Engineering DOCSIS™ for Voice over IP
2003 SCTE CableTec Expo

Doug Jones
YAS Broadband Ventures
300 Brickstone Square
Andover, MA 01810
voice: (978) 749-9999 x226
doug@yas.com

1 Introduction

1.1 Executive Summary

A G.711 CODEC generates 64 kbps of voice sample. Depending on how the VoIP parameters are set, that CODEC could use up to 180 kbps of return path bandwidth per call. The network engineer should understand all the parameters associated with a VoIP call, including the DOCSIS parameters, to properly engineer the cable data network. This paper discusses choices an operator should consider in order to have the network operate at peak efficiency.

1.2 Abstract

Based on recent work with cable operators, a key driver for moving to DOCSIS™ 1.1 is to deploy PacketCable™ Voice over IP (VoIP) service. This paper discusses:

1. How VoIP impacts bandwidth on the DOCSIS 1.1 link,

2. DOCSIS and PacketCable strategies to engineer the access network for VoIP.

Considering an IP network that includes all the data connections in the metro and backbone networks, the DOCSIS link probably has the least amount of available bandwidth. As a result, the DOCSIS link is the most likely to experience congestion. Engineering the DOCSIS link is necessary for a good VoIP service.

Adding PacketCable VoIP service to a cable data network will require both an understanding of how DOCSIS and PacketCable work together and new engineering guidelines to ensure data and VoIP services coexist harmoniously on the DOCSIS link. VoIP calls generally require a dedicated amount of bandwidth that must be available regardless of other data traffic on the DOCSIS link.

The analysis will cover several areas, including:

- Issues with mixing data and voice on the same DOCSIS™ channel
- Calculating the number of voice channels available over a DOCSIS™ link by including the physical layer overhead. (Most models just include higher layer protocol overhead, e.g., IP/UDP/RTP.) This paper will show that non-optimized parameters can add as much as 200% additional overhead to a VoIP call, using up return path bandwidth.
- Determining the number of VoIP subscribers that can be supported on a DOCSIS channel.
- QoS “adjustments” needed on a DOCSIS™ 1.1 CMTS to control PacketCable™ 1.0 service.

PacketCable Voice is based on DOCSIS 1.1 and the reader is assumed to have a basic familiarity with both.

1.3 Document Organization

The document is organized as follows:

- Section 2 contains background on the DOCSIS and PacketCable projects and includes additional background information on issues surrounding packet voice service.
- Section 3 describes how bandwidth is allocated on the DOCSIS return path, including how DOCSIS QoS mechanisms operate.
- Section 4 discusses engineering DOCSIS for VoIP including how to determine the amount of bandwidth used by a VoIP call and the number of VoIP subscribers that can be supported on a DOCSIS system.
- Section 5 finishes the engineering discussion by introducing strategies to support more VoIP subscribers on a DOCSIS system. These strategies are part of the DOCSIS/PacketCable toolkits and are possible areas for discussion with suppliers.

2 VoIP Background

2.1 DOCSIS 1.1 and PacketCable

Engineering a DOCSIS network for PacketCable means being able to engineer that DOCSIS network to support Quality of Service (QoS). DOCSIS 1.0 was designed to offer “best effort” high-speed data service. The Internet was built on this “best effort” assumption that guaranteed packet delivery was not required. The network could be slow, drop packets, or even fail from time to time. This generally works for services such as email, peer-to-peer, and web surfing. But "best effort" Internet service does not properly support the emerging Internet-based voice and video applications.

In 1998 as the DOCSIS 1.0 specification was being wrapped up, the engineering team was directed to begin work on a version of the specification to include Quality of Service (QoS). The DOCSIS 1.1 specification, which includes QoS, was published in 1999.

In the same time frame, the PacketCable team was designing a method to offer Voice over IP service. The PacketCable specifications were also first published in 1999. At this time, there are four published versions of PacketCable as follows:

- PacketCable 1.0
  Eleven specifications and six technical reports that define the call signaling, Quality of Service (QoS), CODEC, client provisioning, billing event message collection, PSTN (Public Switched Telephone Network) interconnection, and security interfaces necessary to implement a single-zone PacketCable solution for residential Internet Protocol (IP) voice services.

- PacketCable 1.1
  Five specifications and four technical reports define requirements for offering a Primary Line-capable service using the PacketCable architecture. The designation of a communications service as "primary" means that the service is sufficiently reliable to meet an assumed consumer expectation of essentially constant availability. This

also includes, specifically, availability during power failure at the customer's premises and (assuming the service is used to connect to the PSTN) access to emergency services (911, etc.).

- PacketCable 1.2
  Two specifications and one technical report define the functional components and interfaces necessary to allow communication between PacketCable 1.0 networks using an IP transport or backbone network. These specifications describe the call signaling and quality-of-service (QoS) extensions to the PacketCable 1.0 architecture to enable cable operators to directly exchange session traffic. This will allow a subscriber on one PacketCable network to establish end-to-end IP or "on-net" sessions with subscribers on other PacketCable networks. For PacketCable, "on-net" means that the call is established end-to-end on the IP network without traversing the Public Switched Telephone Network (PSTN) network at any time.

- PacketCable 1.3
  A single specification that defines the functional components, interfaces, and data model to perform subscriber provisioning on a Call Management Server (CMS).

Together, this group of specifications and reports is collectively referred to as PacketCable 1.x.

DOCSIS 1.1 is specifically included as a part of PacketCable 1.0 in the Dynamic Quality of Service (DQoS) specification. It is this specification that describes how DOCSIS 1.1 QoS is dynamically allocated to VoIP calls based on PacketCable signaling.

2.2 Packet Voice

Everyone should be familiar with circuit-switched voice service; it's what the telephone companies have been offering for over 100 years. In this scenario, the network is configured such that there is a virtual dedicated circuit connecting the callers. Creating these connections is known as circuit switching. The call connection, once established, is designed to be a zero loss environment.

Packet voice relies on a different network architecture where there is no dedicated connection between the callers. Rather, the "call" is placed into packets and sent onto a packet switched network. Once on the packet network, individual packets could be delayed or dropped altogether, adding new challenges for the engineers to ensure voice quality standards are met (or exceeded).

In packet voice, a key component is called the CODEC, which is short for "COder/DECoder." At the transmitting end, the CODEC samples the spoken analog voice signal, digitizes it, and packages it into individual samples that are then sent across the IP network. At the far end, there is a matching CODEC that receives the voice samples and at the correct times converts them back into an analog signal, which the listener then hears. CODECs are generally described by the number of bits per second

they generate during a voice call, e.g., a G.711 CODEC generates 64 kilobits per second (kbps). The "G.711" name comes from the ITU-T (International Telecommunication Union – Telecom), which has adopted this particular CODEC as an international standard.

Packet voice has interesting consequences that are handled by different types of Internet protocols.

1. The packets may not be delivered in order. The Real Time Protocol (RTP) handles this by timestamping each packet. The far end connection uses these timestamps to order the voice samples (packets) for playback.

2. Packets may be delayed. This is called latency and is a measure of the time between when a sound is uttered on the calling end and that sound is heard at the receiving end. The generally accepted one-way maximum latency for a voice call is 250 milliseconds whereas a "toll quality" call is generally allocated 150 milliseconds of latency. Toll Quality is a term used by telephone carriers and represents a minimum service objective for their calls. Packets generated by the sending CODEC must be received at the terminating CODEC within 250 milliseconds or it becomes difficult for the parties in the conversation to tell when one person is finished speaking, thus increasing the probability that the parties will talk at the same time. As a rough rule of thumb, a packet voice network is engineered such that 70 milliseconds of latency is allowed on the access network, 70 milliseconds is allowed across the metro/backbone network, and 70 milliseconds is allocated for the receiver buffer. This worst-case total of 210 milliseconds is not quite toll quality, but does meet the 250 millisecond limit where call quality is degraded. In a later section, it will be described how DOCSIS 1.1 QoS provides access network latency more on the order of 20 – 40 milliseconds, depending on the CODEC choice.

3. Since it's an IP network with several router hops, packets may take varying amounts of time to get from the sender to the receiver. This is called jitter. The receiving end usually implements a buffer where voice samples are ordered and queued for playback. In a later section, we'll see that DOCSIS 1.1 QoS can control jitter to within a millisecond, therefore making the access network a negligible source of jitter.

4. Packets may be dropped. After all, it's an IP network and congestion can happen. There are several methods for the receiving end of the connection to handle this and one common method is to have the receiver keep replaying the most recent received packet until it is time to playback the next sample in the receive buffer. With DOCSIS 1.1 QoS, the voice samples will be guaranteed a place on the access network; therefore, dropping a voice packet would be due to congestion on either the metro or backbone network and not the DOCSIS network.

5. Silence is really silence. If the sending CODEC does not detect the speaker talking, it may not send any packets. This implementation choice relies on the fact that on average a person talks less than 50% of the time during a normal phone call. Why send packets that contain 50% silence ? While the user is not speaking, the CODEC does not generate voice samples which has the benefit of conserving network bandwidth. A side effect of this technique is that to the receiver, during periods when the sender's CODEC does not transmit, will hear complete silence. This could be unsuitable as the phone may appear to chop in and out during the conversation and the user may think the call has dropped. This is overcome by the receiver implementing "comfort noise." Basically this is noise added into the call when the far end is not speaking to mimic the background noise present on an actual call.

Creating a quality packet voice solution clearly presents technical challenges but much work has been done in this area and solutions are available.

3 How DOCSIS QoS Works

When a packet network is congested and packets are being delayed or dropped, QoS can guarantee that certain packets will always get through on time. DOCSIS 1.1 has these mechanisms.

3.1 DOCSIS 1.0

DOCSIS 1.0 does not have QoS; however, it does deliver Class of Service (CoS).

DOCSIS 1.0 CoS is a best-effort service that can offer bandwidth maximums and relative priorities to some traffic. Neither packet latency nor jitter is guaranteed. The network should be fair to all the CMs, but when the traffic applied is more than the connection can carry, the network has no choice except to delay and drop packets, generally in a random fashion. DOCSIS 1.0 does not have the tools to ensure voice packets will always get delivered over the CM/CMTS link.

Packet voice can be offered over DOCSIS 1.0 and there are several services that take advantage of this, including net2phone® and Vonage. However, DOCSIS 1.0 cannot offer QoS to these services. With DOCSIS 1.0, good quality packet voice relies on the DOCSIS return path being lightly loaded. As stated earlier, the DOCSIS network is generally the lowest bandwidth connection in a packet voice network and so more likely to experience congestion.

3.2 DOCSIS 1.1

DOCSIS 1.1 provides Quality of Service, including latency, jitter, and bandwidth guarantees. The mechanisms to provide QoS differ on the forward and return paths and are discussed below.

3.2.1 Return path DOCSIS QoS

The cable data return path has the unique property of having many attached CMs that may all want to transmit at the same time. Clearly this needs to be controlled, and in fact the CMTS is the "traffic cop" that tells each CM when it can transmit and how much data it can send.

The key underlying technology needed to offer QoS on the return path is called the minislot. On the return path time is divided into periods called minislots. The minislot size is based on a number of parameters, but is generally set to one of 4, 8, 16, 32, or 64 Bytes of data. The size of the minislot is fixed on the return path, but will be one of the above sizes based on calculations done by the CMTS.

When a CM has data to transmit, it requests enough minislots on the return path to send that data. For instance if the CM has 500 Bytes of data to send and the minislot size is 32 Bytes, the CM would "request" 16 minislots (divide 500 by 32 and round up) from the CMTS. The CMTS will "grant" these minislots to the CM and the CM will transmit appropriately. The minislots are numbered to allow the CMTS and the CM to agree on which minislots to use.

The CMTS is generally configured to use a minislot size in the "sweet spot" of 8, 16, 32 or 64 Bytes. Note that a chief competitor to cable data, Digital Subscriber Line (DSL), is based on Asynchronous Transfer mode (ATM) which uses a 53-Byte "cell." Cells and minislots are the same concept. But whereas the ATM cell is fixed at 53 Bytes, DOCSIS can provide even finer granularity by using 8, 16, or 32 Byte minislots or comparable granularity using a 64-Byte minislot.

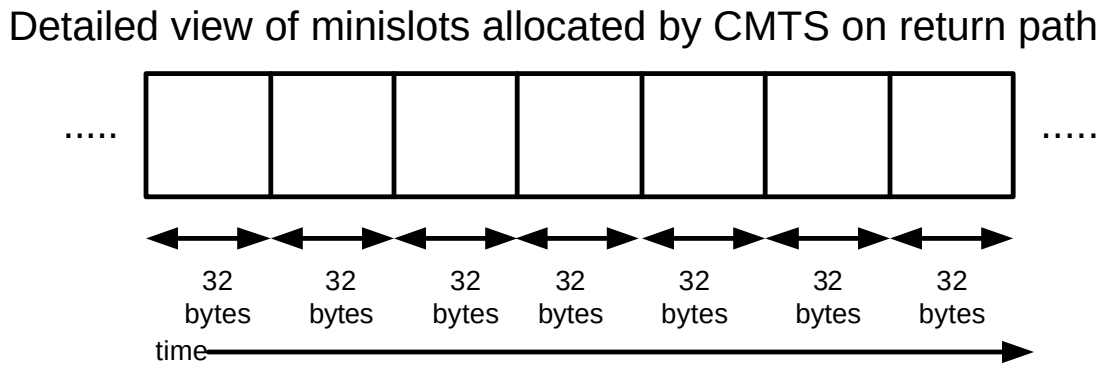

Figure 1

Figure 1 shows the conceptual view of 32 Byte minislots flowing on the return path.

Figure 2 shows how groups of minislots can be allocated for a single phone call. The CMTS assigns groups of minislots to the CM for upstream data transmission. By assigning certain groups of minislots at certain times to a CM, the latency and jitter of data from the CM is managed. Similarly, the total amount of bandwidth available to that

CM can be managed. The remaining minislots could be assigned to other CMs for upstream data transmission.

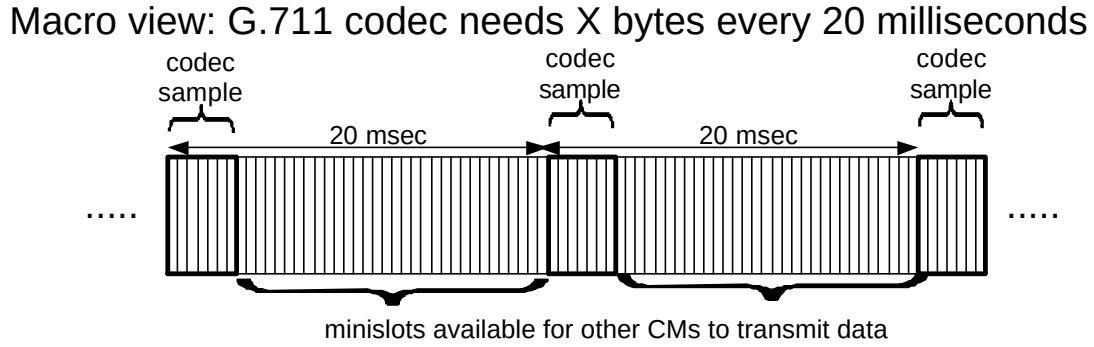

Figure 2

The underlying QoS capabilities in DOCSIS 1.1 can provide a better customer experience for time-sensitive services including PacketCable voice.

3.2.2 Forward path DOCSIS QoS

The cable data forward path is different from the return in that only one device transmits at a time, the CMTS. Therefore, on the forward path the CMTS internally queues and transmits packets as necessary to meet the QoS guarantees. There are no minislots on the forward path; these are available only on the return path. But because there is only one transmitter on the forward path, there is no need to allocate minislots for bandwidth. All the bandwidth is allocated to the CMTS to transmit packets.

3.2.3 Unsolicited Grant Service (UGS)

As can be gleaned from the preceding two sections, the real action in DOCSIS 1.1 QoS is on the return path. Because the return path is the lowest bandwidth connection on the cable data network, the need for QoS is greater. In fact, DOCSIS 1.1 has several types of QoS on the return path, but the one used for PacketCable is called Unsolicited Grant Service (UGS). UGS is not available on the forward path.

UGS is designed to support real-time services that generate fixed size data packets on a periodic basis, such as VoIP. With UGS, the CMTS allocates a fixed number of minislots to a CM on a real-time periodic basis.

The “U” in UGS stands for unsolicited. This means the opportunities for the CM to transmit are not solicited, rather, the CMTS just provides these opportunities to the CM as they are needed. UGS eliminates the overhead and latency of the usual CM requests/grant state machine (as used in best effort data). In PacketCable, UGS service is configured between the CM and CMTS using DQoS as described in the next section.

3.3 PacketCable Dynamic Quality of Service (DQoS)

DQoS is a very powerful tool that takes advantage of DOCSIS 1.1 QoS. Using DQoS, the operator does not need to provision the CM to offer QoS. Rather, the PacketCable application (in this case, the voice application of the Media Terminal Adapter [MTA]) causes the CM to signal the CMTS to request the QoS dynamically, as it's needed for the phone call. Since voice uses USG service, there is bandwidth dedicated to the call. This bandwidth should only be dedicated while the call in progress. Without the ability to dynamically manage the bandwidth, it might otherwise always be dedicated whether a call is in progress or not. DQoS allows the QoS to be dedicated only when it's needed.

In PacketCable Voice, DQoS works like this. After the MTA has initiated a call with the softswitch, the MTA will cause the CM to signal the CMTS to start the UGS QoS for that phone call. This CM/CMTS signaling is defined in DOCSIS 1.1, but it is the MTA voice application that tells the CM when to initiate it. When the call is completed, the MTA again signals the CM to work with the CMTS to end that QoS. Because DQoS is under the control of the PacketCable MTA, the associated CM configuration file does not contain QoS settings for the phone call. Rather, the MTA signals this information to the CM which then creates the UGS flow with the CMTS.

4 DOCSIS Bandwidth Needed for a VoIP Call

The paper has so far discussed the concepts of packet voice, and some of the specifics of PacketCable voice. In this section, the specifics of PacketCable are further discussed, including the direct impact on the DOCSIS connection. The intent is to determine how much bandwidth is needed for a VoIP call. This is a fundamental quantity needed to engineer the DOCSIS network for VoIP. While the most well-known voice CODEC uses 64 kbps of bandwidth, when wrapped in unoptimized PacketCable and DOCSIS overheads, the total amount of bandwidth used can climb to almost triple that. Good engineering will optimize the bandwidth usage, allowing more VoIP customers.

Carrying a CODEC sample on the return path takes more bandwidth than a similar sample on the forward path because there is additional overhead on the return path. The additional return path overhead will be explained in this section, including calculations.

Note, the bandwidth calculations that come out of this section are applicable for the DOCSIS connection only, CM to CMTS. They do not apply to the metro or backbone links because there is less overhead associated with these connections.

4.1 CODEC sample rates

How many bits per second are needed for a voice call? It depends on several factors including what CODEC is used and what the protocol overheads are.

CODECs have names such as G.711, G.728, and G.729. These particular names, G.XXX, come from their reference as a standard in the International Telecommunications Union (ITU). CODECs do a fairly simple job; they take an analog signal and digitize it (and visa versa). In this case the analog waveform is voice. The trade-off with CODECs is using the fewest bits to give good sound quality.

PacketCable requires the G.711 CODEC, and the G.728 and G.729E CODECs are optional. A brief description of each CODEC is given below:

– G.711 (required) - This CODEC provides toll-quality bit rate (64 kbps or 8,000 Bytes per second) and is ubiquitous. It provides the "fallback" position for services such as fax, modem, and hearing-impaired services support, as well as common Media Gateway support. G.711 is free of Intellectual Property Rights (IPR).

– G.728 (optional) - a mid-bit rate (16 kbps or 2000 Bytes per second), high-quality solution.

– G.729 Annex E (optional) - a mid-bit rate (12 kbps or 1500 Bytes per second), high-quality solution.

While each CODEC uses a specific number of bits per second, the number of IP packets per second and the size of those packets is the key to calculating the bandwidth used by a VoIP call on the return path. Since IP packets are usually referred to as a certain number of Bytes in length (and not bits), from now on, the paper will deal with Bytes.

A voice CODEC can generally sample at any of 10 milliseconds, 20 milliseconds, or 30 milliseconds. That is, while the G.711 CODEC puts out a steady stream of 8,000 Bytes per second, it can do this one of several ways. These ways include generating 80 Bytes samples 100 times per second (sampling every 10 milliseconds), generating 160 Byte samples 50 times per second (sampling every 20 milliseconds), or generating 240 Byte samples at a rate of 33.3 times per second (sampling every 30 milliseconds). The following table illustrates this for the three CODECs that have been introduced:

	Sample Rate		
	10 milliseconds	20 milliseconds	30 milliseconds
G.711 (8000 Bytes/sec)	80 Bytes	160 Bytes	240 Bytes
G.728 (2000 Bytes/sec)	20 Bytes	40 Bytes	60 Bytes
G.729E (1500 Bytes/sec)	15 Bytes	30 Bytes	45 Bytes

Table 1

So while a G.711 CODEC generates 8000 Bytes per second, it can do this by generating any of 33, 50, or 100 voice samples per second. Each voice sample goes in its own IP packet. For further calculations, the G.711 CODEC will be used with a 20 millisecond

sample rate (50 samples per second). That is, the VoIP call will generate 50 IP packets per second and each of these packets will include 160 Bytes of voice sample plus additional protocol overhead (RTP, UDP, IP, and DOCSIS), which will be explored in the next section.

Note that the sampling interval is related to the latency of the access network. That is, since the samples are generated every 20 milliseconds, this is an automatic 20 milliseconds of latency. As described earlier, UGS is used on the DOCSIS connection to give that MTA (or more specifically the CM embedded in the MTA) an opportunity to transmit that sample every 20 milliseconds. Now suppose the CODEC generates the sample but the opportunity to transmit (the UGS grant) just passed a millisecond ago and the MTA has to hold onto that sample for another 19 milliseconds before the next UGS grant comes along. Now the access latency is up to 39 milliseconds (20 + 19). This is the worst case possibility for the 20 millisecond sample rate.

If a G.711 CODEC is used at a 10 millisecond sample rate and DOCSIS gives a UGS grant every 10 milliseconds, and again the sample comes right after the last grant, the access latency would be 19 milliseconds. Similarly, a G.711 CODEC at a 30 millisecond sample rate could experience a maximum of 59 milliseconds latency (30 millisecond samples from the CODEC plus waiting up to 29 milliseconds for the next opportunity to transmit).

4.2 Protocol Overhead

In the previous section, it was shown that a G.711 CODEC operating at a sample rate of 20 milliseconds will generate 160 Bytes of voice sample at a rate of 50 times a second. The 160 Bytes is just voice sample; neither the IP packet overhead nor the DOCSIS overhead has been figured in yet. In this section, each protocol overhead will be explained.

The protocol stack for a CODEC sample is shown in Figure 3. As can be seen, each protocol requires a certain number of overhead Bytes to be added to the packet.

These protocol Bytes are considered overhead because they are not the voice sample, but, rather they are needed to get that voice sample across the network. An individual CODEC sample is 160 Bytes, but each protocol will add to the total size of that frame that gets sent on the DOCSIS link.

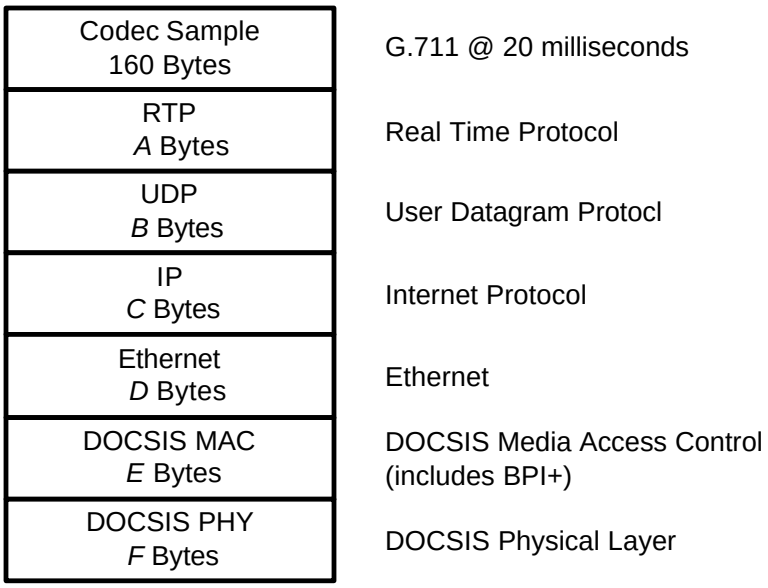

Figure 3

Another way to represent the overheads is shown in Figure 4. While the lengths of the overheads used in this diagram are not intended to be relative, it does show that the size of the overall frame sent on the DOCSIS return path will grow based on the protocol overheads.

Figure 4

All of these overhead Bytes add up and must be accounted for when engineering the return path for VoIP.

4.2.1 Upper Layers

For the purpose of this section, upper layer protocol overhead include protocols associated with the DOCSIS MAC layer and up. Specifically, these are the DOCSIS MAC, Ethernet, IP, UDP, and RTP. It is fairly straightforward to show the overheads for these layers, and they are as follows:

RTP	12 Bytes
UDP	8 Bytes
IP	20 Bytes
Ethernet	18 Bytes
DOCSIS MAC	14 Bytes

Table 2

Briefly, each of these protocols will be described.

– RTP – Real-time Transport Protocol provides for end-to-end delivery of data with real-time characteristics, such as interactive audio and video. The delivery services include payload type identification, sequence numbering, time stamping, and delivery monitoring. RTP itself does not provide any mechanism to ensure timely delivery nor does it provide quality-of-service guarantees, rather, it carries a timestamp of when the voice sample was generated and a sequence number that orders that particular voice sample against other voice samples.

– UDP – User Datagram protocol. The Internet has defined two general transport protocols, TCP and UDP. UDP is "fire and hope," that is, the packet is launched into the network and there is no guaranteed delivery of the packet. When going through the Internet, there can be router congestion, over subscribed connections, etc. In these cases packets can be dropped and with UDP there is no retransmission of dropped packets. TCP is different in that there is a guarantee of packet delivery, e.g., "fire and forget." TCP causes packets to be numbered; therefore, a dropped TCP packet is missed and will be retransmitted.

– IP – Internetworking Protocol, generally known as the Internet Protocol. The IP layer carries source and destination IP addresses as well as other information needed for a packet to be routed across an IP network.

– Ethernet – Ethernet provides addressing on a particular point-to-point connection, specifically the Ethernet MAC addresses. In this case, the point-to-point connection is between the CM and CMTS.

– DOCSIS MAC – the DOCSIS MAC header provides information for carrying a frame of data on the connection between the CM and CMTS. In the above table, an overhead of 14 Bytes is included. These 14 Bytes are accounted for as follows:
  a. 6 Bytes for the basic DOCSIS MAC header
  b. 5 Bytes for the BPI+ Header
  c. 3 Bytes for UGS/PHS extended DOCSIS MAC header

  The PacketCable security specification calls for BPI+ to be enabled on the DOCSIS connection. A specific UGS header is needed because DQoS uses UGS for the voice traffic. Extra Bytes must be added to the DOCSIS header to support UGS. This particular extended header supports both UGS and a service called Payload Header Suppression (PHS), which will be discussed later in the document.

These overheads add 72 Bytes to the voice sample, bringing the size of the frame up to 72+160 = 232 Bytes. Since in this example there are 50 samples per second, so far the VoIP connection is using:

$$\frac{232\ \text{bytes}}{\text{sample}} \times \frac{50\ \text{samples}}{\text{second}} \times \frac{8\ \text{bits}}{\text{byte}} = 92.8\ \text{Kbps}$$

But there are still additional overheads to be included in the next section.

4.2.2 Physical Layer

This section describes overhead associated with the DOCSIS physical layer, e.g., guard time, preamble length, FEC, and conversion to minislots.

4.2.2.1 Return Path Physical Layer Attributes

The return path physical layer attributes are defined in the CMTS and signaled to the CMs over the forward path. CMs use these parameters when transmitting on the return path. Table 3 contains example return path physical layer settings used on a DOCSIS 1.1 CMTS.

IUC	Modulation	Preamble length (Bytes)	FEC (T)	FEC Code-word size (Bytes)	Max mini slots	Guard Time (Bytes)	Short Last Codeword
Short Data	QPSK	9	5	75	6	2	yes
Long Data	QPSK	10	8	220	0	2	yes

Table 3

The information in the table will be explained below. Note, some physical layer attributes that are kept at the CMTS but that are not relevant to the calculations in this section have been removed from the table. Also, all values are converted to decimal whereas some CMTSes will display these values in hexadecimal. Finally where applicable, values are shown in Bytes whereas some CMTSes display these values in bits.

The calculations in this section depend on the parameters in Table 3. For instance, the modulation is assumed to be QPSK. Using 16 QAM on the return path will change the results. There is a lot of flexibility in the DOCSIS physical layer, and the operator has the ability to tune the parameters to gain more efficiency on the return path. This is a part of engineering the DOCSIS return path for VoIP.

4.2.2.2 DOCSIS Physical Layer Overhead

Consider a CM with has to send data onto the return path. This section will go through the steps the CM follows, and these are:

1. Add Reed-Solomon Forward Error Correction (FEC) by blocking the data (specifically the 232 Bytes described in a previous section) into codewords that include both information Bytes and parity Bytes. The applicable columns in the table are "FEC Codeword Size" which is the number of information Bytes and "FEC (T)" which is used to calculate the number of parity Bytes included in each codeword. This is very similar to the old telco modem lingo of 7 bits data, 1 bit parity. FEC uses X Bytes of codeword information and Y Bytes of parity. X is the number of codeword information Bytes and Y = (2 * T) Bytes.

   The CM will first use the row in the table for the short data grant and if the amount of data is too large to fit, it will try to use the long data grant. Following this, the original 232 Bytes of data is blocked into codewords that are 85 Bytes long, each with 75 Bytes of information and an additional 10 Bytes of parity (two times the "T" value in the table).

   Following this logic, there will be four codewords as shown in Figure 5. The first three codewords will have 75 Bytes of information and an additional 10 Bytes of parity. The remaining information will go into a "shortened last codeword" of 16 Bytes that also has 10 Bytes of parity. The 16 Bytes comes from 7 Bytes of information and 9 Bytes of padding. Note the table has a column labeled "Short Last Codeword" that is set to "yes" for the short data grant. This feature allows the last codeword to be shortened below the value specified in "FEC Codeword Size," but the last codeword must be a minimum of 16 Bytes long. When doing the last codeword, if the data to be sent is less than 16 Bytes, padding will be added to bring it to 16 Bytes. If "Short Last Codeword" is set to "no", then the last codeword would have to be padded out to the full number of information Bytes before adding the parity Bytes. Only the last codeword can be shortened but sometimes the first codeword will be the only codeword. In this case, that one codeword may be shortened since when there is only one codeword, it is also the last codeword.

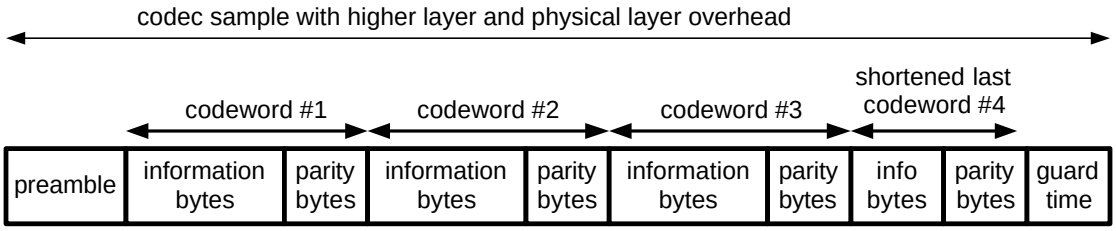

Figure 5

Doing the math, there will be four codewords. The first three will include 75 Bytes of information and 10 Bytes of parity. The fourth codeword will include 16 Bytes of information/padding and 10 Bytes of parity. After adding FEC, the amount of data to send is now up to 281 Bytes [(75+10)+(75+10)+(75+10)+(16+10)].

2. Add Preamble
   After the FEC step, a preamble is added to the data. The preamble is a known sequence of bits that the CMTS expects to be at the front of the data. The CMTS uses this known bit string to train its receiver to the channel characteristics to better decode the rest of the data in that transmission. The time during which the preamble is sent is also used for the CMs transmitter power to ramp up and stabilize.

   In the case of the short data grant, the preamble is 9 Bytes long, and the amount of data to send is now up to 290 Bytes.

3. Include Guard Time
   Guard time is the amount of time allowed for one CM transmitter to ramp down before the next CM is allowed to ramp up its transmitter to send data. The guard time is designated as a number of modulation symbols (e.g., QPSK or 16 QAM) and from the table for a short data grant, the guard time works out to be 2 Bytes. The amount of data to be sent is now up to 292 Bytes.

4. Map to minislots
   Minislots are the unit of bandwidth used on the return path. A Minislot is a certain number of Bytes long, and the return path bandwidth is divided up into these minislots as shown in Figure 1.

   One of the key steps in CM registration is synchronizing time between the CMTS and all the CMs such that everyone knows when one minislot ends and the next one begins. The CMTS indicates to a CM when it can transmit by instructing that CM which minislots it can use on that return path.

   As stated previously, minislots are generally one of 8, 16, 32, or 64 Bytes long. There are a number of parameters that go into calculating the size of a minislot, but once calculated, that size is fixed and used by all CMs.

   When a CM has data to send, it must decide how many minislots are needed to carry that data. In the example so far, the CM has 292 Bytes to send. Assuming a 32 Byte minislot, the CM needs 9.125 minislots to send that data. The trouble is, only a whole number of minislots can be requested. The CM has to request 10 minislots meaning there is additional overhead in the form of a "wasted" 0.875^{th} of a minislot that is not used. (A wasted 0.875^{th} of a 32 Byte minislot, 50 times per second, translates to a wasted 11.2 kbps associated with each call).

   There is another slight gotcha to be explained here. The table has a column titled "max minislots" which is the maximum number of minislots that can be requested for a short data grant. From the table for a short data grant, a maximum of 6 minislots can be requested but the CM needs 10. Not a problem, the CM re-calculates using the physical layer information for a long data grant (the upper layer protocol overhead

remains the same). This is left as an exercise for the reader but the numbers are as follows:

Voice sample = 160 Bytes
Upper layer protocol = 72 Bytes
Padding last codeword = 4 Bytes
FEC Parity = 32 Bytes
Preamble = 10 Bytes
Guard Time = 2 Bytes

Total = 280 Bytes

For a 32 Byte minislot, the CM has to request 9 minislots. Doing the math, this works out to using 115.2 kbps on the return path to carry a 64 kbps voice sample. The math is shown below.

$$\frac{9 \text{ minislots}}{\text{sample}} \times \frac{50 \text{ samples}}{\text{second}} \times \frac{32 \text{ bytes}}{\text{minislot}} \times \frac{8 \text{ bits}}{\text{byte}} = 115.2 \text{ Kbps}$$

The CM needs 9 minislots, each of which are 32 Bytes long, at a rate of 50 times a second.

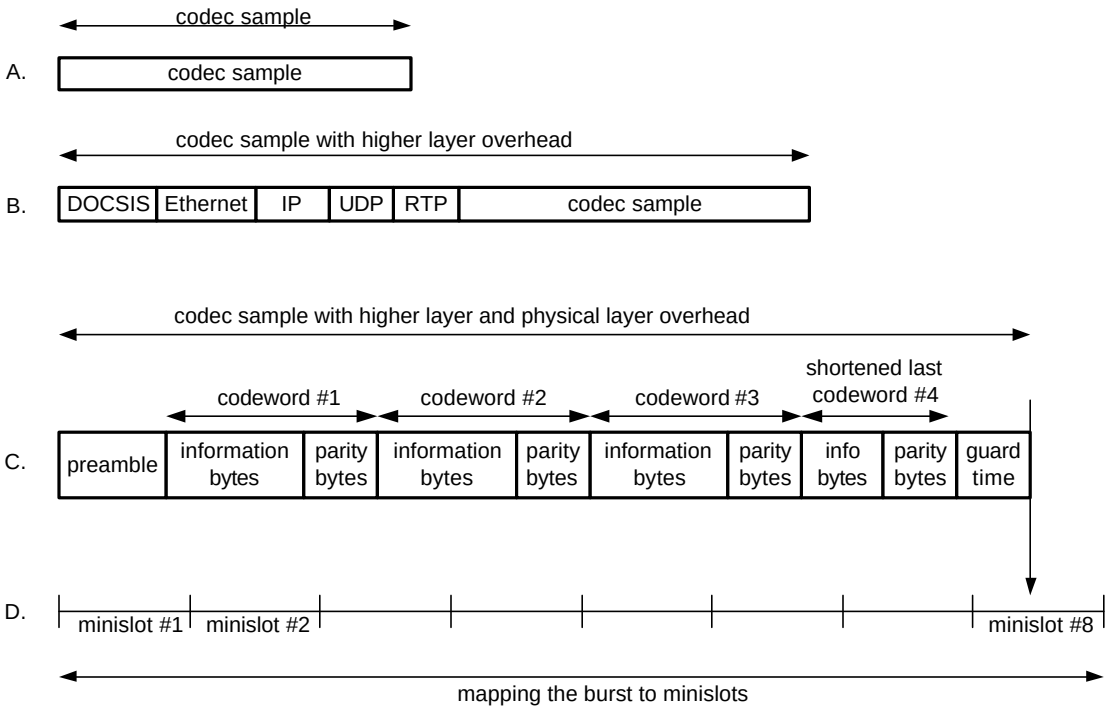

Figure 6

Figure 6 reviews the 4 steps used to determine the amount of return path bandwidth needed for a voice sample.

In step A, there is simply the CODEC sample. In step B, the upper layer protocol over head is added. In step C the DOCSIS physical layer overhead is added. Finally in step D, the sample is mapped to minislots. With each step, the relative amount of bandwidth needed to transmit the sample grows, as is shown by the arrows accompanying each step. To have the most efficient use of return path bandwidth for voice, it can be seen that several parameters need to be tuned.

4.2.2.3 Changing the CODEC Sample Rate

This is an interesting example because it shows that just by changing the CODEC sample rate from 20 milliseconds to 10 milliseconds, the amount of return path bandwidth needed is almost triple the 64 kbps CODEC output.

At 10 milliseconds, the math works like this (hint: a long data grant is needed and shortened last codeword is in effect as there is only one codeword):

CODEC sample = 80 Bytes (half of before due to change in sample time)
Upper layer protocol = 72 Bytes (no change from before)
FEC parity = 16 Bytes (only 1 FEC codeword)
Preamble = 10 Bytes
Guard Time = 2 Bytes

Total = 180 Bytes

When using a 32 Byte minislot, the CM should only have to request 6 minislots. This is another gotcha because if the CM requests 6 minislots, the CMTS will assume the CM is using the physical layer parameters for the short data grant. If this occurred, the CMTS would not recover the data because it would assume different FEC parameters and the received bits would look like garbage. As a result, the CM must request 7 minislots to alert the CMTS to use the physical layer parameters associated with the long data grant even though none of that last minislot (and, due to round off, part of the next to last minislot) is used to carry data. Wasted bandwidth because the Physical layer parameters are not tuned.

Now the bandwidth associated with the call is:

$$\frac{7 \text{ minislots}}{\text{sample}} \times \frac{100 \text{ samples}}{\text{second}} \times \frac{32 \text{ bytes}}{\text{minislot}} \times \frac{8 \text{ bits}}{\text{byte}} = 179.2 \text{ Kbps}$$

Note since the CODEC is sampling at 10 millisecond intervals, there will be 100 samples per second. Under these conditions, the 64 kbps stream out of the CODEC becomes almost 180 kbps on the return path. Such are the vagaries of bandwidth used by a VoIP

call on the return path. Engineering can conserve bandwidth and increase the number of subscribers that can be supported.

Section 5 discusses various choices an operator can make to move beyond default settings on the CMTS and MTA and thereby conserve bandwidth on the return path. Conserving return path bandwidth translates directly to supporting more VoIP users as will be shown in the next section.

4.3 Using an Erlang Calculator

Erlang theory is a lot of math, but the practical usage is pretty easy. Erlang calculations revolve around three quantities. The operator chooses any 2 and uses a calculator to solve for the third one. The three quantities are:

Number of Circuits (N) – This is the number of "circuits" the operator chooses to make available on the return path. These are of course virtual circuits and the number depends on both how much bandwidth the operator chooses to allocate on that return path for VoIP calls and how much bandwidth each call uses. As shown in the previous section, under certain conditions a call using a G.711 CODEC can use 115.2 kbps on the return path. If a 5 Mbps return path is used and the operator allocates 40% of that to voice, a total of 2 Mbps is available for voice. This amount of bandwidth would provide 17 "circuits" (2 Mbps divided by 115.2 kbps) that are available to the VoIP users of that return path.

Blocking Percentage (B) – This is a grade of service that the operator chooses and reflects, on average, the percentage of phone calls that can be blocked due to running out of circuits. There are a finite number of circuits available. Continuing the example above, this number reflects the possibility that all 17 circuits are in use and a 18^{th} call is originated which would be blocked because no more VoIP bandwidth is available on the return path. A reasonable blocking percentage is 0.01, meaning its acceptable for 1 in 100 calls to be blocked due to no circuit available. A better grade of service is 0.001, which means 1 in 1000 calls can be blocked. The lower the blocking percentage, the more circuits that are needed to support a particular call load.

Offered Call Load (A) – This is the number that will be calculated and tells how many VoIP telephones can be supported on that return path. The unit on this number is Erlangs. The North American telephone industry converts this number into Centum Call Seconds (CCS), where a CCS is equal to 100 seconds of

active call time. The conversion is 1 Erlang = 36 CCS. On average, a residential phone line can be weighted at 4.8 CCS, though numbers will vary.

There are several web-based Erlang calculators available, as well as books complete with tables of Erlang information. The following link was used for the remainder of this example:

http://mmc.et.tudelft.nl/~frits/Erlang.htm

For 17 circuits on the return path and a blocking probability of 0.01, the offered call load can be 9.65 Erlangs or 347.4 CCS. At an average residential line weighting of 4.8 CCS, this return path could support about 72 telephones. Not 72 VoIP households, but 72 VoIP telephones.

Taking the example one step further, this return path (with this CODEC, at this sample rate, with these physical layer parameters, etc.) can support 72 telephones with a probability of 1 call out of 100 being blocked. If this return path is shared over two 500 HHP homes (a total of 1000 HHP), that means the operator could sell 72 telephones into those 1000 HHP. Is this enough to support the VoIP business model? Each operator has to do their own calculations. Proper engineering can raise that number and will help the business case.

If the grade of telephone service is made better by lowering the blocking percentage to 1 call out of 1000 (0.001) calls, the number of telephones that can be supported drops to about 55 in the same number of HHP. The subscribers get better service, but fewer telephones can be supported.

The next section will present methods to increase the number of circuits to be available on the return path. This may be the key number the operator has to tinker with and powerful spreadsheets are needed to view all the possible combinations.

5 Network Engineering Considerations

The previous section showed examples using default configurations generally available on DOCSIS and PacketCable equipment. This section describes strategies to move beyond the default configurations in order to have the return path support more VoIP customers.

The previous section also showed that what was supposed to be a 64 kbps VoIP call could use up to three times that amount of bandwidth on the return path. There is only so much bandwidth available on the return path and that translates directly, using Erlang tables, to how many VoIP users can be supported. Strategies to lower the amount of bandwidth needed for a VoIP call are described below.

5.1 Low Bit Rate CODEC

A G.711 CODEC has been used for examples in this paper. This is a widely available CODEC that has a very positive non-technical attribute; there is no royalty fee associated with its use. On the other hand, it is considered a high bandwidth voice CODEC because it generates 64 kbps.

There are other CODECs available that use lesser amounts of bandwidth, on the order 12 kbps to 16 kbps. The same protocol overheads will have to be dealt with, but at least the CODEC payload will be lower. This alone will increase the relative number of circuits available on the return path. However, most of these "compressed" CODECs, as they are known, have royalty fees associated with them.

In a PacketCable voice system, the choice of CODEC is up to the operator, although that CODEC has to be available in the MTA. PacketCable only requires the G.711 CODEC, with two other compressed CODECs being optional.

To support the PSTN interconnect, the Media Gateway (MG) must also support the compressed CODEC that is used in the MTA. This is a system issue for the operator to consider.

5.2 Payload Header Suppression (PHS)

In Payload Header Suppression, a repetitive portion of the protocol overhead is suppressed by the CM and restored by the CMTS (in the return path direction, visa versa in the forward path).

PHS was intended for use with UGS. PHS always suppresses the same number of Bytes in each frame and the savings can be substantial.

In an earlier section it was shown that the upper layer protocol overhead was a total of 72 Bytes. Table 4 shows that PHS can save 42 Bytes per VoIP packet, reducing the upper layer protocol overhead to 30 Bytes:

	Bytes before PHS	Bytes suppressible with PHS	Bytes after PHS
RTP	12	0	12
UDP	8	8	0
IP	20	20	0
Ethernet	18	14	4
DOCSIS MAC	14	0	14
Total	72	42	30

Table 4

PHS is required in DOCSIS 1.1 CMTSes and CMs, however, its use by an eMTA is optional. In other words, PHS does not come automatically with DQoS, but will be used if requested by the eMTA. An operator could consider working with eMTA suppliers on the topic of PHS support.

5.3 Reduced Physical Layer Overhead

It is possible to tune the Physical layer parameters to reduce overhead, however, there is no such thing as a free lunch. For instance, the amount of FEC could be reduced, however, this makes the data more susceptible to RF impairments on the cable plant.

The preamble could be shortened, but if made too short the CMTS receiver may not be trained sufficiently to recover the data.

The minislot size could be made small, but this may place a burden on the CMTS because it now has to track more of them.

There are numerous strategies possible here that rely on both a big spreadsheet and real word data gathering. Operators could consider studying these options with their CMTS suppliers.

5.4 Unsolicited Grant Service with Activity Detection

Unsolicited Grant Service with Activity Detection (UGS/AD) is a variation on the basic UGS service that is used by PacketCable on the DOCSIS connection. UGS/AD is designed for UGS flows that may become inactive for substantial portions of time, i.e., tens of milliseconds or more, such as VoIP CODECs that support silence suppression.

UGS/AD provides UGS only when the call is active. When the call is inactive, no packets are generated by the eMTA. The CMTS periodically checks with the CM and restarts UGS when the CM has data to send. Using UGS/AD, bandwidth on the return path is only used when that end of the connection is talking, as opposed to UGS, which uses bandwidth as long as the call is connected. Given that a telephone conversation is on average half silence, there is the potential here to conserve a lot of bandwidth.

Silence suppression is a characteristic of the CODEC and hence the eMTA. Both implementing a silence suppression CODEC and having the eMTA do the correct UGS/AD signaling with the CM/CMTS is a vendor differentiator that could be explored by the operator.

5.5 PacketCable Interacting with DOCSIS 1.0 CMs

DOCSIS 1.1 CMs implement a feature called “fragmentation” that allows the CM to break up a data frame into several smaller chunks and send those chunks at different times. In DOCSIS 1.0, the CM had to send all the data it had, it could not fragment it.

Why fragment? Consider a DOCSIS upstream that has several active VoIP calls in progress. The CMTS must schedule transmit opportunities for each eMTA at set intervals, the example used most often in this paper is every 20 milliseconds.

If a CM has a maximum length IP packet to send (e.g., 1500 Bytes), a certain amount of time is needed to send it. If that amount of time used by one CM means that an eMTA on that same return path has to delay a voice sample, then jitter and latency are introduced into that voice call.

Notice how this section is titled how PacketCable affects DOCSIS 1.0 CMs, because this is one way a CMTS supplier could implement PacketCable. It's the CMTS that tells the CMs when to transmit. If the CMTS cannot fit a maximum sized IP packet in with all the VoIP samples it needs to schedule, the CMTS does not have to grant an opportunity for that long packet to be sent and it may just wind up being dropped by that CM. If the CM supports fragmentation, the CMTS can instruct that CM to send only a portion of that large packet at a time, making it easier to meet commitments for VoIP QoS.

DOCSIS 1.0 CMs cannot fragment; therefore, there is the risk that when mixed on a return path with PacketCable eMTAs (DOCSIS 1.1 CMs), large data packets from the 1.0 CM may interfere with the VoIP traffic. Or that large data packet just may never get an opportunity to be sent. The choice is an implementation option for the CMTS.

The risk of this occurring is reduced by using higher speed return paths. The DOCSIS 1.1 return path can operate at any of 5 speeds. On a "slower" DOCSIS return path (i.e., 1.28 Mbps) it takes about 10 milliseconds to send a maximum length IP packet. Clearly if VoIP samples are due every 20 milliseconds, a CMTS could not schedule two long packets back to back without interfering with VoIP QoS. On a "faster" DOCSIS return path (i.e., 5 Mbps), it takes about 2.5 milliseconds to send a maximum length IP packet. Even with a 20 millisecond CODEC sampling rate and several active VoIP calls, it's easier to fit in a few maximum length IP packets with the VoIP packets on the higher speed return path.

5.6 DOCSIS 2.0 UGS data grant

Both DOCSIS 1.0 and 1.1 support short and long data grants. The reason there is both a short and long data grant is to allow "tuning" the physical layer parameters for optimum efficiency based on the amount of data to be sent. One set of parameters is used for "short" amounts of data and another set of parameters is used for "long" amounts of data.

When DOCSIS 1.0 was designed, the predominant traffic on the return path was 64 Byte TCP ACK packets. As recent as 2001, about 40% of return path traffic was 64 Byte TCP ACKs. It was intended that the short data grant be tuned for this packet size and the long data grant used for all other packet sizes. Tuning the short data grant for VoIP packets does not necessarily make sense because that may make for lower efficiency when sending the 64 Byte packets. Again, a good spreadsheet is needed.

PacketCable VoIP uses the DOCSIS Unsolicited Grant Service (UGS). In DOCSIS 2.0, a grant type was added specifically for UGS to allow physical layer parameters to be tuned for VoIP packets (N.B., this works best when only one set of CODEC parameters is in use).

That is, in DOCSIS 2.0 there are three types of bursts, a short data grant, a long data grant, and a UGS data grant. The short data grant should be tuned for 64 Byte TCP ACKs, the UGS data grant tuned for VoIP packets, and the long data grant is used for everything else.

This works as long as the only UGS traffic on the return path is associated with a particular CODEC. If other types of UGS traffic are present (that have different grant sizes), less than optimum bandwidth usage might result.

5.7 CMTS "Knobs" Desirable for PacketCable

This section describes CMTS tools that are not required by either the DOCSIS 1.1 or PacketCable voice specifications, but that are desirable for deploying a VoIP service. Because they are not in the specifications, the implementation of these tools will vary by supplier. This is not an exhaustive list, but provides leads where operators may choose to talk with suppliers

– Bandwidth partition between voice and data
  If both voice and data are run on the same return path, an operator can choose to reserve a certain amount of bandwidth for both services. Generally this would be implemented as an amount of bandwidth available for UGS (i.e., VoIP) flows with the remainder available for data. The amount of bandwidth reserved for VoIP is not necessarily a guarantee, that is, if no voice calls are in progress then data traffic could use that capacity for web surfing, etc. On the other hand, a certain amount of bandwidth could be set aside to ensure that web surfing and email still work on Mothers Day when there are a lot of telephone calls. Tools are needed to manage the bandwidth when VoIP and data are mixed.

– Reporting when voice/data boundary met
  When VoIP traffic bumps into the partition between voice and data, an event or report should be generated to let the operator know there may be an issue with bandwidth engineering.

– Choice between jitter in voice flow or letting a best effort packet thru
  If in order to allow a CM send a maximum length IP packet, should jitter and latency be added to a VoIP call ? This is an implementation decision. Generally VoIP bandwidth is considered premium and it's the data traffic that should be delayed. But the operator could have this decision.

There are several other knobs that could be provided by the CMTS. As these discussions are outside the scope of both the DOCSIS and PacketCable Voice specifications, they are left to the operator and supplier to discuss.

6.0 Summary

This paper contains engineering considerations for adding PacketCable VoIP to a DOCSIS network. Efficient use of return path bandwidth will allow the network to support more VoIP calls and as a result, larger service penetration can be realized. Default CMTS and eMTA settings may not make the most efficient use of bandwidth; there is room for the operator to optimize settings.

References

1. [DOCSIS 1.0] SP-RFI-C01-011106, DOCSIS v1.0 Radio Frequency Interface Specification, http://www.cablemodem.com/specifications, November 2001.

2. [DOCSIS 1.1], SP-RFIv1.1-I09-020830, DOCSIS v1.1 Radio Frequency Interface Specification, http://www.cablemodem.com/specifications, August 2002.

3. [DOCSIS 2.0], SP-RFIv2.0-I03-021219, DOCSIS v2.0 Radio Frequency Interface Specification, http://www.cablemodem.com/specifications, December 2002.

4. [PacketCable 1.x], http://www.packetcable.com/specifications.

5. [DQoS], PKT-SP-DQOS-I05-021127, PacketCable Dynamic Quality of Service Specification, November 2002.

6. Ferguson, P. and G. Huston, Quality of Service; Delivering QoS on the Internet and in Corporate Networks, Wiley Computer Publishing, 1998.

7. Evans, D., Digital Telephony Over Cable, Addison-Wesley, 2001.

8. Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, RFC 1889, http://www.ietf.org, January 1996.

9. Postel, J., User Datagram Protocol, RFC 768, http://www.ietf.org, August 1980.

10. Postel, J., Internet Protocol, RFC 791, http://www.ietf.org, September 1981.

11. IEEE, Ethernet, "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", ANSI/IEEE Std 802.3-1985.