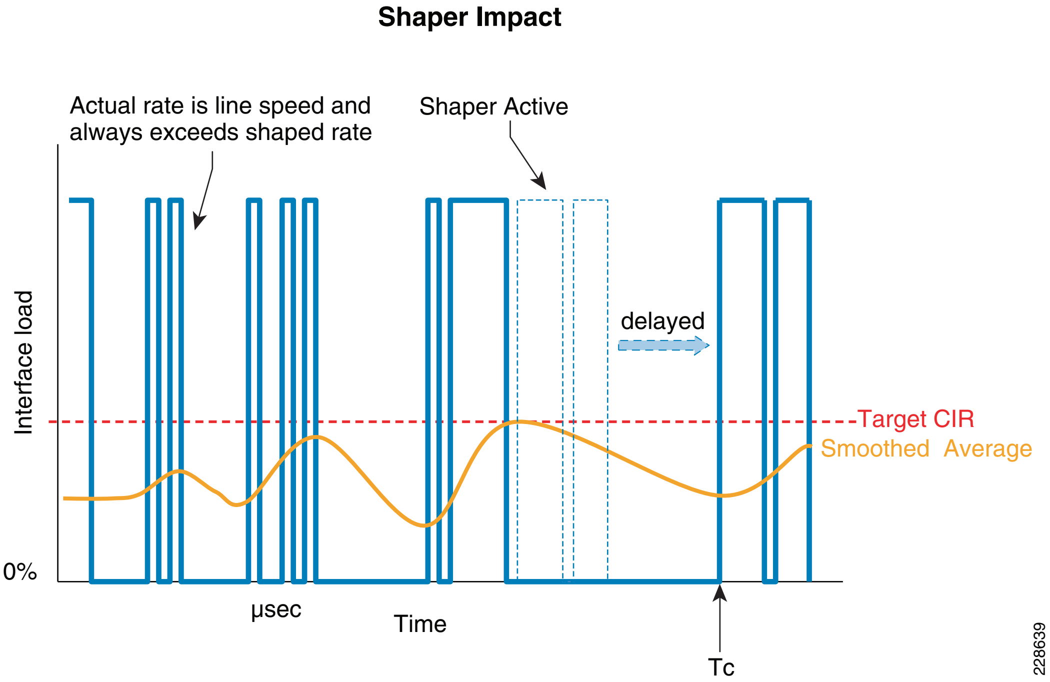
Shapers

Shapers are the most common tool used to try to mitigate the effect of bursty traffic. Their operation should be well understood so that other problems are not introduced. Shapers work by introducing delay. Ideally, the idle time is distributed between each packet. Only hardware-based shapers such as those found in the Cisco Catalyst 3750 Metro device can do this. Cisco IOS shapers use a software algorithm to enforce packets to delay. Cisco IOS-based shapers follow the formula *Bc = CIR * Tc*. The target bandwidth (*CIR*) is divided into fixed time slices (*Tc*). Each Tc can send only *Bc* bytes worth of data. Additional traffic must wait for the next available time slice. This algorithm is generally effective, but keep in mind some details. First, IOS shapers can meter only traffic with time slices of at least 4 msec. This means that idle time cannot be evenly distributed between all packets. Within a time slice, the interface still sends packets at line rate. If the queue of packets waiting is deeper than *Bc* bytes, all the packets are sent in sequence at the start of each Tc, followed by an idle period. In effect, if the offered rate exceeds the CIR rate for an extended period, the shaper introduces microbursts that are limited to *Bc* in size. Each time slice is independent of the previous time slice. A burst of packets may arrive at the shaper and completely fill a Bc at the very last moment, followed immediately by a new time slice with another *Bc* worth of available bandwidth. This means that although the interface routinely runs at line rate for each *Bc* worth of data, it is possible that it will run at line rate for *2*Bc* worth of bytes. When a shaper first becomes active, the traffic alignment in the previous Tc is not considered.

Partial packets are another feature of shapers to consider. Partial packets occur when a packet arrives whose length exceeds the remaining Bc bits available in the current time slice. There are two possible approaches to handle this. First, delay the packet until there are enough bits available in the bucket. The down side of this approach is twofold. First, the interface is not able to achieve CIR rate because time slices are expiring with bits still left in the Bc bucket. Secondly, while there may not be enough Bc bits for a large packet, there could be enough bits for a much smaller packet in queue behind the large packet. There are problems with trying to search the queue looking for the best use of the remaining Bc bits. Instead, the router allows the packet to transmit by borrowing some bits from the next time slice.

Figure 2-4 shows the impact of using shapers.

Figure 2-4 Shaper Impact



Choosing the correct values for Tc, Bc, and CIR requires some knowledge of the traffic patterns. The CIR must be above the sustained rate of the traffic load; otherwise, traffic continues to be delayed until shaper drops occur. In addition, the shaper should delay as few packets as possible. Finally, if the desire is to meet a service level enforced by a policer, the shaper should not send bursts (*Bc*) larger than the policer allows. The attributes of the upstream policer are often unknown, yet these values are a dominant consideration when configuring the shaper. It might be tempting to set the shaper Bc to its smallest possible value. However, as Tc falls below 2 * 33 msec, the probability of delaying packets increases, as does the jitter. Jitter is at its worst when only one or two packets are delayed by a large Tc. As Tc approaches 0, jitter is reduced and delay is increased. In the limit as Tc approaches 0, the introduced delay equals the serialization delay if the circuit can be clocked at a rate equal to CIR. With TelePresence, the shaper Tc should be 20 msec or less to get the best balance between delay and jitter. If the service provider cannot accept bursts, the shaper can be set as low as 4 msec.

With shapers, if packets continue to arrive at a rate that exceeds the CIR of the shaper, the queue depth continues to grow and eventually saturates. At this point, the shaper begins to discard packets. Normally, a theoretically ideal shaper has infinite queue memory and does not discard packets. In practice, it is actually desirable to have shapers begin to look like policers if the rate exceeds CIR for a continued duration. The result of drops is that the sender throttles back its transmission rate. In the case of TCP flows, window sizes are reduced. In the case of UDP, lost transmissions cause upper layers such as TFTP, LDAP, or DNS to pause for the duration of a response timeout. UDP flows in which the session layer has no feedback mechanism can overdrive a shaper. Denial-of-service (DoS) attacks are in this class. Some Real-Time Protocol (RTP)/UDP video may also fall in this class where Real-Time Control Protocol (RTCP) is not used. Real-Time Streaming Protocol (RTSP)-managed RTP flows are an example of this type of video. In these cases, it is very important to ensure that the shaper CIR is adequately configured. When a shaper queue saturates, all non-priority queuing (PQ) traffic can be negatively impacted.

Shapers versus Policers

Policers and shapers are related methods that are implemented somewhat differently. Typically, a shaper is configured on customer equipment to ensure that traffic is not sent out of contract. The service provider uses a policer to enforce a contracted rate. The net effect is that shapers are often used to prevent upstream policers from dropping packets. Typically, the policer is set in place without regard to customer shapers. If the customer knows what the parameters of the policer are, this knowledge can be used to correctly configure a shaper.

Understanding the difference between policers and shapers helps in understanding the difference in implementation. First, a policer does not queue any packets. Any packets that do not conform are dropped. The shaper is the opposite. No packets are dropped until all queue memory is starved. Policers do not require the router to perform an action; instead, the router only reacts. Shaping is an active process. Queues must be managed. Events are triggered based on the fixed Tc timer. The algorithm for shaping is to maintain a token bucket. Each *Tc* seconds, Bc tokens are added to the bucket. When a packet arrives, the bucket is checked for available tokens. If there are enough tokens, the packet is allowed onto the TxRing and the token bucket is debited by the size of the packet. If the bucket does not have enough tokens, the packet must wait in queue. At each *Tc* interval, Bc tokens are credited to the bucket. If there are packets waiting in queue, these packets can be processed until either the queue is empty or the bucket is again depleted of tokens.

By contrast, policing is a passive process. There is no time constant and no queue to manage. A simple decision is made to pass or drop a packet. With policing, the token bucket initially starts full with Bc tokens. When a packet arrives, the time interval since the last packet is calculated. The time elapsed is multiplied by the CIR to determine how many tokens should be added to the bucket. After these tokens have been credited, the size of the packet is compared with the token balance in the bucket. If there are available tokens, the packet is placed on the TxRing and the size of the packet is subtracted from the token bucket. If the bucket does not have enough available tokens, the packet is dropped. As the policed rate approaches the interface line rate, the size of the bucket become less important. When *CIR = Line Rate*, the bucket refills at the same rate that it drains.

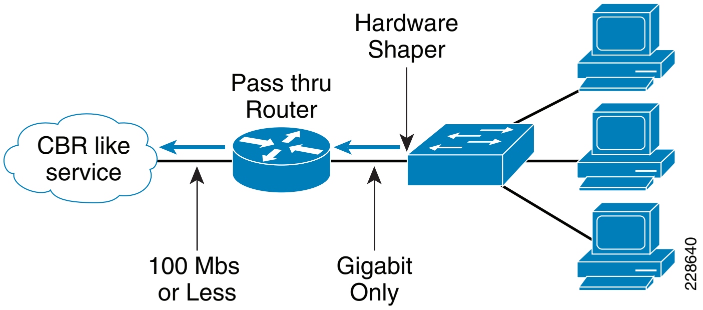
Because tokens are added based on packet arrival times, and not as periodic events as is done with shapers, there is no time constant (Tc) when discussing policers. The closest equivalent is the time required for an empty bucket to completely refill if no additional packets arrive. In an ideal case, a shaper sends Bc bytes at line rate, which completely drains the policer Bc bucket. The enforced idle time of the shaper for the remaining Tc time then allows the Bc bucket of the policer to completely refill. The enforced idle time of the shaper is *Tc*(1-CIR/Line_Rate)*. In practice, it is best to set the shaper so that the policer Bc bucket does not go below half full. This is done by ensuring that when the shaped CIR equals the policed CIR, the shaper Bc should be half of the policer Bc.

It is not always possible to set the shaper Bc bucket to be smaller than the policer Bc bucket, because shapers implemented in software have a minimum configurable Tc value of 4 msec. The shaper Tc is not directly configured; instead, Bc and CIR are configured and Tc is derived from the equation *Tc = Bc/CIR*. This means that the shaper Bc cannot be set to less than *0.004*CIR*. If the policer does not allow bursts of this size, some adjustments must be made. Possible workarounds are as follows:

- Place a hardware-based shaper inline (see Figure 2-5).

  Examples of devices that support hardware based shaping are the Cisco Catalyst 3750 Metro Series Switches. However, the Cisco Catalyst 3750 Metro supports hardware shaping only on 1 Gigabit uplink interfaces. These interfaces do not support any speed other than 1 Gigabit. This can be a problem if the service provider is not using a 1 Gigabit interface to hand off the service. In this case, if the Cisco Catalyst 3750 Metro is to be used, the hardware shaping must occur before the customer edge (CE) router. The Cisco Catalyst 3750 Metro would attach to a router instead of directly with the service provider. The router would handle any Border Gateway Protocol (BGP) peering, security, encryption, and so on. The Cisco Catalyst 3750 Metro would provide wiring closet access and the shaping. This works only if the router is being fed by a single Metro device. Of course, if more than 48 ports are needed, additional switches must be fed through the Cisco Catalyst 3750 Metro such that the hardware shaper is metering all traffic being fed into the CE router.

Figure 2-5 Hardware-Based Shaper Inline



- Contract a higher CIR from the service provider.

  As the contracted CIR approaches the line rate of the handoff circuit, the policer bucket refill rate begins to approach the drain rate. The shaper does not need to inject as much idle time. When the contracted CIR equals the line rate of the handoff circuit, shaping is no longer needed because the traffic never bursts above CIR. Testing in the lab resulted in chart shown in Figure 2-6, which can be used to determine the contracted service provider CIR necessary when shaping is required but the shapers Bc cannot be set below the maximum burst allowed by the service provider. This is often a concern when the service provider is offering a constant bit rate (CBR) service. Video is generally thought of as a variable bit rate, real-time (VBR-RT) service.

Figure 2-6 Higher CIR

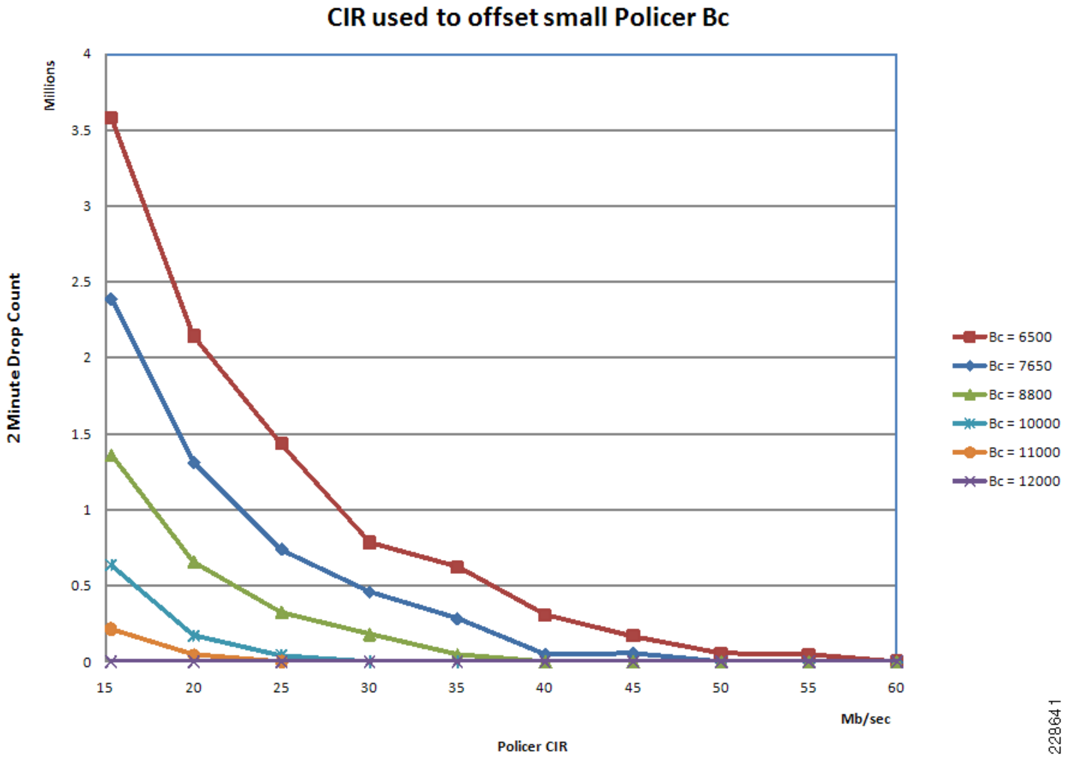


Figure 2-6 shows validation results and gives some guidance about the relationship between policers and shapers. The plots are the result of lab validation where a shaper was fixed with a CIR of 15.3 Mbps and a Bc of 7650 bytes. The plot show the resulting policer drops as the policer CIR values are changed. The Y-Axis shows the drops that were reported by the service provider policer after two minutes of traffic. The X-Axis shows the configured CIR on the policer. This would be the equivalent bandwidth purchased from the provider. Six plots are displayed, each at a unique Policer Bc. This represents how tolerant the service provider is of bursts above CIR. The objective is to minimize the drops to zero at the smallest policer CIR possible. The plot that represents a Bc of 7650 bytes is of particular interest because this is the case where the policer Bc equals the shaper Bc.

The results show that the policed CIR should be greater than twice that of the shaped CIR. Also note that at a policed Bc of 12 KB. This represents the smallest policer Bc that allows the policed CIR to equal to the shaped CIR. As a best practice, it is recommended that the policer Bc be at least twice large as the shaper Bc if the CIR is set to the same value. As this chart shows, if this best practice cannot be met, additional CIR must be purchased from the service provider.

Key points are as follows:

- Shapers do not change the speed at which packets are sent, but rather introduce idle times.
- Policers allow traffic at line rate until the Bc bucket is empty. Policers do not enforce a rate, but rather a maximum burst beyond a rate.
- Shapers that feed upstream policers should use a Bc that is half of the policer Bc.

In the case of TelePresence, the validation results plotted above can be used to derive the following recommendations:

- The shaper Tc should be 20 msec or less. At 20 msec, the number of delayed P-frames is minimized.
- The cloud should be able to handle a burst of at least two times the shaper Bc value. At 20 msec Tc and 15.3 MB CIR, this would be buffer space or an equivalent policer Bc of at least 76.5 KB.
- If the burst capabilities of the cloud are reduced, the shaper Tc must be reduced to maintain the 2:1 relationship (policer Bc twice that of the shaper Bc).
- The minimum shaper Tc is 4 msec on most platforms. If the resulting Bc is too large, additional bandwidth can be purchased from the service provider using the information in Table 2-1.

Note Table 2-1 applies to the Cisco TelePresence System 3000.

Table 2-1 CIR Guidelines

Policed Bc or interface buffer (Kbyte)		CIR (Mbit/sec)
Less than	But more than	
15	12	20
12	11	25
11	10	30
10	8.8	40
8.8	7.65	50
7.65	6.50	75

Table 2-1 CIR Guidelines (continued)

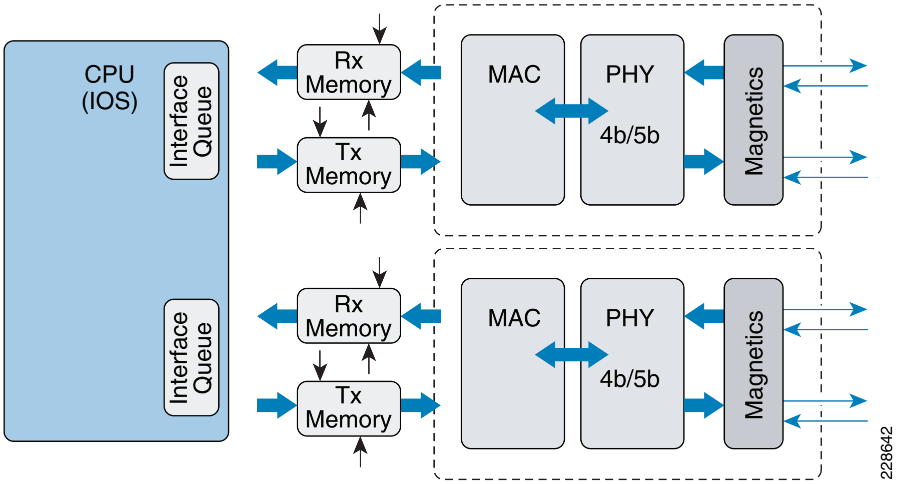
Policed Bc or interface buffer (Kbyte)		CIR (Mbit/sec)
Less than	But more than	
15	12	20
12	11	25
11	10	30
10	8.8	40
6.50	3.0	100
3.0	0.0	N/A

Because shapers can send Bc bytes at the beginning of each Tc time interval, and because shapers feed indirectly into the TxRing of the interface, it is possible to tune the TxRing to accommodate this traffic.

TxRing

The TxRing and RxRings are memory structures shared by the main processor and the interface processor (see Figure 2-7). This memory is arranged as a first in, first out (FIFO) queue. The ring can be thought of as a list of memory pointers. For each ring, there is a read pointer and a write pointer. The main processor and interface process each manage the pair of pointers appropriate to their function. The pointers move independently of one another. The difference between the write and read pointers gives the depth of the queue. Each pointer links a particle of memory. Particles are an efficient means of buffering packets of all different sizes within a pool of memory. A packet can be spread over multiple particles depending on the size of the packet. The pointers of a single packet form a linked list.

Figure 2-7 TxRings and RxRings



The rest of the discussion on Cisco IOS architecture is out of scope for this section, but some key points should be mentioned. Because a shaper can deposit Bc bytes of traffic onto an interface at the beginning of each Tc time period, the TxRing should be at least large enough to handle this traffic. The exact number of particles required depends on the average size of the packets to be sent, and the average number of particles that a packet may link across. It may not be possible to know these values in all cases.

But some worst case assumptions can be made. For example, video flows typically use larger packets of approximately 1100 bytes (average). Particles are 256 bytes. An approximate calculation for a shaper configured with a CIR of 15 Mb and a Tc of 20 msec would yield a Bc of 37.5 Kb. If that much traffic is placed on the TxRing at once, it requires 146 particles.

However, there are several reasons the TxRing should not be this large. First, a properly configured shaper is not active most of the time. QoS cannot re-sequence packets already on the TxRing. A smaller TxRing size is needed to allow QoS to properly prioritize traffic. Second, the downside of a TxRing that is too small is not compelling. In the case where the shaper is active and a Bc worth of data is being moved to the output interface, packets that do not fit onto the ring wait on the interface queue. Third, in a converged network with voice and video, the TxRing should be kept as small as possible. A ring size of 10 is adequate in a converged environment if a slow interface such as a DS-3 is involved. This provides the needed back-pressure for the interface queueing. In most other cases, the default setting provides the best balance between the competing objects.

The default interface hold queue may not be adequate for video. There are several factors such as the speed of the link, other types of traffic that may be using the link as well as the QoS service policy. In most cases, the default value is adequate, but it can be adjusted if output drops are being reported.

Converged Video

Mixing video with other traffic, including other video, is possible. Chapter 4, "Medianet QoS Design Considerations" discusses techniques to mark and service various types of video.

In general terms, video classification follows the information listed in Table 2-2.

Table 2-2 Video Classification

Application Class	Per-Hop Behavior	Media Application Example
Broadcast Video	CS5	IP video surveillance/enterprise TV
Real-Time Interactive	CS4	TelePresence
Multi-media Conferencing	AF4	Unified Personal Communicator
Multi-media Streaming	AF3	Digital media systems (VoDs)
HTTP Embedded Video	DF	Internal video sharing
Scavenger	CS1	YouTube, iTunes, Xbox Live, and so on

Queuing is not video frame-aware. Each packet is treated based solely on its markings, and the capacity of the associated queue. This means that it is possible, if not likely, that video frames can be interleaved with other types of packets. Consider a P-frame that is 20 packets deep. The encoder places those twenty packets in sequence, with the inter-packet gaps very close to the minimum 9.6 usec allowed. As these twenty packets move over congested interfaces, packets from other queues may be interleaved on the interface. This is the normal QoS function. If the video flow does not cross any interface where there is congestion, queuing is not active and the packet packing is not be disturbed.

Congestion is not determined by the one-second average of the interface. Congestion occurs any time an interface has to hold packets in queue because the TxRing is full. Interfaces with excessively long TxRings are technically less congested than the same interface with the same traffic flows, but with a smaller TxRing. As mentioned above, congestion is desirable when the objective is to place priority traffic in front of non-priority traffic. When a video frame is handled in a class-based queue structure, the result at the receiving codec is additional gaps in packet spacing. The more often this occurs, the greater the fanout of the video frame. The result is referred to as application jitter. This is slightly different than packet jitter. Consider again the P-frame of video. At 30 fps, the start of each frame is aligned on 33 msec boundaries; this means the initial packet of each frame also aligns with this timing. If all the interfaces along the path are empty, this first packet arrives at the decoding station spaced exactly 33 msec apart. The delay along the path is not important, but as the additional packets of the frame transit the interface, some TxRings may begin to fill. When this happens, the probability that a non-video packet (or video from a different flow) will be interleaved increases. The result is that even though each frame initially had zero jitter, the application cannot decode the frame until the last packet arrives.

Measuring application jitter is somewhat arbitrary because not all frames are the same size. The decoder may process a small frame fairly quickly, but then have to decode a large frame. The end result is the same, the frame decode completion time is not a consistent 33 msec. Decoders employ playout buffers to address this situation. If the decoder knows the stream is not real-time, the only limit is the tolerance of the user to the initial buffering delay. Because of this, video that is non-real-time can easily be run on a converged network. The Internet is a perfect example. Because the stream is non-real-time, the video is sent as a bulk transfer. Within HTML, this usually a progressive load. The data transfer may complete long before the video has played out. What this means is that a video that was encoded at 4 Mbps flows over the network as fast as TCP allows, and can easily exceed the encoded rate. Many players make an initial measurement of TCP throughput and then buffer enough of the video such that the transfer completes just as the playout completes. If the video is real-time, the playout buffers must be as small as possible. In the case of TelePresence, a dynamic playout buffer is implemented. The duration of any playout has a direct impact on the time delay of the video. Running real-time flows on a converged network takes planning to ensure that delay and jitter are not excessive. Individual video applications each have unique target thresholds.

As an example, assume a network with both real-time and non-real-time video running concurrently with data traffic. Real-time video is sensitive to application jitter. This type of jitter can occur any time there is congestion along that path. Congestion is defined as a *TxRing* that is full. RxRings can also saturate, but the result is more likely a drop. Traffic shapers can cause both packet jitter and application jitter. Jitter can be reduced by placing real-time video in the PQ. TxRings should be fairly small to increase the effectiveness of the PQ. The PQ should be provisioned with an adequate amount of bandwidth, as shown by Table 2-1. This is discussed in more depth in Chapter 4, "Medianet QoS Design Considerations."

Note TxRings and RxRings are memory structures found primarily in IOS-based routers.

Bandwidth Over Subscription

Traditionally, the network interfaces were oversubscribed in the voice network. The assumption is that not everyone will be on the phone at the same time. The ratio of oversubscription was often determined by the type of business and the expected call volumes as a percent of total handset. Oversubscription was possible because of Call Admission Control (CAC), which was an approach to legacy time-division multiplexing (TDM) call blocking. This ensured that new connections were blocked to preserve the quality of the existing connections. Without this feature, all users are negatively impacted when call volumes approached capacity.

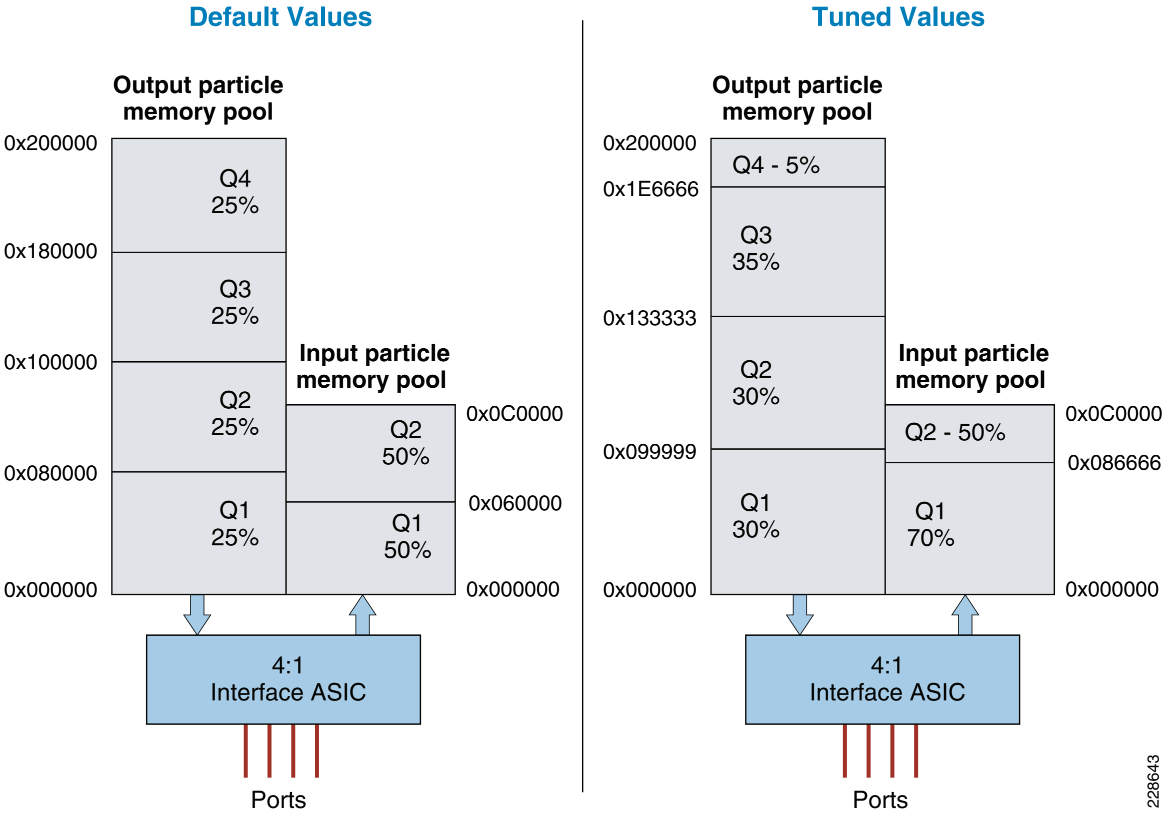
With medianet, there is not a comparable feature for video. As additional video is loaded onto a circuit, all user video experience begins to suffer. The best method is to ensure that aggregation of all real-time video does not exceed capacity, through provisioning. This is not always a matter of dividing the total bandwidth by the per-flow usage because frames are carried in grouped packets. For example, assume that two I-frames from two different flows arrive on the priority queue at the same time. The router places all the packets onto the outbound interface queue, where they drain off onto the TxRing for serialization on the wire. The next device upstream sees an incoming microburst twice as large as normal. If the RxRing saturates, it is possible to begin dropping packets at very modest 1 second average loads. As more video is added, the probability that multiple frames will converge increases. This can also load Tx Queues, especially if the multiple high speed source interfaces are bottlenecking into a single low-speed WAN link.

Another concern is when service provider policers cannot accept large, or back-to-back bursting. Video traffic that may naturally synchronize frame transmission is of particular concern and is likely to experience drops well below 90 percent circuit utilization. Multipoint TelePresence is a good example of this type of traffic. The Cisco TelePresence Multipoint Switch replicates the video stream to each participant by swapping IP headers. Multicast interfaces with a large fanout are another example. These types of interfaces are often virtual WAN links such as Dynamic Multipoint Virtual Private Network (DMVPN), or virtual interfaces such as Frame Relay. In both cases, multipoint flows fanout at the bandwidth bottleneck. The same large packet is replicated many times and packed on the wire close to the previous packet.

Buffer and queue depths of the Tx interface can be overrun. Knowing the queue buffer depth and maximum expected serialization delay is a good method to determine how much video an interface can handle before drops. When multiple video streams are on a single path, consider the probability that one frame will overlap or closely align with another frame. Some switches allow the user some granularity when allocated shared buffer space. In this case, it is wise to ensure buffers that can be expected to process long groups of real-time packets and have an adequate pool of memory. This can mean reallocating memory away from queues where packets are very periodic and groups of packets are generally small.

For now, some general guidelines are presented as the result of lab verification of multipoint TelePresence. Figure 2-8 shows both the defaults and tuned buffer allocation on a Cisco Catalyst 3750G Switch. Additional queue memory has been allocated to queues where tightly spaced packets are expected. By setting the buffer allocation to reflect the anticipated packet distribution, the interface can reach a higher utilization as a percent of line speed.

Figure 2-8 Default and Tuned Buffer Allocation



It may take some fine tuning to discover the values most appropriate to the load placed on the queues. Settings depend on the exact mix of applications using the interface.

Capacity Planning

Capacity planning involves determining the following:

- How much video is currently running over the network
- How much future video is expected on the network
- The bandwidth requirements for each type of video
- The buffer requirements for each type of video

The first item above is discussed in Chapter 6, "Medianet Management and Visibility Design Considerations." Tools available in the network such as NetFlow can help understand the current video loads.

The future video requirements can be more subjective. The recent trend is for more video and for that video to be HD. Even if the number of video streams stays the same, but is updated from SD to HD, the video load on the network will grow substantially.

The bandwidth requirements for video as a 1 second smoothed average are fairly well known. Most standard definition video consumes between 1–3 MB of bandwidth. High definition video takes between 4–6 Mbps, although it can exceed this with the highest quality settings. There are some variances because of attributes such as frame rate (fps) and encoding in use. Table 2-3 lists the bandwidth requirements of common video streams found on a medianet.

Table 2-3 BAndwidth Requirements of Common Video Streams

Video Source	Transport	Encoder	Frame Rate	Resolution	Typical Load[1]
Cisco TelePresence System 3000		H.264	30 fps	1080p	12.3 Mbps
Cisco TelePresence System 3000		H.264	30 fps	720p	6.75 Mbps
Cisco TelePresence System 1000		H.264	30 fps	1080p	4.1 Mbps
Cisco TelePresence System 1000		H.264	30 fps	720p	2.25 Mbps
Cisco 2500 Series Video Surveillance IP Camera		MPEG-4	D1 (720x480)	15 fps	1 Mbps
Cisco 2500 Series Video Surveillance IP Camera		MPEG-4	D1 (720x480)	30 fps	2 Mbps
Cisco 2500 Series Video Surveillance IP Camera		M-JPEG	D1 (720x480)	5 fps	2.2 Mbps
Cisco 4500 Series Video Surveillance IP Camera		H.264	1080p	30 fps	4–6 Mbps
Cisco Digital Media System (DMS)—Show and Share VoD		WMV	720x480	30 fps	1.5 Mbps
Cisco Digital Media System (DMS)—Show and Share Live		WMV	720x480	30 fps	1.5 Mbps
Cisco DMS—Digital Sign SD (HTTP)		MPEG-2	720x480	30 fps	3–5 Mbps
Cisco DMS—Digital Sign HD (HTTP)		MPEG-2	1080p	30 fps	13–15 Mbps
Cisco DMS—Digital Sign SD (HTTP)		H.264	720x480	30 fps	1.5–2.5 Mbps
Cisco DMS—Digital Sign HD (HTTP)		H.264	1080p	30 fps	8–12 Mbps
Cisco Unified Video Advantage	UDP/5445	H.264	CIF	variable	768 Kbps
Cisco WebEx	TCP/HTTPS		CIF	variable	128K per small thumbnail
YouTube	TCP/HTTP	MPEG-4	320x240		768 Kbps
YouTube HD	TCP/HTTP	H.264	720p		2 Mbps

[1] This does not include audio or auxiliary channels.

The one second smoothed average is influenced by the stream of P-frame. Although I-frames do not occur often enough to have a substantive influence over the average load, they do influence the burst size. From an overly simplified planning capacity standpoint, if a 10 percent overhead is added to the one second load, and the high end of the range is used, the planning numbers become 3.3 MB for standard definition and 6.6 MB for HD video. If you allow 25 percent as interface headroom, Table 2-4 provides some guidance for common interface speeds.

Table 2-4 Common Interface Speeds

Interface	Provisioned Rate	HD	SD
10 Gbps	7.5 Gbps	1136	2272
1 Gbps	750 Mbps	113	226
155 Mbps	11633 Mbps	17	34
100 Mbps	75 Mbps	11	22
45 Mbps	33 Mbps	5	10

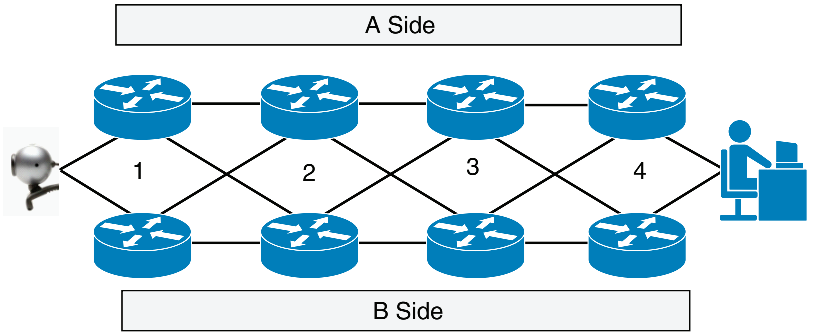
Note These values are based on mathematical assumptions about the frame distribution. They give approximate guidance where only video is carried on the link. These values, as of this writing, have not yet been validated, with the exception of TelePresence, where the numbers modeled above are appropriate. In cases where the encoder setting results in larger video streams, the values given here are not appropriate.

Load Balancing

Inevitably, there are multiple paths between a sender and receiver. The primary goal of multiple paths is to provide an alternate route around a failure in the network. If this is done at each hop, and the metrics are equal to promote load balancing, the total number of paths can grow to be quite large. The resulting binary tree is 2^(hop count). If the hop count is 4, the number of possible paths is 16 (2^4). If there were three next hops for each destination, the total number of paths is 3^(hop count). (See Figure 2-9).

Support and troubleshooting issues arise as the number of possible paths increases. These are covered in Chapter 6, “Medianet Management and Visibility Design Considerations.”

Figure 2-9 *Load Balancing*



Trace Route. Total possibilities = 2”

A-A-A-A	A-A-A-B	A-A-B-A	A-A-B-B
A-B-A-A	A-B-A-B	A-B-B-A	A-B-B-B
B-A-A-A	B-A-A-B	B-A-B-A	B-A-B-B
B-B-A-A	B-B-A-B	B-B-B-A	B-B-B-B

228644

Although not the primary purpose of redundant links, most designs attempt to use all bandwidth when it is available. In this case, it is prudent to remember that the load should be still be supported if any link fails. The more paths available, the higher utilization each path can be provisioned for. If a branch has two circuits, each circuit should run less than 50 percent load to allow failover capacity. If there are three paths available, each circuit can be provisioned to 66 percent capacity. At four paths, each is allowed to run at 75 percent total capacity, and still mathematically allow the load of any single failed path to be distributed to the remaining circuits. In the extreme case, the total bandwidth can be distributed onto so many circuits that a large size flow would congest a particular path.

The exercise can easily be applied to upstream routers in addition to the feeder circuits. As is often the case, there are competing objectives. If there are too many paths, troubleshooting difficulties can extend outages and decrease overall availability. If there are too few paths, expensive bandwidth must be purchased that is rarely used, and some discipline must be employed to ensure the committed load does not grow beyond the single path capacity. Port channels or Multilink PPP are methods to provide L1 redundancy without introducing excessive Layer 3 complexity. These each introduce other complexities and will be discussed in more detail in a future version of this document.

Another approach is to restrict some load that can be considered non-mission critical, such as the applications in the scavenger class. This is valid if you are willing to accept that not all applications will be afforded an alternate path. There are various ways to achieve this, from simple routing to more advanced object tracking.

Consider the following guidelines when transporting video with multi-path routing:

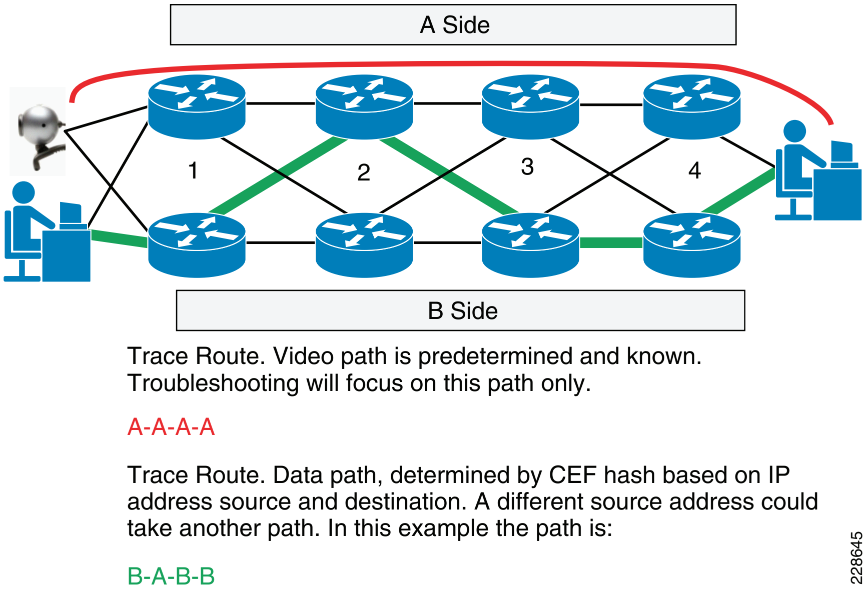
- Ensure that load balancing is per-flow and not per-packet—This helps prevent out-of-order packets. Per-flow also minimizes the chance of a single congested or failing link ruining the video. Because each frame is composed of multiple packets, in a per-packet load balancing scenario, each frame is spread over every path. If any one path has problems, the entire frame can be destroyed.
- Determine a preferred path—With equal cost metrics, the actual path may be difficult to discover. Tools such as trace cannot effectively discover the path a particular flow has taken over equal cost routes, because Cisco Express Forwarding considers both the source and destination address in the hash to determine the next hop. The source address used by trace may not hash to the same path as the stream experiencing congestion. If there are problems, it takes longer to isolate the issue to a particular link if the path is not deterministic. Enhanced Interior Gateway Routing Protocol (EIGRP) provides an offset list that can be used to influence the metric of a particular route by changing its delay. To make use of this feature, mission-critical video such as TelePresence needs to be on

dedicated subnets. The specific routes need to be allowed through any summary boundaries. Offset lists are used at each hop to prefer one path over another for just that subnet (or multiple subnets, as defined in the access control list). This method is useful only to set a particular class of traffic on a determined route, while all other traffic crossing the interface is using the metric of the interface. Offset lists do take additional planning, but can be useful to manage a balanced load in a specific and known way.

- When possible, load balance multiple circuits such that similar traffic is flowing together, and competing traffic is kept apart. For example, video and VoIP should both be handled in the priority queue as real-time RTP traffic. This can be done with the dual-PQ algorithm, or by setting each to prefer a unique path. Without special handling, it is possible that the large packets packed tightly in a video frame can inject jitter into the much smaller and periodic VoIP packets, especially on lower speed links where serialization delay can be a concern.
- Hot Standby Routing Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), and Gateway Load Balancing Protocol (GLBP)—These are all gateway next-hop protocols. They can be used to direct media traffic off of the LAN into the routed domain. HSRP and VRRP are very similar. VRRP is an open standards protocol while HSRP is found in Cisco products. HSRP does not provide load balancing natively but does allow multiple groups to serve the same LAN. The Dynamic Host Configuration Protocol (DHCP) pool is then broken into two groups, each with its gateway address set to match one of the two HSRP standby addresses. GLBP does support native load balancing. It has only a single address, but the participating devices take turns responding to Address Resolution Protocol (ARP) requests. This allows the single IP address to be load balanced over multiple gateways on a per-client basis. This approach also allows a single DHCP pool to be used.

  Both HSPR and GLBP can be used in a video environment. Ideally a given LAN is mission-specific for tasks such as video surveillance, digital media signage, or TelePresence. These tasks should not be on the same LAN as other types of default traffic. This allows unique subnets to be used. The design should consider the deterministic routing in the network as discussed above. Often multiple VLANs are used for data, voice, video, and so on. In this case, it may make operational sense to set the active address of each VLAN on a predetermined path that aligns with the routing. For example, real-time voice would use box A as the active gateway, while real-time video would use box B. In the example shown in Figure 2-10, voice and video are both treated with priority handling. Data and other class-based traffic can be load balanced over both boxes.

Figure 2-10 Load Balancing Example



Trace Route. Video path is predetermined and known. Troubleshooting will focus on this path only.

A-A-A-A

Trace Route. Data path, determined by CEF hash based on IP address source and destination. A different source address could take another path. In this example the path is:

B-A-B-B

These design considerations attempt to reduce the time required to troubleshoot a problem because the interfaces in the path are known. The disadvantage of this approach is that the configurations are more complicated and require more administrative overhead. This fact can offset any gains from a predetermined path, depending on the discipline of the network operations personnel. The worst case would be a hybrid, where a predetermined path is thought to exist, but in fact does not or is not the expected path. Processes and procedures should be followed consistently to ensure that troubleshooting does not include false assumptions. In some situations, load balancing traffic may be a non-optimal but less error-prone approach. It may make operational sense to use a simplified configuration. Each situation is unique.

EtherChannel

It is possible to bond multiple Ethernet interfaces together to form an EtherChannel. This effectively increases the bandwidth because parallel paths are allowed at Layer 2 without spanning tree blocking any of the redundant paths. EtherChannel is documented by the IEEE as 802.ad. Although EtherChannels do effectively increase the bandwidth to the aggregation of the member interfaces, there are a few limitations worth noting. First, packets are not split among the interfaces as they are with Multilink PPP. In addition, packets from the flow will use the same interface based on a hash of that flow. There are some advantages of this approach. Packets will arrive in the same order they were sent. If a flow was sent over multiple interfaces, some resolution is needed to reorder any out of order packets. However, this also means that the bandwidth available for any single flow is still restricted to a single member interface. If many video flows hash to the same interface, then it is possible that the buffer space of that physical interface will be depleted.

Bandwidth Conservation

There are two fundamental approaches to bandwidth management. The first approach is to ensure there is more bandwidth provisioned than required. This is easier in the campus where high speed interfaces can be used to attach equipment located in the same physical building. This may not always be possible where distance is involved, such as the WAN. In this case, it may be more cost-effective to try to minimize the bandwidth usage.

Multicast

Broadcast video is well suited for the bandwidth savings that can be realized with multicast. In fact, IPTV was the original driver for multicast deployments in many enterprise environments. Multicasts allow a single stream to be split by the network as it fans out to receiving stations. In a traditional point-to-point topology, the server must generate a unique network stream for every participant. If everyone is essentially getting the same video stream in real-time, the additional load on both the server and shared portions of the network can negatively impact the scalability. With a technology such as Protocol Independent Multicast (PIM), listeners join a multicast stream. If hundreds or even thousands of users have joined the stream from the same location, only one copy of that stream needs to be generated by the server and sent over the network.

There are some specific cases that can benefit from multicast, but practical limitations warrant the use of unicast. Of the all various types of video found on a medianet, Cisco DMS is the best suited for multicast. This is because of the one-to-many nature of signage. The benefits are best suited when several displays are located at the same branch. The network savings are not significant when each branch has only a single display, because the fanout occurs at the WAN aggregation router, leaving the real savings for the high speed LAN interface.

Aside from DMS, other video technologies have some operational restrictions that limit the benefits of multicast. For example, TelePresence does support multipoint conference calls. However, this is accomplished with a Multipoint Conferencing Unit (MCU), which allows for unique control plane activity to manage which stations are sending, and which stations are the receivers. The MCU serves as a central control device. It also manipulates information in the packet header to control screen placement. This helps ensure that participants maintain a consistent placement when a conference call has both one screen and three screen units.

IP Virtual Server (IPVS) is another technology that can benefit from multicast in very specific situations. However, in most cases, the savings are not realized. Normally, the UDP/RTP steams from the camera terminate on a media server, and not directly on a display station. The users use HTTP to connect to the media server and view various cameras at the discretion of the user. Video surveillance is a many-to-one as opposed to one-to-many. Many cameras transmit video to a handful of media servers, which then serve unicast HTTP clients.

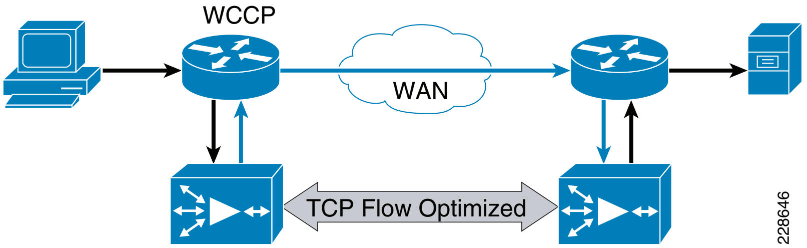
For a more detailed look at video over multicast, see the Multicast chapter in the *Cisco Digital Media System 5.1 Design Guide for Enterprise Medianet* at the following URL: http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/DMS_DG/DMS_DG.html.

Cisco Wide Area Application Services

Cisco Wide Area Application Services (WAAS) is another technique that can be used to more efficiently use limited bandwidth. Cisco WAAS is specifically geared for all applications that run over the WAN. A typical deployment has a pair or more of Cisco Wide Area Application Engines (WAEs) on either side of the WAN. The WAEs sit in the flow of the path, and replace the segment between the WAEs with an optimized segment. Video is only one service that can benefit from WAAS. Other WASS components include the following:

- TCP Flow Optimization (TFO)—This feature can help video that is transported in TCP sessions (see Figure 2-11). The most common TCP transport for video is HTTP or HTTPS. There are also video control protocols such as RTSP and RTP Control Protocol (RTCP) that use TCP and benefit from WAAS, such as Cisco DMS. TFO can shield the TCP session from WAN conditions such as loss and congestion. TFO is able to better manage the TCP windowing function. Whereas normal TCP cuts the window size in half and then slowly regains windowing depth, TFO uses an sophisticated algorithm to set window size and recover from lost packets. Video that is transported over TCP can benefit from WAAS, including Adobe Flash, Quicktime, and HTTP, which commonly use TCP. RTP or other UDP flows do not benefit from TFO.

Figure 2-11 TFO



- Data Redundancy Elimination—WAAS can discover repeating patterns in the data. The pattern is then replaced with an embedded code that the paired device recognizes and replaces with the pattern. Depending on the type of traffic, this can represent a substantial savings in bandwidth. This feature is not as useful with video, because the compression used by the encoders tends to eliminate any redundancy in the data. There may still be gains in the control plane being used by video. Commonly these are Session Initiation Protocol (SIP) or RTSP.
- Persistent LZ Compression—This is a compression technique that also looks for mutual redundancy, but in the bit stream, outside of byte boundaries. The video codecs have already compressed the bit stream using one of two techniques, context-adaptive binary arithmetic coding (CABAC) or context-adaptive variable-length coding (CAVLC). LZ Compression and CABAC/CAVLC are both forms of entropy encoding. By design, these methods eliminate any mutual redundancy. This means that compressing a stream a second time does not gain any appreciable savings. This is the case with LZ compression of a video stream. The gains are modest at best.

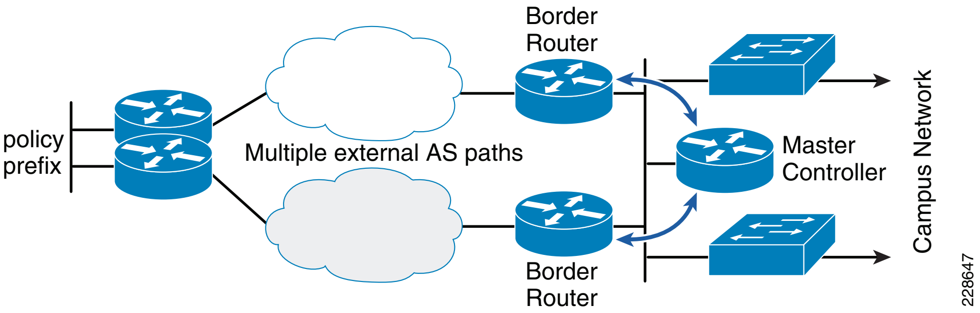
Cisco Application and Content Network Systems

Cisco Application and Content Network Systems (ACNS) is another tool that can better optimize limited WAN bandwidth. Cisco ACNS runs on the Cisco WAE product family as either a content engine or content distribution manager. Cisco ACNS saves WAN bandwidth by caching on-demand content or prepositioning content locally. When many clients in a branch location request this content, ACNS can fulfill the request locally, thereby saving repeated requests over the WAN. Of the four technologies that form a medianet, ACNS is well suited for Cisco DMS and desktop broadcast video. For more information, 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_dgbk.pdf.

Cisco Performance Routing

Cisco Performance Routing (PfR) is a feature available in Cisco routers that allows the network to make routing decisions based on network performance. This tool can be used to ensure that the WAN is meeting specific metrics such as loss, delay, and jitter. PfR can operate in either a passive or active mode. One or more border routers is placed at the edge of the WAN. A master controller collects performance information from the border routers and makes policy decisions. These decisions are then distributed to the border routers for implementation. Figure 2-12 shows a typical topology.

Figure 2-12 Typical Topology using PfR



Multiprotocol Environments

In the early days of networking, it was common to see multiple protocols running simultaneously. Many networks carried IP, Internetwork Packet Exchange (IPX), Systems Network Architecture (SNA), and perhaps AppleTalk or DEC. It was not uncommon for an IPX Service Advertising Protocol (SAP) update to occasionally cause 3270 sessions to clock. Modern networks are increasingly IP only, yet convergence remains a concern for the same reason: large blocks of packets are traveling together with small time-sensitive packets. The difference now is that the large stream is also time-sensitive. QoS is the primary tool currently used to ensure that bandwidth is used as efficiently as possible. This feature allows UPD RTP video to be transported on the same network as TCP-based non-real-time video and mission-critical data applications. In addition to many different types of video along with traditional data and voice, new sophisticated features are being added to optimize performance, including those discussed here: Cisco WAAS, multicast, Cisco ACNS, PfR, and so on, as well as other features to support QoS, security, and visibility. New features are continuously being developed to further improve network performance. The network administrator is constantly challenged to ensure that the features are working together to obtain the desired result. In most cases, features are agnostic and do not interfere with one another.

Note Future revisions to this chapter will include considerations where this is not the case. For example, security features can prevent WAAS from properly functioning.

Summary

Bandwidth is an essential base component of a medianet architecture. Other features can help to maximize the utilization of the circuit in the network, but do not replace the need to adequately provisioned links. Because CAC-like functionality is not yet available for video, proper planning should accommodate the worst-case scenario when many HD devices are present. When network bandwidth saturates, all video suffers. Near-perfect HD video is necessary to maximize the potential in productivity gains. Bandwidth is the foundational component of meeting this requirement, but not the only service needed. Other functionality such as QoS, availability, security, management, and visibility are also required. These features cannot be considered standalone components, but all depend on each other. Security requires good management and visibility. QoS requires adequate bandwidth. Availability depends on effective security. Each feature must be considered in the context of an overall medianet architecture.