CHAPTER 12

DQoS Architecture and Framework

A cable operator's ability to offer quality of service (QoS) treatment for multimedia applications gives that cable operator a significant edge over other service providers that have no control over the Multiple Systems Operator (MSO)'s transport network. This ability allows the cable MSO to be more than just a raw bit provider. Not only can an MSO offer a service like telephony that makes use of their DQoS resources, but they can also provide dynamic quality of service (DQoS) resources for services operated by third-party providers. Whether operated in-house or externally, these QoS-enabled services can open up new sources of revenue. The concepts presented in this chapter, as well as in Chapter 13, "Analyzing, Implementing, and Troubleshooting DQoS," and Chapter 14, "Multimedia Applications," detail how DQoS is provided and how these resources are accounted for.

Throughout this book, you have seen numerous discussions on the importance of a QoS mechanism to manage Data-Over-Cable Service Interface Specification (DOCSIS) resources. In Chapter 1, "PacketCable Overview," you learned that this QoS requirement ensures that voice calls are of sufficient quality and provides a mechanism to authorize and control DOCSIS resources—an essential part of PacketCable. Chapter 2, "PacketCable Functional Components," covered the multimedia terminal adapter (MTA), cable modem (CM), cable modem termination system (CMTS), call management server (CMS), and Record Keeping Server (RKS) PacketCable components, all of which contribute to providing this QoS. Chapter 7, "NCS," covered some parameters within the Network-based Call Signaling (NCS) protocol that communicate DQoS information between the CMS and MTA. Finally, in Chapter 10, "Audio CODECs," and Chapter 11, "RTP and RTCP," you learned how voice is transmitted in a PacketCable VoIP network.

This chapter covers DQoS in greater detail, including how this functionality is provided and managed and what the protocols are that make it work. The DOCSIS 1.1 protocol, the NCS protocol, and the Common Open Policy Service (COPS) protocol all contribute to PacketCable DQoS functionality. Chapter 13 builds on the technical information covered here by examining the implementation of these technologies in providing quality of service for PacketCable telephony. Chapter 14 goes into detail on the theory and technical aspects behind PacketCable Multimedia (PCMM). The PCMM specifications detail a generalized framework for providing dynamic QoS for any type of multimedia application. The topics covered in Chapter 12 are the foundation for many of the topics covered in Chapter 14.

DQoS Overview

When it comes to QoS for a PacketCable call, the call is separated into three pieces:

- The access network between the originating subscriber and the backbone
- The access network between the backbone and the terminating subscriber
- The backbone itself

QoS across the backbone can be provided using Differentiated Services Code Point (DSCP) marking or Multiprotocol Label Switching (MPLS). How backbone QoS is provided is not explicitly defined by PacketCable; however, Chapter 16, "PacketCable Network Design Considerations," introduces some QoS mechanisms that can be used.

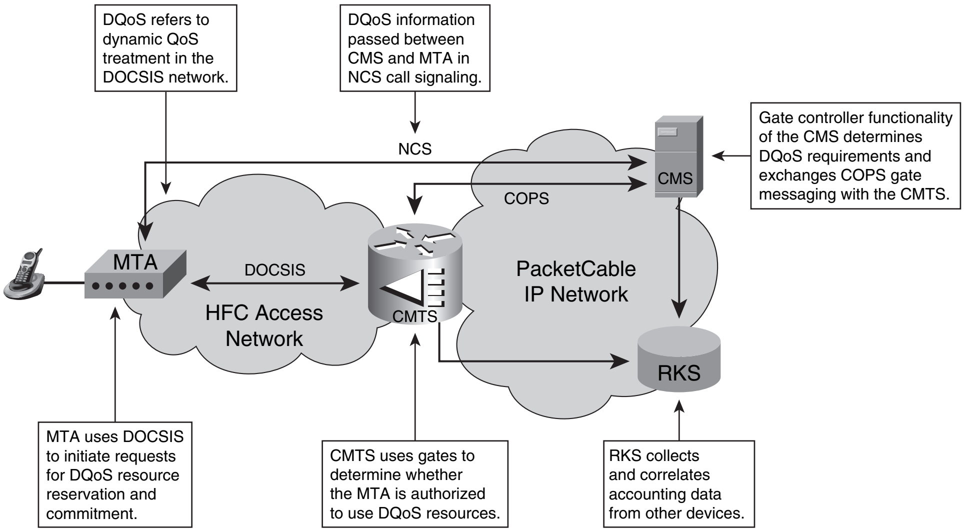
If the originating or terminating subscriber is a PacketCable customer, the access network contains the DOCSIS network between the CMTS and the MTA. QoS across the DOCSIS access network is provided via PacketCable DQoS, as described in this chapter.

NOTE QoS discussed in this chapter is only for bearer packets; QoS for signaling packets (NCS) is also an important consideration and is covered in Chapter 16.

It is no secret that DOCSIS resources are a valuable commodity; typically, hundreds of users are competing for this bandwidth across a DOCSIS MAC domain. Remember, a DOCSIS MAC domain is the collection of users connected to a common set of downstream and upstream DOCSIS channels. Normally, high-speed data users and PacketCable customers utilize this same bandwidth pipe. PacketCable DQoS ensures that voice customers are given preferential treatment in receiving the amount of bandwidth and at the intervals needed for quality voice communication.

Figure 12-1 illustrates the PacketCable components and protocols involved in providing DQoS functionality.

Figure 12-1 *DQoS Architecture*


DQoS refers to dynamic QoS treatment in the DOCSIS network.
DQoS information passed between CMS and MTA in NCS call signaling.
NCS
Gate controller functionality of the CMS determines DQoS requirements and exchanges COPS gate messaging with the CMTS.
CMS
COPS
MTA
DOCSIS
HFC Access Network
CMTS
PacketCable IP Network
RKS
MTA uses DOCSIS to initiate requests for DQoS resource reservation and commitment.
CMTS uses gates to determine whether the MTA is authorized to use DQoS resources.
RKS collects and correlates accounting data from other devices.

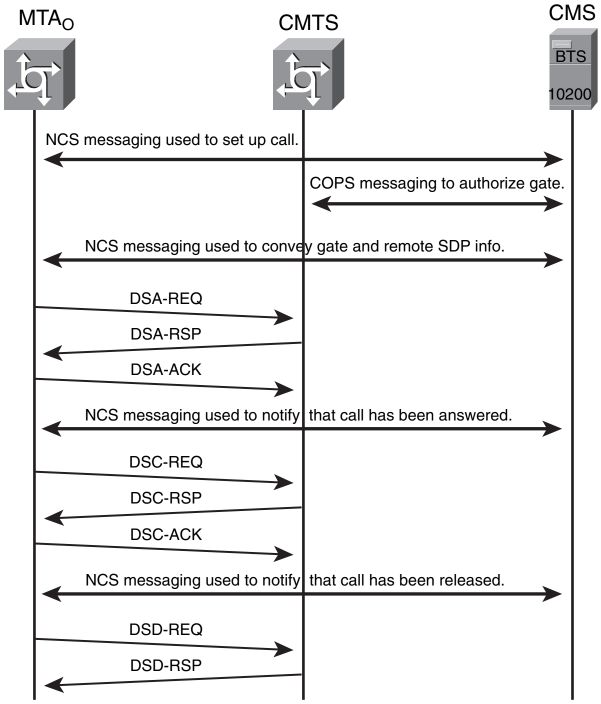
As you can see, four PacketCable components are involved in providing DQoS functionality:

- **Embedded MTA (eMTA)**—DQoS information is exchanged between an eMTA and the CMS using the NCS protocol. An embedded MTA internally signals its cable modem component to request QoS using the DOCSIS protocol. This is done through the use of DOCSIS 1.1 dynamic service flow messages exchanged between the CMTS and eMTA.
- **CMTS**—DOCSIS bandwidth is authorized, reserved, and committed by the CMTS. Access to this premium bandwidth is identified at the CMTS by a DQoS gate. What a gate is and how it works is covered later—for now, think of a gate as the key to premium DOCSIS resources. Because the CMTS is where DOCSIS resources are controlled, it is also referred to as the Policy Enforcement Point (PEP) in the DQoS architecture.
- **CMS**—Gates are created on the CMTS under the direction of the CMS. Thus, the CMS also functions as a gate controller. The CMS directs the CMTS to set up DQoS gates and then relays this information to the MTA using the NCS protocol. Because the CMS is controlling the manipulation of DQoS gates, it is referred to as the Policy Decision Point (PDP) in the DQoS architecture.
- **RKS**—The value of premium DQoS resources in a DOCSIS network has already been discussed. Because these resources are so valuable, they must be accurately accounted for. The CMTS uses event messaging to record DQoS usage per subscriber

and passes this information to an RKS for accounting purposes. The identity of the RKS(s) is relayed to the CMTS by the CMS. Keep in mind that the RKS has nothing do with providing DQoS functionality; it simply records DQoS usage.

Figure 12-2 shows a high-level overview of the messaging between the MTA, CMS, and CMTS used in providing DQoS functionality.

Figure 12-2 *DQoS Messaging*


MTA_O
CMTS
CMS
BTS
10200
NCS messaging used to set up call.
COPS messaging to authorize gate.
NCS messaging used to convey gate and remote SDP info.
DSA-REQ
DSA-RSP
DSA-ACK
NCS messaging used to notify that call has been answered.
DSC-REQ
DSC-RSP
DSC-ACK
NCS messaging used to notify that call has been released.
DSD-REQ
DSD-RSP

The details of the COPS messaging between the CMS and CMTS components and the DOCSIS messaging between the MTA and CMTS components are the focus for much of this chapter.

Gates

DQoS gates control access to premium DOCSIS resources between an MTA and a CMTS. A 4-byte gate identifier assigned by the CMTS references these gates. For PacketCable 1.x the gate ID typically refers to two gates—one for the upstream direction and one for the downstream direction. Thus, a single gate ID references the DQoS resources for a single leg in an NCS call.

Although the gate ID can reference a bidirectional flow, the gates themselves are unidirectional. In fact, in Chapter 14 you'll learn that PCMM upstream and downstream gates are generally created separately and have separate gate IDs.

Gates are allocated and scheduled on the CMTS under the direction of the CMS. A gate consists of the following:

- Classification components used to identify packets permitted to use the premium resources. The classifiers consist of a direction (upstream or downstream), the IP protocol, the source and destination IP addresses, and port numbers.
- Policing components ensuring only resources allocated for the gate are used. These components describe the traffic needing DQoS resources.

Other information is often associated with a gate. An example is accounting components ensuring resources are accurately billed. This information includes a billing identifier and the location of the RKS(s).

Gate States

A PacketCable 1.x gate transitions through four states or phases:

- Allocated
- Authorized
- Reserved
- Committed

Before an MTA can request DQoS resources, the allocation and authorization phases need to complete. In the allocation phase the CMS tells the CMTS (using the COPS protocol) to allocate a pair of gates for a particular MTA. The messaging here includes the maximum number of gates the MTA is allowed to have at once; the CMTS checks to see whether adding two additional gates violates this maximum. Assuming the MTA is permitted to receive two additional gates, the CMTS assigns a 4-byte gate identifier for the newly allocated gate pair and returns this value to the CMS.

After the CMS learns what type of resources the MTA requires and the classification parameters needed to identify packets permitted to use those resources, it sends this information to the CMTS in a COPS "Gate Set" message. The CMTS then determines whether it is capable of providing these resources for the MTA. The CMTS responds to the CMS with an indication of whether the authorization was successful. At this point the gate is in the *authorized state*. Remember, no resources have actually been reserved at this point.

The allocation and authorization phases can occur in separate steps, or they can occur simultaneously; this is up to the CMS vendor implementation. If separate steps are used, a *Gate Alloc* message is first sent from the CMS to the CMTS. If the CMTS was able to allocate the gates, a *Gate Alloc Ack* message is sent to the CMS containing the gate ID. If the CMTS was unable to allocate the gates, a *Gate Alloc Err* message, indicating the reason for the error, is sent to the CMS instead.

In the authorization phase the CMS sends a *Gate Set* message to the CMTS containing the DQoS resource and classification parameters. If the CMTS is able to authorize the request a *Gate Set Ack* message is returned; if the CMTS was unable to authorize the request a *Gate Set Error* message is returned, indicating the reason for the error. In a single-step implementation only the *Gate Set* message exchange occurs; when the CMTS receives the *Gate Set*, it does the allocation check and gate ID assignment in addition to the authorization check. The 4.x and later versions of the BTS 10200 use a single-step implementation.

After resources are authorized, the MTA requests the CMTS (using DOCSIS 1.1) for all or some of these resources to be reserved for use. The MTA is directed to reserve resources in NCS messaging from the CMS. This usually occurs right before the CMS requests ring tone and ring back tone be provided to the call parties. The reserved resources envelope is always less than or equal to the authorized resources envelope.

In the reservation phase, resources are held for an MTA, ensuring they are available when needed. Until they are needed, these resources can be used for ordinary best effort data service. Remember, reserved resources cannot be used by other DQoS customers because the same resources cannot be reserved multiple times; they can be used only by Best Effort services. This protects against the possibility of a MTA attempting to commit DQoS resources that aren't available because they were assigned to other users.

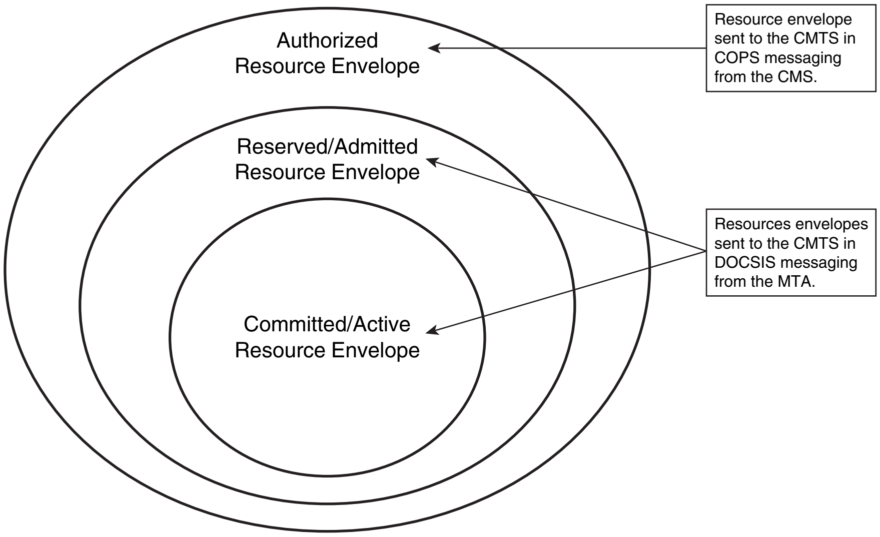
In the commit phase, resources are made available for use by the MTA, and no one else can use them. This usually happens at the point the phone call is answered. As with the reservation phase, NCS messaging from the CMS tells the MTA to commit resources, and DOCSIS 1.1 messaging signals the CMTS to commit resources.

Usually, DQoS resources are provided to an MTA in two separate phases, the reservation phase and the commitment phase. This dual-phase model ensures that resources are available when needed and does not tie up resources that are not needed. For example, you don't need to commit resources for a phone call if the terminating party isn't there to answer the phone. On the other hand, if quality resources cannot be provided, there is no point in allowing the call to go through. Another benefit of the dual-phase approach is that the billing of resources doesn't begin until the commit phase, where the voice call is actually active.

NOTE The CMS might decide to reserve and commit resources at the same time, but this is generally discouraged. One situation in which this might occur is for OFFNET calls where ring-back tone is supplied from the PSTN. In this case, DQoS resources in the direction from the PSTN to the MTA might be committed immediately to let the ring back signal use DQoS resources.

Keep in mind the amount of resources committed can be different from the amount reserved, and the amount of resources reserved can be different from the amount of resources authorized, although in most PacketCable 1.x deployments they are typically the same. The committed resources envelope is always less than or equal to the reserved resources envelope, which is always less than or equal to the authorized resources envelope. Figure 12-3 illustrates the relationship between these resource categories.

Figure 12-3 *DQoS Resources*


Authorized
Resource Envelope
Reserved/Admitted
Resource Envelope
Committed/Active
Resource Envelope
Resource envelope sent to the CMTS in COPS messaging from the CMS.
Resources envelopes sent to the CMTS in DOCSIS messaging from the MTA.

As you can tell in Figure 12-3, reserved resources are also referred to as *admitted resources*, and committed resources are also referred to as *active resources*. The *admitted* and *active* terms are used in the DOCSIS specification to refer to the status of DOCSIS service flows. A DOCSIS service flow that is statically defined in the cable modem configuration file is classified as *provisioned*. A DOCSIS service flow (either static or dynamic) that the CMTS accepts and is able to service is classified as *admitted*. And a DOCSIS service flow (either static or dynamic) that the CMTS allows traffic to be sent on is classified as *active*. As you can see, the DOCSIS concepts of *admitted* and *active* service flows map directly to the PacketCable concepts of *reserved* and *committed* DQoS resources.

Earlier it was mentioned that in PacketCable 1.x the *authorized*, *reserved*, and *committed* resource envelopes are typically equal. That’s because a majority of today’s deployments use only the G.711 CODEC; consequently, the CMS knows from the beginning the CODEC the call is going to use so it can authorize the exact amount of DQoS resources necessary to support it. Even if Voice Band Data such as fax is detected and the MTA doesn’t support a relay mechanism, no CODEC change is required because G.711 supports Voice Band Data in-band. Conversely, if CODEC negotiation results in multiple possible

CODECs, the CMS does not know what CODEC is going to be used for the call. So the CMS generates an authorization envelope big enough to handle any of the possible CODECs; this is also known as the least upper bound (LUB) constraint. For example, if the CODEC list contains G.711 and G.729E, the CMS authorizes enough resources to support G.711 even though the actual call might end up using G.729E, which has a much smaller bandwidth requirement.

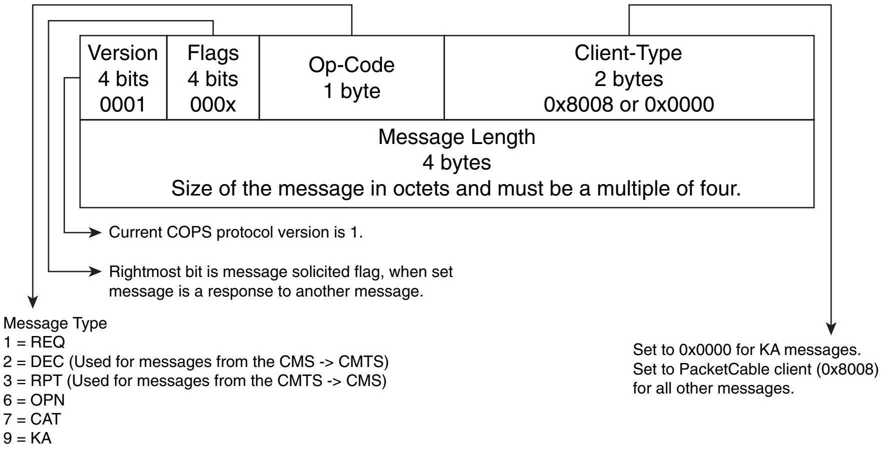
Now, you might be wondering in what situations the reserved and committed resource envelopes would be different. One example is safeguarding for CODEC changes because of Voice Band Data detection. For example, if voice calls are set up to use G.729E, but the MTA wants to be sure a CODEC up-speed to G.711 to support fax transmission gets the necessary resources, it can reserve the G.711 envelope but only commit the G.729E envelope. Then if fax tone is detected requiring a CODEC change, the MTA is assured the process of committing additional resources will be successful.

Another point to remember about reserved and committed resources is that they are dynamic and can be changed during the life of a call. For example, if a CODEC change is necessary to support fax transmission (usually triggered by NCS signaling from the CMS), the MTA sends DOCSIS messaging to the CMTS to change the committed resource envelope (and possibly the reserved as well).

COPS Protocol

DQoS gate information is communicated between the CMS and the CMTS using the Common Open Policy Service (COPS) protocol. COPS is a server/client protocol that runs over TCP, the COPS server is also referred to as the Policy Decision Point (PDP), and the COPS client is also referred to as the Policy Enforcement Point (PEP). For PacketCable 1.x, the CMS/GC is the COPS server/PDP, and the CMTS is the COPS client/PEP. Each COPS message consists of a common header followed by a variable number of objects. Figure 12-4 details the format of this header.

Figure 12-4 *COPS Common Header Format*



Current COPS protocol version is 1.

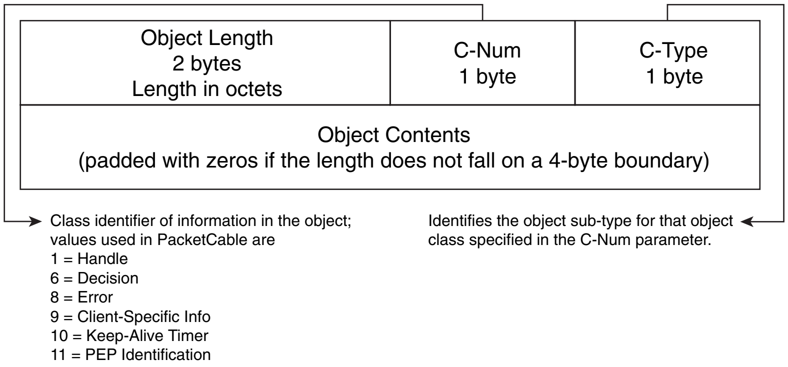
Rightmost bit is message solicited flag, when set message is a response to another message.

Message Type
1 = REQ
2 = DEC (Used for messages from the CMS -> CMTS)
3 = RPT (Used for messages from the CMTS -> CMS)
6 = OPN
7 = CAT
9 = KA

Set to 0x0000 for KA messages.
Set to PacketCable client (0x8008)
for all other messages.

Figure 12-5 illustrates the format for the variable objects.

Figure 12-5 *COPS Object Format*



Class identifier of information in the object; values used in PacketCable are
1 = Handle
6 = Decision
8 = Error
9 = Client-Specific Info
10 = Keep-Alive Timer
11 = PEP Identification

Identifies the object sub-type for that object class specified in the C-Num parameter.

For messages going from the CMS to the CMTS in a Decision Message, the C-Num is 6 (Decision) and the C-Type is 4 (Client-Specific Decision Data) for PacketCable objects.

For messages going from the CMTS to the CMS in a Report-State Message, the C-Num is 9 (Client-Specific Info) and the C-Type is 1 (Signaled Client SI) for PacketCable objects.

COPS Initialization

After the CMTS is configured for PacketCable operation and the CMTS is configured and activated on the CMS, the COPS initialization sequence begins. First, the CMS opens the TCP connection used for COPS on TCP port 2126. Notice that for this connection to be established, the CMS must be configured with the location of the CMTS; however, the CMTS does not need to be configured with the location of the CMS. This is assuming that IP Security (IPsec) is not being used for the COPS connection. If IPsec is being used, both the CMS and CMTS must be explicitly configured to have a security association between them.

NOTE Refer to Chapter 7 for how to configure a CMTS on the Cisco BTS 10200.

After this TCP connection is established, the CMTS identifies itself by sending a COPS Client-Open (message type 6 - OPN) to the CMS. This identification consists of a PEP-ID (Policy Enforcement Point ID), typically the configured FQDN of the CMTS. The CMS responds by sending a Client-Accept (message type 7 - CAT) to the CMTS. In this message is the value of the keepalive timer to be used by the CMTS. The CMTS completes the initialization process by sending a Request (message type 1 - REQ) to the CMTS. This message defines the Handle object used in future messages between the CMS and CMTS. Figure 12-6 illustrates this initialization sequence as well.

Figure 12-6 *COPS Initialization Sequence*

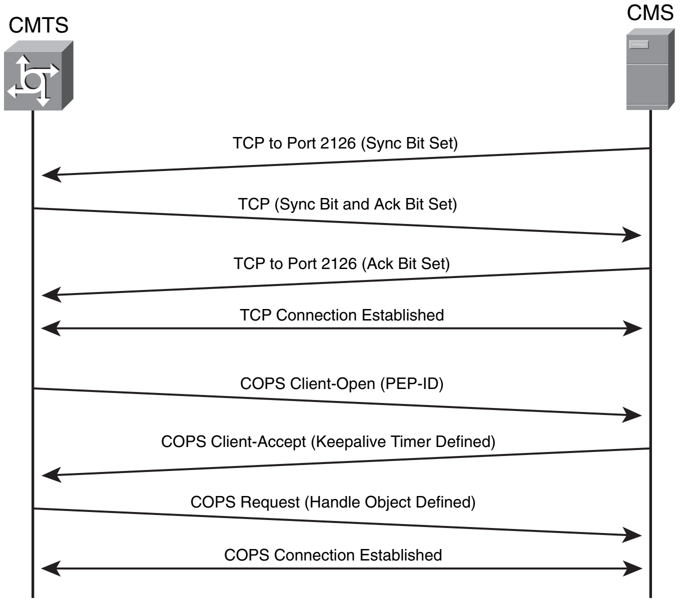
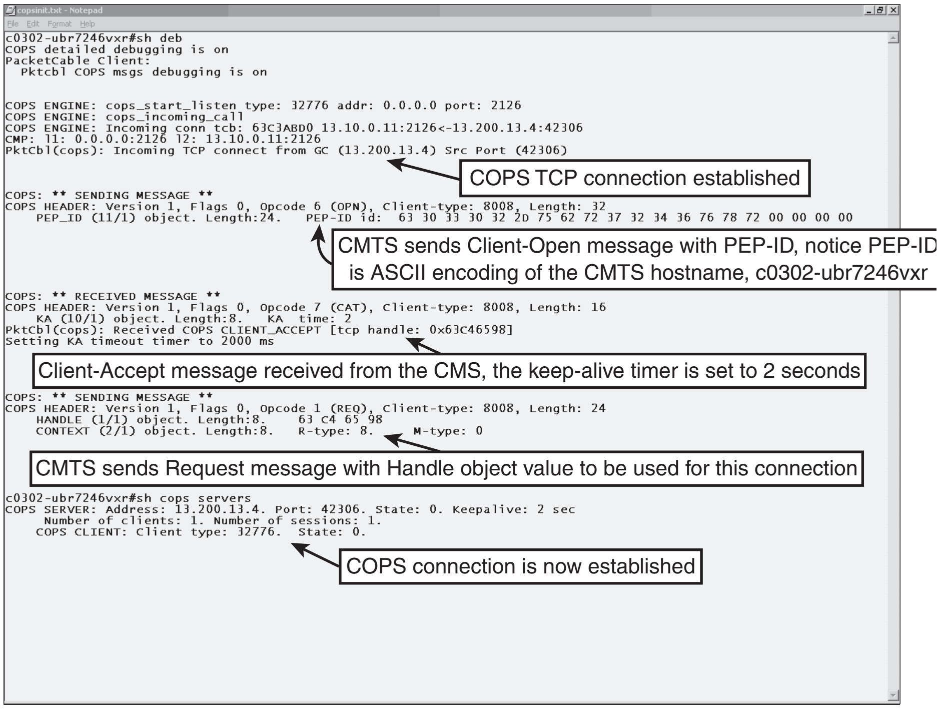

CMTS
CMS
TCP to Port 2126 (Sync Bit Set)
TCP (Sync Bit and Ack Bit Set)
TCP to Port 2126 (Ack Bit Set)
TCP Connection Established
COPS Client-Open (PEP-ID)
COPS Client-Accept (Keepalive Timer Defined)
COPS Request (Handle Object Defined)
COPS Connection Established

Figure 12-7 shows an example of this initialization sequence on a Cisco uBR7246VXR. IOS debugs were enabled to show the messages. Please be very cautious when enabling debugs such as the **debug cops detail** command because it causes quite a bit of output.

Figure 12-7 *COPS Initialization on a Cisco CMTS*



```
c0302-ubr7246vxr#sh deb
COPS detailed debugging is on
PacketCable Client:
  Pktcbl COPS msgs debugging is on


COPS ENGINE: cops_start_listen type: 32776 addr: 0.0.0.0 port: 2126
COPS ENGINE: cops_incoming_call
COPS ENGINE: Incoming conn tcb: 63C3ABD0 13.10.0.11:2126<-13.200.13.4:42306
CMP: l1: 0.0.0.0:2126 l2: 13.10.0.11:2126
PktCbl(cops): Incoming TCP connect from GC (13.200.13.4) Src Port (42306)


COPS: ** SENDING MESSAGE **
COPS HEADER: Version 1, Flags 0, Opcode 6 (OPN), Client-type: 8008, Length: 32
    PEP_ID (11/1) object. Length:24.   PEP-ID id:  63 30 33 30 32 2D 75 62 72 37 32 34 36 76 78 72 00 00 00 00



COPS: ** RECEIVED MESSAGE **
COPS HEADER: Version 1, Flags 0, Opcode 7 (CAT), Client-type: 8008, Length: 16
    KA (10/1) object. Length:8.   KA  time: 2
PktCbl(cops): Received COPS CLIENT_ACCEPT [tcp handle: 0x63C46598]
Setting KA timeout timer to 2000 ms

COPS: ** SENDING MESSAGE **
COPS HEADER: Version 1, Flags 0, Opcode 1 (REQ), Client-type: 8008, Length: 24
    HANDLE (1/1) object. Length:8.    63 C4 65 98
    CONTEXT (2/1) object. Length:8.   R-type: 8.    M-type: 0

c0302-ubr7246vxr#sh cops servers
COPS SERVER: Address: 13.200.13.4. Port: 42306. State: 0. Keepalive: 2 sec
     Number of clients: 1. Number of sessions: 1.
    COPS CLIENT: Client type: 32776.  State: 0.
```

COPS TCP connection established

CMTS sends Client-Open message with PEP-ID, notice PEP-ID is ASCII encoding of the CMTS hostname, c0302-ubr7246vxr

Client-Accept message received from the CMS, the keep-alive timer is set to 2 seconds

CMTS sends Request message with Handle object value to be used for this connection

COPS connection is now established

CMTS to CMS Heartbeats

Keepalive messages (message type 9 - KA) are sent from the CMTS to the CMS periodically. The CMS responds to these keepalive messages by sending a keepalive message back to the CMTS. If the CMS fails to receive a keepalive message from the CMTS by the keepalive timer value, it assumes the connection has been lost. The CMS then attempts to reestablish the TCP connection that if successful causes the CMTS to reinitialize the COPS connection. Similarly, if the CMTS fails to receive the keepalive echo from the CMS, it assumes the connection has been lost and listens for a new TCP connection.

PacketCable Objects in COPS

As shown in Figure 12-5, PacketCable objects are included in either Client Specific Decision Data objects (C-Num 6, C-Type 4) or Signaled Client SI objects (C-Num 9, C-Type 1), depending on the direction of the message. These objects begin with a 2-byte length field followed by another 1-byte object number and 1-byte subtype definition. These are referred to as the S-Num and S-Type, respectively.

PacketCable COPS objects are as follows:

- Transaction Identifier (S-Num=1, S-Type=1)
- Subscriber Identifier (S-Num=2, S-Type=1 for IPv4 and 2 for IPv6)
- Gate Identifier (S-Num=3, S-Type=1)
- Activity Count (S-Num=4, S-Type=1)
- Gate Specification (S-Num=5, S-Type=1)
- Event Generation Info (S-Num=7, S-Type=1)
- PacketCable Error (S-Num=9, S-Type=1)
- Electronic Surveillance Parameters (S-Num=10, S-Type=1)
- PacketCable Reason (S-Num=13, S-Type=1)

The sections that follow cover these objects in greater detail.

Transaction Identifier

This object correlates responses to commands. It consists of a 2-byte transaction identifier followed by the 2-byte gate command type. The gate command type identifies the DQoS message type. Table 12-1 lists the possible values.

Table 12-1 *DQoS Message Types*

Gate Command Type	Gate Command	Message Direction
1	GATE-ALLOC	CMS → CMTS
2	GATE-ALLOC-ACK	CMTS → CMS
3	GATE-ALLOC-ERR	CMTS → CMS
4	GATE-SET	CMS → CMTS
5	GATE-SET-ACK	CMTS → CMS
6	GATE-SET-ERR	CMTS →CMS
7	GATE-INFO	CMS → CMTS
8	GATE-INFO-ACK	CMTS → CMS
9	GATE-INFO-ERR	CMTS → CMS

Table 12-1 *DQoS Message Types (Continued)*

Gate Command Type	Gate Command	Message Direction
10	GATE-DELETE	CMS → CMTS
11	GATE-DELETE-ACK	CMTS → CMS
12	GATE-DELETE-ERR	CMTS → CMS
13	GATE-OPEN	CMTS → CMS
14	GATE-CLOSE	CMTS → CMS

You’ve already learned how the GATE-ALLOC and GATE-SET commands are used in the gate allocation and authorization phases. The GATE-INFO commands are used by the CMS to query information about a gate. The GATE-OPEN command is used by the CMTS to inform the CMS that DQoS resources have been committed. The GATE-CLOSE command is used by the CMTS to inform the CMS that a gate has been deleted because of MTA interaction or inactivity. The GATE-DELETE commands are typically seen only in error scenarios where the CMS needs to tear down a gate.

Subscriber Identifier

The Subscriber ID object is the IP address of the subscriber, used to associate QoS information with a particular subscriber to help prevent denial of service attacks.

Gate Identifier

The Gate ID is the 4-byte gate identifier assigned by the CMTS.

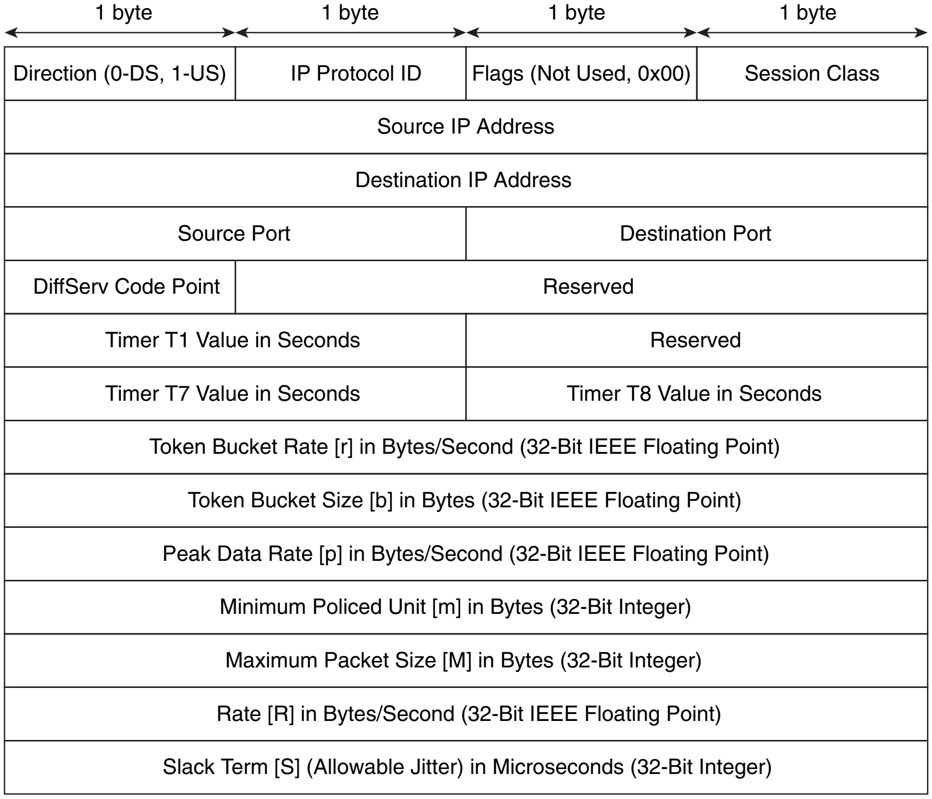
Activity Count

When sent in a GATE-ALLOC or GATE-SET message from the CMS, this field indicates the maximum amount of simultaneous sessions permitted for the subscriber. When sent in a GATE-ALLOC-ACK or GATE-SET-ACK message from the CMTS, this field indicates the number of gates currently assigned to a subscriber. This field is also used to help prevent denial of service attacks.

Gate Specification

This field describes the gate parameters authorized for a particular subscriber. Figure 12-8 illustrates the format for this field.

Figure 12-8 *Gate Specification Format*



<table>
<tr><th>1 byte</th><th>1 byte</th><th>1 byte</th><th>1 byte</th></tr>
<tr><td>Direction (0-DS, 1-US)</td><td>IP Protocol ID</td><td>Flags (Not Used, 0x00)</td><td>Session Class</td></tr>
<tr><td colspan="4">Source IP Address</td></tr>
<tr><td colspan="4">Destination IP Address</td></tr>
<tr><td colspan="2">Source Port</td><td colspan="2">Destination Port</td></tr>
<tr><td>DiffServ Code Point</td><td colspan="3">Reserved</td></tr>
<tr><td colspan="2">Timer T1 Value in Seconds</td><td colspan="2">Reserved</td></tr>
<tr><td colspan="2">Timer T7 Value in Seconds</td><td colspan="2">Timer T8 Value in Seconds</td></tr>
<tr><td colspan="4">Token Bucket Rate [r] in Bytes/Second (32-Bit IEEE Floating Point)</td></tr>
<tr><td colspan="4">Token Bucket Size [b] in Bytes (32-Bit IEEE Floating Point)</td></tr>
<tr><td colspan="4">Peak Data Rate [p] in Bytes/Second (32-Bit IEEE Floating Point)</td></tr>
<tr><td colspan="4">Minimum Policed Unit [m] in Bytes (32-Bit Integer)</td></tr>
<tr><td colspan="4">Maximum Packet Size [M] in Bytes (32-Bit Integer)</td></tr>
<tr><td colspan="4">Rate [R] in Bytes/Second (32-Bit IEEE Floating Point)</td></tr>
<tr><td colspan="4">Slack Term [S] (Allowable Jitter) in Microseconds (32-Bit Integer)</td></tr>
</table>

The Session Class field identifies the gate as belonging to a normal VoIP session (0x01), a high-priority VoIP session (0x02), or an unspecified session (0x00). The high-priority session class could be used for emergency 911 calls so that they receive preferential treatment over ordinary phone calls. For example, if an attempt to reserve resources for a 911 call is made and no bandwidth is available, ordinary calls could be dropped to create the necessary resources for this high-priority session. Another possible implementation is to pre-designate a certain amount of bandwidth explicitly for high-priority calls.

Event Generation Information

This object passes along RKS information to the CMTS. This includes primary and secondary RKS server IP addresses and port numbers, a billing correlation identifier, and the mode in which the CMTS is to send event messages. The CMTS can either send event messages in real time or accumulate them and send them at periodic time intervals. Please refer to Chapter 15, "Event Messaging and Lawful Intercept," for more information on how event messaging is used in PacketCable.

PacketCable Error

This object consists of a 2-byte error code and a 2-byte error sub-code. Table 12-2 lists the currently defined error codes.

Table 12-2 *Error Codes*

Error Code	Definition
1	No gates currently available
2	Unknown Gate ID
3	Illegal session class value
4	Subscriber exceeded gate limit
5	Gate already set
6	Missing required object
7	Invalid object
8	Illegal DSCP field value
127	Other, unspecified error

The error sub-code conveys further information about the error. For example, for error codes 6 and 7 this field contains the S-Num and S-Type values of the object missing or invalid.

Electronic Surveillance Parameters

This object contains all the information needed to support electronic surveillance or lawful intercept. This includes the lawful intercept Delivery Function (DF) IP addresses and port numbers used for both "call detail" and "call content" taps, the call content correlation identifier, the billing correlation identifier, and the flags indicating whether the CMTS should duplicate "call detail" and/or "call content" information. Please refer to Chapter 15 for more information on how electronic surveillance is used in PacketCable.

PacketCable Reason

This object communicates why a gate is being deleted or closed. It consists of a 2-byte reason code and a 2-byte reason sub-code. The reason code identifies the message as a

GATE-DELETE operation (0) or a GATE-CLOSE operation (1). Gate Delete messages are sent from the CMS to the CMTS, whereas Gate Close messages are sent from the CMTS to the CMS. Table 12-3 defines the values of the reason sub-code for a Gate Delete operation.

Table 12-3 *Gate Delete Reason Sub-Codes*

Reason Sub-Code	Definition
0	Normal operation
1	Local gate-coordination not completed
2	Remote gate-coordination not completed
3	Authorization revoked
4	Unexpected GATE-OPEN
5	Local GATE-CLOSE failure
127	Other, unspecified error

Table 12-4 defines the values of the reason sub-code for a Gate Close operation.

Table 12-4 *Gate Close Reason Sub-Codes*

Reason Sub-Code	Definition
0	Client initiated release (normal operation)
1	Reservation reassignment (e.g., for priority session)
2	Lack of reservation maintenance
3	Lack of DOCSIS MAC-layer responses
4	Timer T0 expiration (no GATE-SET from CMS)
5	Timer T1 expiration (no COMMIT from MTA)
6	Timer T7 expiration (Service Flow reservation timeout)
7	Timer T8 expiration (Upstream service flow activity timeout)
127	Other, unspecified error

DQoS Timers

Timers used in PacketCable 1.x DQoS include

- **T0**—The amount of time a gate can be in the allocated state. Timer starts when a gate is allocated and stops when the gate is authorized. This timer is defined on the CMTS. The recommended value of this timer is 30.
- **T1**—The amount of time a gate can be in the authorized state. Timer starts when a gate is authorized and stops when the gate is committed. Note: The timer does not stop when the gate is put into the reserved state. This timer is typically set in the GATE-SET message from the CMS, but if it is absent, the provisioned value on the CMTS is used. The recommended value of this timer is between 200 and 300 seconds.
- **T5**—The amount of allowable time between when a CMS signals an MTA to delete a connection and the time the resulting GATE-CLOSE command is received from the CMTS. If the GATE-CLOSE command is not received within this time period, the CMS issues a GATE-DELETE to the CMTS to delete the gate. The recommended value of this timer is 5 seconds.
- **T7**—The amount of time reserved but not committed resources are to be held. This timer is specified by the CMS and is sent to the CMTS in a GATE-SET message. The CMTS then maps this parameter to the DOCSIS Timeout for Admitted QoS parameter, which is sent to the MTA in a DSA-RSP or DSC-RSP message. The recommended value of this parameter is 200 seconds.
- **T8**—The amount of time committed resources can remain unused. This is also referred to as the inactivity timer. Like T7, this timer is specified by the CMS and is sent to the CMTS in a GATE-SET message. The CMTS maps this parameter to the DOCSIS Timeout for Active QoS parameter, which is sent to the MTA in a DSA-RSP or DSC-RSP message. The default value of this timer is 0, meaning the CMTS should not poll the service flow for activity. It is recommended that a non-zero value be used for this parameter to ensure that “hung” resources can be reclaimed.

Authorizing DQoS Resources

The CMS determines what resources the MTA is authorized to use and signals this information to the CMTS. In the gate specification parameter in the Gate Set message, DQoS resource envelopes are defined using Resource Reservation Protocol (RSVP) flow specifications. The values of the RSVP FlowSpec are determined by information the CMS knows about the MTA DQoS requirements for a call.

You might be asking yourself, how does the CMS know what amount of DQoS resources a MTA requires? The answer is from information obtained from the NCS call signaling. As you might recall, in NCS the Local Connection Descriptor and Remote Connection Descriptor parameters describe the endpoint’s bearer information. This information is encoded using the Session Description Protocol (SDP). SDP was introduced in Chapter 6, “Signaling Interfaces and MGCP Overview,” and its use in the NCS protocol was covered in Chapter 7. Please refer to these chapters for more information on how the NCS and SDP protocols function. The specific SDP fields used are the media announcement parameter (m), the bandwidth parameter (b), and the “packetization period” attributes (either mptime or ptime) as demonstrated in the following example:

```
200 161096008 OK
I: 575

v=0
o=- 161096008 1397 IN IP4 24.34.240.245
s=-
c=IN IP4 24.34.240.245
b=AS:82     <- Bandwidth requirement in kbps of the media sessions
t=0 0
m=audio 53456 RTP/AVP 0 8 15    <- Possible CODECs used by the MTA endpoint
a=X-pc-secret:base64:<hex digit string omitted>
a=X-pc-csuites-rtp:62/51 64/51 60/51 60/50
a=X-pc-csuites-rtcp:81/71 81/70 80/70
a=X-pc-nrekey:0
a=mptime:20 20 20     <- Corresponding packetization periods used with CODECs
```

NOTE In Chapter 10 you learned that this CODEC list is negotiated between the gateway endpoints. In a majority of today’s deployments only the G.711 CODEC is used; therefore, the CODEC list consists of this single CODEC.

Summary of RSVP Flow Specifications

An RSVP flow specification consists of two parts: the traffic description (TSpec) and the resource description (RSpec). Within RSVP you have two types of services: controlled load and guaranteed. The guaranteed type of service is used for applications where delay is an issue, such as voice.

A TSpec of a guaranteed type of service consists of the following parameters:

- **(b)**—Token bucket depth in bytes, which represents a reservoir of credit correlating to the maximum amount of data permitted to be sent in a given time interval
- **(r)**—Token bucket rate in bytes/second, which specifies how fast credits are added to this reservoir
- **(p)**—Peak rate in bytes/second that the source sends
- **(m)**—Minimum policed unit in bytes, which is typically the smallest packet size the source sends
- **(M)**—Maximum packet size in bytes that the source sends

An RSpec of a guaranteed type of service consists of the following parameters:

- **(R)**—Reserved rate in bytes/second, which is the amount of bandwidth allocated to the flow
- **(S)**—Slack term in microseconds (PacketCable uses this term to define maximum allowable packet jitter)

In PacketCable, voice packets in a call are sent one at a time and are of a constant size. Thus, the (b), (m), and (M) parameters must be equal to each other. These values equate to the size of a single digitized voice packet, including the IP and higher layer headers. The MTA transmits these voice packets at a constant rate; consequently, the (r), (p), and (R) parameters must also be equal to each other. The (S) parameter is defined on the CMS; by default, it has a value of 800 microseconds for the upstream and 0 for the downstream.

Mapping SDP Information into RSVP Flow Specifications

Now that you have some understanding of how these protocols are formatted it's time to examine how the CMS converts SDP information into the RSVP flow specifications used in the COPS PacketCable Gate Specification object.

The CMS uses the CODEC and the packetization period obtained from SDP to create the RSVP flow specifications. The easiest way to illustrate how this works is with an example.

Assume the MTA has a SDP descriptor with the following parameters:

```
m=audio 49182 RTP/AVP 0
a=ptime:20
```

RTP payload type 0 is the G.711 μ-law CODEC; the packetization period is 20 ms. In Chapter 11, you saw the resulting IP packet size in this case is 200 bytes. This is the value of the (b), (m), and (M) parameters in the TSpec portion of the RSVP flow specification. The (r), (p), and (R) parameters in the RSVP FlowSpec correspond to the rate at which the MTA sends data. In this example, a 200-byte packet is sent every 20 ms, equating to a rate of 10,000 bytes/sec (200 byte/20 ms). The only other remaining RSVP FlowSpec parameter is the slack term that is either defined on the CMS or the default value is used.

Figure 12-9 illustrates this process; also for more detail refer to Part IV of this book.

Figure 12-9 *Mapping SDP Parameters into RSVP FlowSpecs*

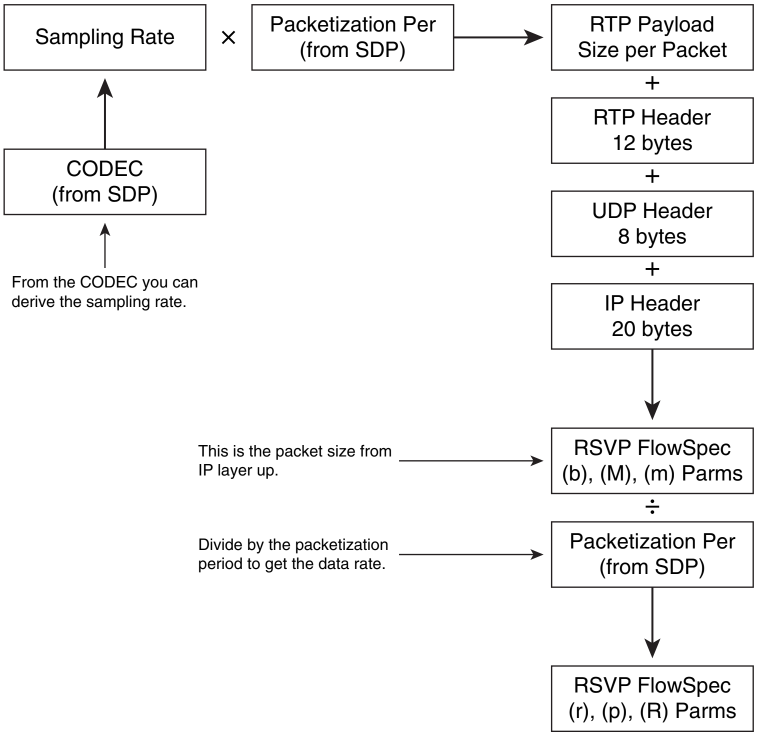

Sampling Rate
×
Packetization Per (from SDP)
RTP Payload Size per Packet
+
RTP Header 12 bytes
+
UDP Header 8 bytes
+
IP Header 20 bytes
CODEC (from SDP)
From the CODEC you can derive the sampling rate.
This is the packet size from IP layer up.
RSVP FlowSpec (b), (M), (m) Parms
÷
Divide by the packetization period to get the data rate.
Packetization Per (from SDP)
RSVP FlowSpec (r), (p), (R) Parms

Table 11-2 in Chapter 11 shows the bandwidth constraints of all the possible CODEC and packetization period combinations for PacketCable. Table 12-5 shows what the corresponding Flow Specification parameters would be for each of these CODECs. Neither this table nor the formula in Figure 12-9 shows the impact of bearer stream security or payload header suppression on these parameters. How payload header suppression works and affects bearer traffic is covered later in this chapter. Bearer stream security works by encapsulating the RTP packets inside Encapsulation Security Payload (ESP) frames used for IP security; the net effect of this is packet sizes are increased by four bytes. More on security is covered in Chapter 16.

Table 12-5 *CODEC Flow Specification Parameters*

CODEC	Pack. Period	Total Packet Size	FlowSpec (b),(M),(m) Parameters	Resulting Bit Rate of One Call	FlowSpec (r),(p),(R) Parameters
G.711	10 ms	120 bytes	120 bytes	96 Kbps	12,000 bytes/sec
G.711	20 ms	200 bytes	200 bytes	80 Kbps	10,000 bytes/sec
G.711	30 ms	280 bytes	280 bytes	~74.67 Kbps	9,334 bytes/sec

Table 12-5 *CODEC Flow Specification Parameters (Continued)*

CODEC	Pack. Period	Total Packet Size	FlowSpec (b),(M),(m) Parameters	Resulting Bit Rate of One Call	FlowSpec (r),(p),(R) Parameters
G.728	10 ms	60 bytes	60 bytes	48 Kbps	6,000 bytes/sec
G.728	20 ms	80 bytes	80 bytes	32 Kbps	4,000 bytes/sec
G.728	30 ms	100 bytes	100 bytes	~26.67 Kbps	3,334 bytes/sec
G.729E	10 ms	55 bytes	55 bytes	44 Kbps	5,500 bytes/sec
G.729E	20 ms	70 bytes	70 bytes	28 Kbps	3,500 bytes/sec
G.729E	30 ms	85 bytes	85 bytes	~22.67 Kbps	2,834 bytes/sec
iLBC	20 ms	78 bytes	78 bytes	31.2 Kbps	3,900 bytes/sec
iLBC	30 ms	90 bytes	90 bytes	24 Kbps	3,000 bytes/sec
BV16	10 ms	60 bytes	60 bytes	48 Kbps	6,000 bytes/sec
BV16	20 ms	80 bytes	80 bytes	32 Kbps	4,000 bytes/sec
BV16	30 ms	100 bytes	100 bytes	~26.67 Kbps	3,334 bytes/sec

Reserving and Committing DQoS Resources

So far you have seen how an authorized resource envelope is created. Next, you will learn how reserved and committed resource envelopes are created. Remember, NCS call signaling parameters trigger an embedded MTA to initiate requests for DQoS resource reservation and commitment. DOCSIS mechanisms are used by the MTA; thus, a review of DOCSIS aspects pertinent to QoS is warranted.

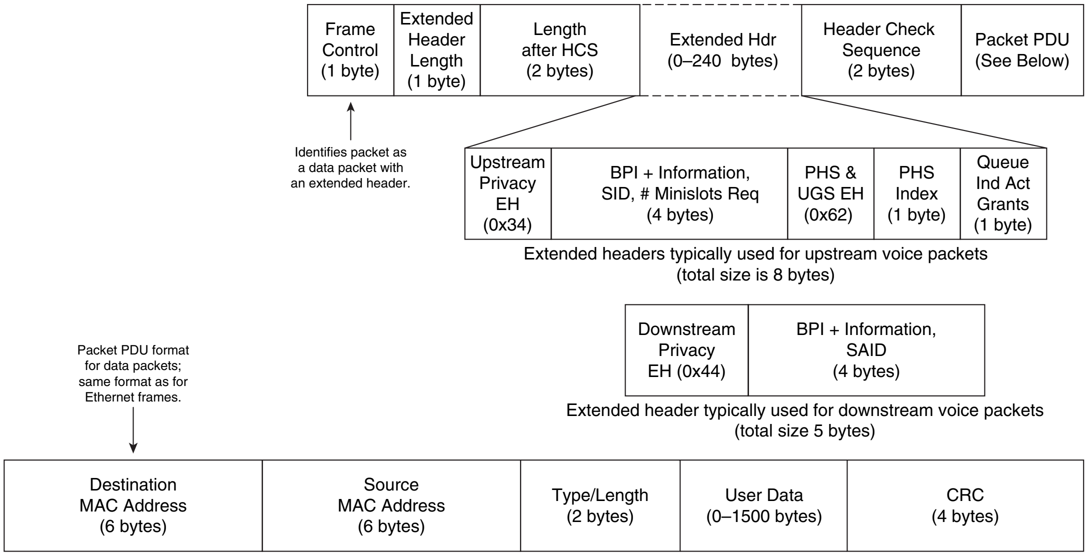
Reviewing DOCSIS 1.1 QoS

In Chapter 1 you learned that the DOCSIS 1.1 RFI specification added support for quality of service mechanisms in DOCSIS. PacketCable makes use of these mechanisms to provide DQoS premium services. The key aspects of DOCSIS used by PacketCable DQoS are summarized here. For more detail please refer to the DOCSIS 1.1 RFI specification, which can be obtained at http://www.cablemodem.com.

DOCSIS Headers

Some parameters in DOCSIS headers convey information on how voice packets use DQoS resources. Consequently, a review of how DOCSIS headers are used is necessary. Also being familiar with how DOCSIS messages are formatted can help you in the capacity planning discussion in Chapter 16. Figure 12-10 illustrates the format of the DOCSIS header and extended headers typically used for PacketCable voice packets.

Figure 12-10 *DOCSIS Headers and Extended Headers*


Frame Control (1 byte)
Extended Header Length (1 byte)
Length after HCS (2 bytes)
Extended Hdr (0–240 bytes)
Header Check Sequence (2 bytes)
Packet PDU (See Below)
Identifies packet as a data packet with an extended header.
Upstream Privacy EH (0x34)
BPI + Information, SID, # Minislots Req (4 bytes)
PHS & UGS EH (0x62)
PHS Index (1 byte)
Queue Ind Act Grants (1 byte)
Extended headers typically used for upstream voice packets (total size is 8 bytes)
Downstream Privacy EH (0x44)
BPI + Information, SAID (4 bytes)
Extended header typically used for downstream voice packets (total size 5 bytes)
Packet PDU format for data packets; same format as for Ethernet frames.
Destination MAC Address (6 bytes)
Source MAC Address (6 bytes)
Type/Length (2 bytes)
User Data (0–1500 bytes)
CRC (4 bytes)

All DOCSIS messages have at least a 6-byte header. This header defines whether the packet is a DOCSIS management message or an actual data packet. This header also contains the length of the packet and a header checksum. Additionally, in the header is a flag that indicates the presence of a DOCSIS extended header. If this flag is set, the length of the extended header is also included in the DOCSIS header. If DOCSIS extended headers are present, they are included between the two bytes in the header used for length and the header checksum. DOCSIS extended headers begin with a 4-bit type field and a 4-bit length field followed by the variable length data.

Some uses of the DOCSIS extended header include piggybacking data grant requests and providing packet fragmentation functionality. In PacketCable, extended headers convey information about Baseline Privacy, Payload Header Suppression, and Unsolicited Grant Service (UGS) grant usage.

The top of Figure 12-10 shows the general format of the DOCSIS header used for IP data packets. Below this are the formats of the extended headers typically used in PacketCable. First the extended header used for upstream (MTA → CMTS) packets is shown, and then the extended header used for downstream (CMTS → MTA) packets. In both cases, a 5-byte Baseline Privacy extended header identifies whether BPI is enabled, the SID or SAID the packet is associated with, and piggyback minislot requests.

NOTE For PacketCable, voice piggybacks are disallowed, so this value is zero. Also SIDs are valid only for upstream packets; downstream packets use a Security Association Identifier (SAID) that is typically the same value as the upstream SID.

Another extended header (EH) used in upstream packets is the 3-byte UGS EH (type 6, length 2). It contains a queue indicator (QI) that lets the CMTS know whether the modem has packets waiting in the queue to be sent and the number of active grants used by the service flow. For PacketCable 1.0 the number of active grants is always one unless silence suppression is used; in this case the number of active grants is equal to zero during periods of silence. In PacketCable 1.5, this parameter can be set to values greater than 1. This can be used to allow multiple calls to share the same set of DQoS resources; the value of this parameter then indicates the number of active calls or sub-flows. If the QI bit is set, this typically indicates the MTA is missing UGS opportunities. This can be because of noise in the DOCSIS plant and/or a malfunctioning CMTS or MTA. If packets are stuck in the queue for too long, jitter problems can result, causing poor voice quality. If the party a PacketCable MTA is talking to complains about hearing garbled voice, look for set QI bits. This extended header also contains the Payload Header Suppression Index (PHSI) if one exists. Payload Header Suppression (PHS) can also be active on a downstream service flow; in this case an additional 2-byte extended header is included (type 5, length 1). PHS is covered later in the chapter.

Dynamic Service Flows

In DOCSIS 1.1 all packets traversing the Hybrid Fiber Coaxial (HFC) network do so via unidirectional service flows; associated with a service flow are parameters describing the traffic. For a typical high-speed data user two Best Effort service flows are defined in the modem's configuration file (one in the upstream direction and one in the downstream direction). All traffic to and from the modem and CPE devices behind the modem use these primary service flows.

For a PacketCable user, voice traffic is separated from the rest of the traffic by using dedicated service flows better suited for voice traffic. There is one upstream service flow and one downstream service flow for voice. These service flows are created when needed by DOCSIS management messages introduced in DOCSIS 1.1. These messages allow for the dynamic creation, modification, and deletion of service flows.

Dynamic service flow creation can be initiated by either the cable modem or the CMTS. For PacketCable DQoS, as described in this chapter, dynamic service flow creation is initiated from the cable modem. The NCS messaging sent from the CMS to the embedded MTA actually triggers the service flow creation by the cable modem, as demonstrated here:

```
MDCX 161096010 aaln/1@x1-6-00-13-10-50-38-02.cc.cisco.com MGCP 1.0 NCS 1.0
I: 575
C: 284
L: dq-gi:1463,  dq-rr:snrcresv        ← DQoS parameters
M: recvonly
K: 161096008

v=0
o=- 161096009 29086 IN IP4 24.34.240.247
s=-
c=IN IP4 24.34.240.247
b=AS:82
t=0 0
m=audio 53456 RTP/AVP 0 8 15
a=X-pc-secret:base64:<hex digit string omitted>
a=X-pc-csuites-rtp:62/51 64/51 60/51 60/50
a=X-pc-csuites-rtcp:81/71 81/70 80/70
a=X-pc-nrekey:0
a=mptime:20 20 20
a=ptime:20
```

This message is Step #7 in the basic call example in Chapter 7. Notice the Local Connection Option parameters dg-gi (DQoS gate identifier) and dq-rr (DQoS resource reservation) in this message. The gate identifier tells the MTA how to correlate the DQoS resources it is trying to reserve to the DQoS resource previously authorized on the CMTS. The resource reservation parameter tells the MTA that it needs to initiate the request for DQoS resource reservation for both the upstream (send) and downstream (receive) directions. Also notice the remote SDP information in this message. The MTA uses this information to identify the amount of resources required and to build the classifiers used to associate IP packets to these resources.

Assuming the CMTS can successfully reserve resources, it assigns a resource identifier and returns this value to the MTA. The MTA sends this value to the CMS in the response to the previous DQoS request. The response for the request in the example is as follows:

```
200 161096010 OK
K:
DQ-RI: 2BF
```

For more information on NCS, please refer to Chapter 7.

The DOCSIS message used to create a dynamic service flow is a Dynamic Service Addition Request (DSA-REQ) message. The CMTS then either accepts or permits this service flow creation and sends the result to the cable modem using the Dynamic Service Addition Response (DSA-RSP) message. The CM acknowledges the receipt of this message with a Dynamic Service Addition Acknowledgement (DSA-ACK) message.

These service flows can also be modified using DOCSIS messaging. Just as with dynamic service flow creation, a 3-way handshake is used to modify a service flow. These messages are the DSC-REQ, DSC-RSP, and DSC-ACK referring to a Dynamic Service Change Request, Response, and Acknowledgement message.

The dynamic service change message exchange is typically used to commit DQoS resources. Again, parameters in the NCS messaging sent from the CMS to the MTA tell it when to initiate the request for resource commitment:

```
MDCX 161096016 aaln/1@x1-6-00-13-10-50-38-02.cc.cisco.com MGCP 1.0 NCS 1.0
I: 575
C: 284
L: dq-gi:1463,  dq-rr:snrccomt,  dq-ri:2BF
M: sendrecv
K: 161096013
```

This is Step #14 from the basic call flow example in Chapter 7. As you can see, the DQoS resource reservation parameter is now set to commit for both the send and receive directions. The DQoS resource identifier (dq-ri) parameter, which identifies the resources previously reserved on the CMTS, is also included.

Logically, these service flows are also deleted using DOCSIS messages. The messages used for this are the DSD-REQ and DSD-RSP referring to a Dynamic Service Deletion Request and Response.

Authorization Block Usage

According to the DOCSIS 1.1 RFI, the Authorization Block parameter contains an authorization "hint" to be passed to the authorization function of the CMTS. The parameter can be in a CM initiated DSA-REQ or DSC-REQ and a CMTS initiated DSA-RSP or DSC-RSP.

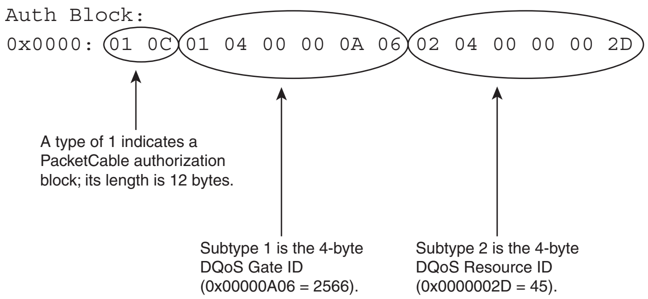
PacketCable uses this parameter to transfer DQoS gate and resource information between the CM and CMTS. In a CM-initiated DSA-REQ or DSC-REQ the Authorization Block parameter is used to pass the gate identifier the MTA/CM wants to use to the CMTS. The CMTS then verifies whether the embedded MTA is permitted to use the resources authorized by that gate. In a CMTS-initiated DSA-RSP or DSC-RSP the authorization block parameter is used to provide the embedded MTA with the DQoS resource identifier that references the premium DOCSIS resources to be used by the MTA. This information can then be passed along to the CMS, where decisions can be made on whether these resources can be shared. For example, in a call-waiting scenario resources can be shared among multiple gates, but these resources can be committed for only one of the gates at a time. In this type of call, the resource identifier parameter would also be included in the CM initiated DSA-REQ or DSC-REQ to notify the CMTS it wants to reuse existing resources.

The Authorization Block parameter is encoded in a Type/Length/Value (TLV) format. Type 1 indicates a PacketCable authorization block; its value is further broken down into TLV pairs. Subtype 1 indicates a DQoS gate ID that is 4 bytes long. Subtype 2 indicates a DQoS resource ID that is also 4 bytes long.

In PacketCable 1.5, subtype 3 is added to the specification. This parameter is the DQoS "sub-flow" status and is valid only when the maximum number of UGS grants per interval parameter is greater than one. It is used when multiple gates share the same DOCSIS service flows. It has four possible values: 0 (Admitted), 1 (Active), 2 (Deleted), and 3 (Move).

Figure 12-11 illustrates an example decode of the Authorization Block TLV.

Figure 12-11 *Authorization Block Encoding*



```
Auth Block:
0x0000: 01 0C 01 04 00 00 0A 06 02 04 00 00 00 2D
```

A type of 1 indicates a PacketCable authorization block; its length is 12 bytes.

Subtype 1 is the 4-byte DQoS Gate ID (0x00000A06 = 2566).

Subtype 2 is the 4-byte DQoS Resource ID (0x0000002D = 45).

Generic Service Flow Parameters

Some of the service flow parameters are applicable only to an upstream service flow, while some of them are applicable only for a downstream service flow. A few apply to both directions.

NOTE Several of the parameters described in the downstream section can also be applied to an upstream Best Effort service flow. However, only parameters relevant to the service flows used for PacketCable 1.x are covered here.

These service flow parameters are

- **Service Flow Reference**—Used to associate a classifier with a new service flow.
- **Service Flow Identifier**—4-byte identifier used by the CMTS that uniquely identifies a service flow.
- **QoS Parameter Set Type**—This 3-bit field defines whether the specified QoS parameters apply to a provisioned service flow (LSB - bit 0), an admitted service flow (bit 1), or an active service flow (MSB - bit 2). These bits can be set individually or in conjunction. In PacketCable this field is one of three values. If this field is set to 010 (2), the QoS parameters apply to a service flow in the admitted state. This setting corresponds to DQoS resources in the reserved state in the two-phase activation model. If the field is set to 100 (4), the QoS parameters apply to a service flow in the

active state equating to the committed state in the two-phase activation model. If the field were set to 110 (6), the QoS parameters would be both admitted and activated (that is, reserved and committed), indicating a single-phase activation.

- **Timeout for Admitted QoS Parameters**—This field indicates the maximum length of time (in seconds) admitted QoS resources in excess of the active QoS resources can be held. If this timer expires, these unused excess resources are released. In PacketCable, this field cannot be populated by the MTA but is typically set by the CMTS in the DSA-RSP or DSC-RSP. The CMTS sets this timer to the DQoS timer T7 received in the Gate Set message from the CMS. If unspecified, a default value of 0 indicating an infinite timeout is used. Also note that in most situations the admitted (reserved) and active (committed) resource envelopes are the same.
- **Timeout for Active QoS Parameters**—This field indicates the maximum length of time (in seconds) resources on an active service flow can remain unused. If this timer expires, the active and admitted QoS resources are released. In PacketCable, this field cannot be populated by the MTA; it is set by the CMTS in the DSA-RSP or DSC-RSP message. The CMTS sets this timer to the DQoS timer T8 received in the Gate Set message from the CMS. If unspecified, a default value of 0 indicating an infinite timeout is used. Note: This field should be used by the CMTS as it ensures that “hung” resources are eventually recovered.

Upstream Service Flow Parameters

In Chapter 1 you learned that DOCSIS 1.1 provides for service scheduling types better suited for real-time applications that are more sensitive to delay and packet drops than the traditional Best Effort service type. Two of these service types, the Unsolicited Grant Service (UGS) and the Unsolicited Grant Service with Activity Detection (UGS-AD) are especially useful for providing voice services.

A UGS service provides fixed size periodic intervals for a modem to send traffic in the upstream direction. Unlike Best Effort intervals, these intervals are dedicated for a particular user. Also no delay is involved in requesting this bandwidth as the grants are provided without an explicit request from the modem—hence, the grants are unsolicited. By now you are aware that a gateway digitizes voice using a CODEC and a packetization period. This results in the periodic generation of fixed size packets, ideal for the UGS service.

The UGS-AD service is very similar to the UGS service except it is designed for calls using silence suppression or VAD to conserve bandwidth. When the voice communication is active, it is identical to the UGS service; however, when there are pauses or silence in the conversation, you need not waste grants so the service switches to a polling service. This is the DOCSIS Real-Time Polling Service (RTPS), where the modem is given periodic unicast opportunities to request bandwidth. When the CMTS detects UGS grants are not being used, it switches to the RTPS service. When conversation starts again, the modem uses one of these bandwidth requests (by sending a DOCSIS request packet), which signals the CMTS to start providing UGS grants again. Keep in mind that the CMTS is determining when to switch between UGS and RTPS; no DOCSIS DSX messaging is exchanged with the MTA in this case.

The service flow parameters used to define a UGS or UGS-AD service are as follows:

- **Service Identifier**—The 14-bit service identifier is assigned by the CMTS to reference the upstream service flow. It is used in the bandwidth allocation messages to identify the user of a data or request opportunity. It is also used in the DOCSIS extended header to reference the device sending the packet.
- **Service Flow Scheduling Type**—This field identifies the scheduling service for upstream data and request transmissions. This is where the service flow is identified as a UGS (6) or UGS-AD (5) service. Other possible values include: Best Effort (2), Non Real-Time Polling (3), and Real-Time Polling (4).
- **Request/Transmission Policy**—This field is a bit map that controls when and how the service flow can transmit data and data requests. A bit value of "1" indicates true, and a bit value of "0" indicates false. For PacketCable UGS and UGS-AD, service flows bits 0 through 6, and 8 are always set; hence, the value is either 0x000001ff or 0x0000017f depending upon whether or not PHS is allowed. Table 12-6 defines these lower nine bit positions.

Table 12-6 *Request/Transmission Policy Bits*

Bit Position	Value	Value for UGS/UGS-AD
0	Use of broadcast request opportunities disallowed	1 (service flow has dedicated data grants, so no need to make bandwidth requests)
1	Use of priority multicast request opportunities disallowed	1 (service flow has dedicated data grants, so no need to make bandwidth requests)
2	Use of Request/Data opportunities for requests disallowed	1 (service flow has dedicated data grants, so no need to make bandwidth requests)
3	Use of Request/Data opportunities for data disallowed	1 (service flow has dedicated data grants, so no need to use these contention data slots)
4	Piggybacking of requests with data disallowed	1 (service flow has periodic dedicated data grants, so no need to piggyback requests)
5	Concatenation disallowed	1 (concatenation of voice packets not allowed)
6	Fragmentation disallowed	1 (fragmentation of voice packets not allowed)
7	Payload Header Suppression disallowed	If set to 0, payload header suppression can be used to increase data efficiency
8	Packets that do not fit the UGS size are dropped	1 (packets larger than the UGS size are dropped)

- **Nominal Polling Interval**—Interval in microseconds for dedicated request opportunities. Used in the UGS-AD service during silent periods to give the MTA uncontested opportunities to signal the CMTS when voice starts again. For PacketCable this parameter must be an integer multiple of the Nominal Grant Interval.
- **Tolerated Poll Jitter**—Amount of time in microseconds the polling interval is permitted to vary from the Nominal Polling Interval. This parameter is used for the UGS-AD service type.
- **Unsolicited Grant Size**—The size of each digitized voice packet in bytes. This size is from the DOCSIS Header Frame Control byte to the CRC at the end of the packet. In other words, the "Ethernet" portion of the DOCSIS overhead is included as well as the general 6-byte DOCSIS header and any DOCSIS extended headers.
- **Nominal Grant Interval**—Interval in microseconds for dedicated data opportunities. Used by the UGS and UGS-AD services to give the MTA uncontested opportunities to transmit voice packets. This field directly corresponds to the MTA packetization period and can only have a value of 10 ms, 20 ms, or 30 ms.
- **Tolerated Grant Jitter**—Amount of time in microseconds the grant interval is permitted to vary from the Nominal Grant Interval. This parameter defines the permissible jitter for the UGS and UGS-AD service types. In PacketCable this parameter is specified by the CMS; if it is left unspecified, a default value of 800 µs is used.
- **Grants per Interval**—Number of dedicated data grants given to the MTA during each grant interval. For PacketCable 1.0, this value is always equal to "1." In PacketCable 1.5 more flexibility is permitted with this parameter. It can be set to larger values to allow multiple calls to share the same DQoS resources. For example with this set to "2", each call gets 1 grant per interval, creating sub-flows within the main DQoS flow. Obviously, for this to be permitted the characteristics of these sub-flows needs to be identical (i.e., same CODEC, packetization, use of silence suppression, etc.).

Downstream Service Flow Parameters

In addition to the service flow for the premium treatment of voice packets in the upstream, a service flow for voice packets is also created for the downstream. Remember, all downstream traffic is Best Effort. However, you have ways to prioritize voice traffic over ordinary data traffic and to reserve bandwidth for this traffic, as you will see. The key parameters used to define the downstream service flow are as follows:

- **Traffic Priority**—Defines the service flow priority. If multiple service flows exist that are identical in all QoS parameters besides priority, the higher priority service flow is given preference. If not included, the default priority of "0" is used. For PacketCable 1.x downstream flows, this is set to the value "5."

- **Maximum Sustained Traffic Rate**—The token bucket (R) parameter for rate limiting packets expressed in bits per second. The packet size is from after the DOCSIS Header Check Sequence to the CRC at the end of the packet. In other words, the "Ethernet" portion of the DOCSIS overhead is included, but the general 6-byte DOCSIS header and any DOCSIS extended headers are not.
- **Maximum Traffic Burst**—The token bucket size (B) parameter for rate limiting packets expressed in bytes. Again, this parameter accounts for the total packet size in the same way as the Maximum Sustained Traffic Rate parameter. The minimum value is 1,522 bytes, and if unspecified, a default value of 3,044 bytes is implied. This parameter is meaningful only if the Maximum Sustained Traffic Rate parameter is non-zero.
- **Minimum Reserved Traffic Rate**—The minimum rate in bits per second reserved for the service flow. Packet size is calculated the same as in the previous parameters.
- **Minimum Reserved Rate Packet Size**—The assumed minimum packet size in bytes for which the Minimum Reserved Traffic Rate is applied. Packet size is calculated the same as in the previous parameters.

Classifying Packets

Packets are assigned to service flows by matching packet classifiers. Some of the things a packet classifier can match packets upon include source and/or destination IP addresses, source and/or destination UDP/TCP port numbers, IP protocol type, and IP TOS bits. You have upstream packet classifiers used to assign packets to upstream service flows and downstream packet classifiers used to assign packets to downstream service flows. A cable modem can be associated with multiple upstream and downstream classifiers; in this case a classifier priority is used to determine the order in which classifiers are applied.

NOTE If a packet fails to match any classifier, it is sent out the primary service flow. Remember, the primary service flows are the first upstream and downstream service flows defined in the cable modem configuration file. Also keep in mind these classifiers can change during the life of a call, for example, with a call transfer scenario.

Some fields associated with DOCSIS classifiers used in PacketCable are

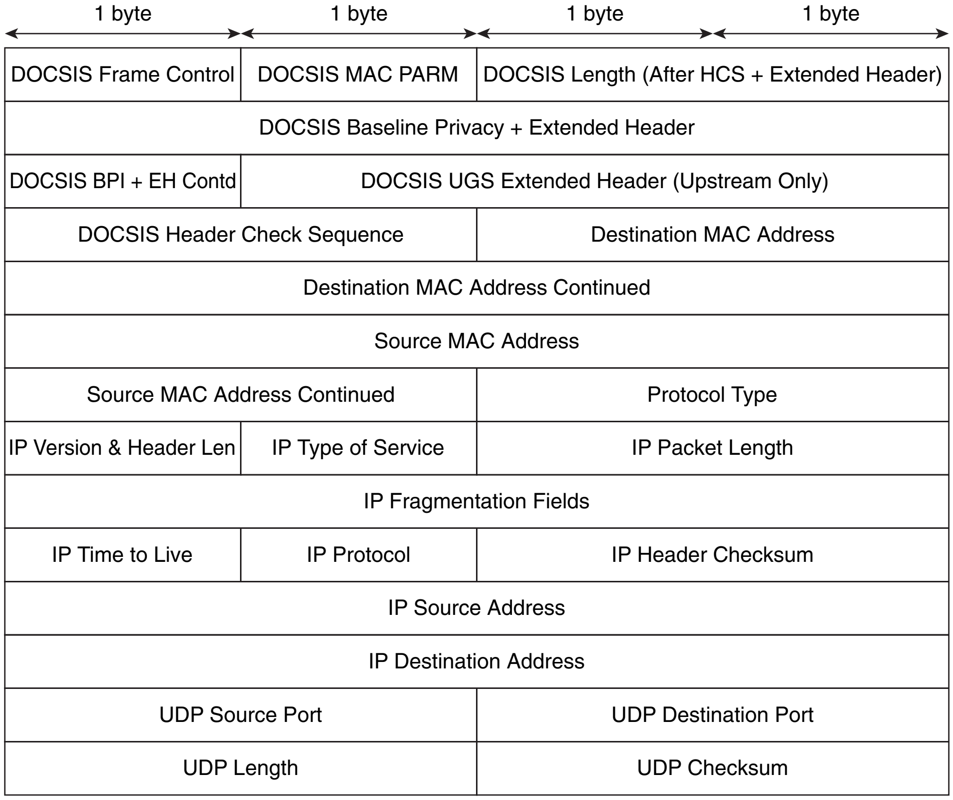
- **Classifier Reference/Classifier Identifier**—If this is the first time the classifier is introduced in the call, the Classifier Reference field is used and populated with a unique value by the MTA. If this classifier already exists, the Classifier Identifier field is used, populated with the value assigned by the CMTS.

- **Service Flow Reference/Service Flow Identifier**—One of these fields is used to correlate the Classifier with its corresponding service flow. If the service flow already exists, the Service Flow Identifier is used; otherwise, the Service Flow Reference is used.
- **Rule Priority**—This field is used to identify the order classifiers are applied. Higher priority classifiers are applied first. For PacketCable voice service flows, you have only one classifier for each service flow, so this field really doesn't matter. In this case, the priority is always the default of 128.
- **Classifier Activation State**—This field determines whether or not this classifier should be used for selecting packets. A value of inactive (0) is typically used for admitted service flows. Whether or not a classifier is used also depends on the state of the service flow; an active classifier still is not used if its associated service flow is not active.
- **Dynamic Service Change Action**—Action to be taken with this classifier. Values are 0 - Add Classifier, 1 - Replace Classifier, and 2 - Delete Classifier. This parameter is used in PacketCable 1.5 for the addition and deletion of sub-flows when multiple calls share the same DQoS resources.

Payload Header Suppression

In Chapter 11, you learned that voice packets are encapsulated using the RTP protocol. These RTP packets are encapsulated in UDP packets, which are encapsulated in IP packets, which are encapsulated in DOCSIS frames. Figure 12-12 shows the packet format.

Figure 12-12 *Voice Packet Header Format*



1 byte	1 byte	1 byte	1 byte
DOCSIS Frame Control	DOCSIS MAC PARM	DOCSIS Length (After HCS + Extended Header)	
DOCSIS Baseline Privacy + Extended Header			
DOCSIS BPI + EH Contd	DOCSIS UGS Extended Header (Upstream Only)		
DOCSIS Header Check Sequence		Destination MAC Address	
Destination MAC Address Continued			
Source MAC Address			
Source MAC Address Continued		Protocol Type	
IP Version & Header Len	IP Type of Service	IP Packet Length	
IP Fragmentation Fields			
IP Time to Live	IP Protocol	IP Header Checksum	
IP Source Address			
IP Destination Address			
UDP Source Port		UDP Destination Port	
UDP Length		UDP Checksum	

This is 56 bytes of overhead before getting to the RTP portion of the packet! Actually, the RTP protocol adds another 12 bytes of overhead before getting to the payload. Are all of these fields really needed in every packet? A closer look at these fields reveals that several are not used and several have the same value throughout the life of the call.

Payload Header Suppression (PHS) functionality eliminates the need to send these repetitive fields. Instead, a rule is created where the rule index represents these repetitive fields; thus, only the rule needs to be transmitted in these packets.

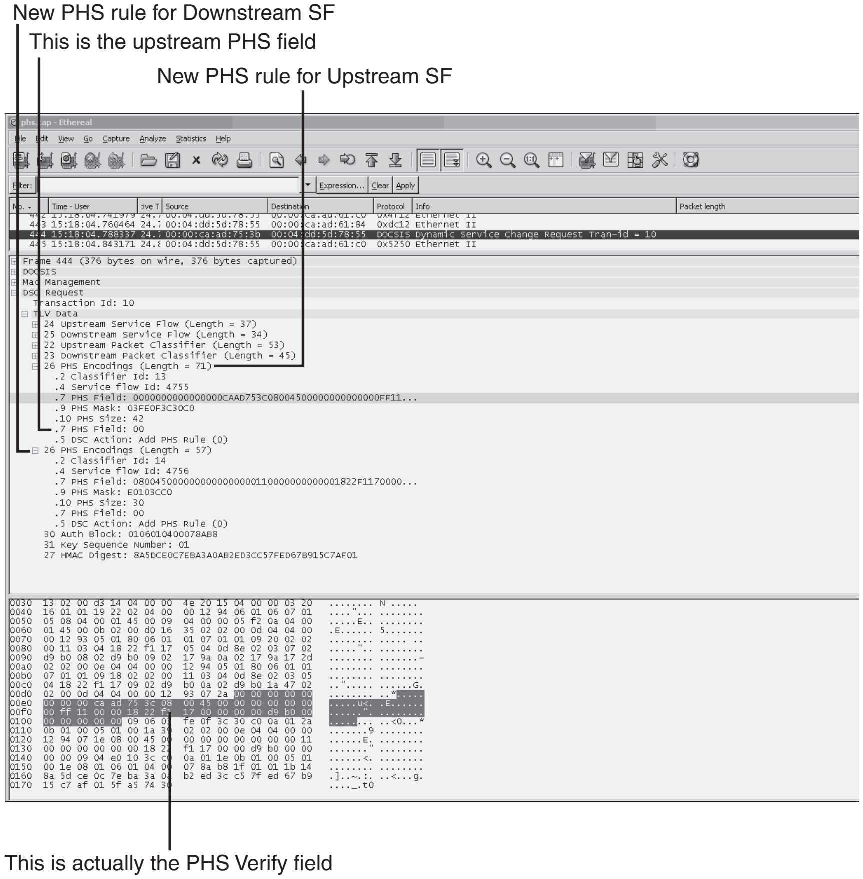
A PHS Rule accomplishes this using a number of parameters including

- **PHS Field (PHSF)**—String of bytes representing the header portion of a packet in which one or more bytes are suppressed. This is the uncompressed header that is kept in memory by the receiving end and then reinserted back into the packet before forwarding. For upstream packets the PHSF begins with the Destination MAC address. For downstream packets the MAC addresses cannot be suppressed because downstream packets are sent to all devices, and the MAC addresses are needed to determine the packet destination. Therefore, the PHSF begins with the protocol type field.

- **PHS Index (PHSI)**—A byte value used in the DOCSIS extended header to reference a PHS Rule.
- **PHS Mask (PHSM)**—A bit map that specifies which bytes in the PHSF are to be suppressed. Note: One bit refers to one byte in the PHSF. A bit value of "1" means to suppress the corresponding byte, whereas a bit value of "0" means the corresponding byte is not suppressed.
- **PHS Size (PHSS)**—Length of the PHSF in bytes and also the number of bits in the PHSM.
- **PHS Verify (PHSV)**—A flag that determines whether the sending entity should verify that the uncompressed header bytes actually match the PHSF bytes before suppressing them.

As with most things, the easiest way to understand how PHS works is through an example. Figure 12-13 shows an Ethereal capture of a DSC-REQ that adds PHS rules to both the upstream and downstream voice service flows.

Figure 12-13 *Payload Header Suppression Example*


New PHS rule for Downstream SF
This is the upstream PHS field
New PHS rule for Upstream SF
phs.cap - Ethereal
File
Edit
View
Go
Capture
Analyze
Statistics
Help
Filter:
Expression...
Clear
Apply
This is actually the PHS Verify field

```
No. .  Time - User        ive T  Source             Destination        Protocol Info                                                Packet length
  442 15:18:04.741979 24.7 00:04:dd:5d:78:55  00:00:ca:ad:61:c0  0x4112 Ethernet II
  443 15:18:04.760464 24.7 00:04:dd:5d:78:55  00:00:ca:ad:61:84  0xdc12 Ethernet II
  444 15:18:04.788337 24.7 00:00:ca:ad:75:3b  00:04:dd:5d:78:55  DOCSIS Dynamic Service Change Request Tran-id = 10
  445 15:18:04.843171 24.8 00:04:dd:5d:78:55  00:00:ca:ad:61:c0  0x5250 Ethernet II
```

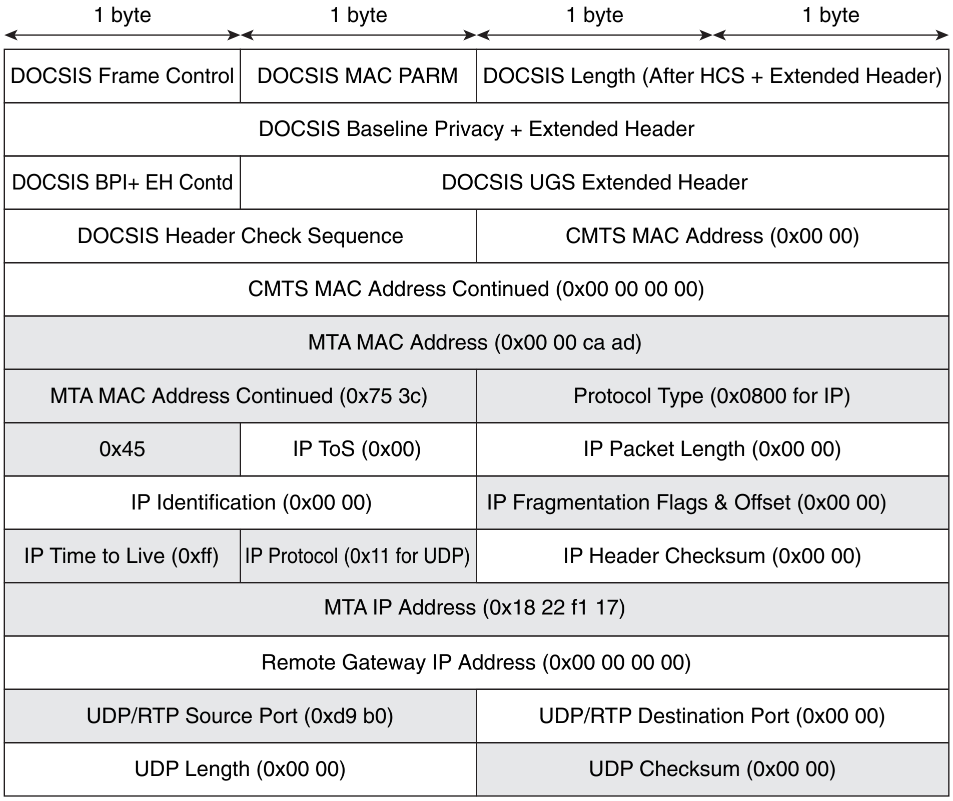
```
⊞ Frame 444 (376 bytes on wire, 376 bytes captured)
⊞ DOCSIS
⊞ Mac Management
⊟ DSC Request
    Transaction Id: 10
  ⊟ TLV Data
    ⊞ 24 Upstream Service Flow (Length = 37)
    ⊞ 25 Downstream Service Flow (Length = 34)
    ⊞ 22 Upstream Packet Classifier (Length = 53)
    ⊞ 23 Downstream Packet Classifier (Length = 45)
    ⊟ 26 PHS Encodings (Length = 71)
        .2 Classifier Id: 13
        .4 Service flow Id: 4755
        .7 PHS Field: 0000000000000000CAAD753C080045000000000000000000FF11...
        .9 PHS Mask: 03FE0F3C30C0
        .10 PHS Size: 42
        .7 PHS Field: 00
        .5 DSC Action: Add PHS Rule (0)
    ⊟ 26 PHS Encodings (Length = 57)
        .2 Classifier Id: 14
        .4 Service flow Id: 4756
        .7 PHS Field: 08004500000000000000000011000000000000001822F1170000...
        .9 PHS Mask: E0103CC0
        .10 PHS Size: 30
        .7 PHS Field: 00
        .5 DSC Action: Add PHS Rule (0)
      30 Auth Block: 0106010400078AB8
      31 Key Sequence Number: 01
      27 HMAC Digest: 8A5DCE0C7EBA3A0AB2ED3CC57FED67B915C7AF01
```

```
0030  13 02 00 d3 14 04 00 00  4e 20 15 04 00 00 03 20   ........ N .....
0040  16 01 01 19 22 02 04 00  00 12 94 06 01 06 07 01   ....."... ........
0050  05 08 04 00 01 45 00 09  04 00 00 05 f2 0a 04 00   .....E.. ........
0060  01 45 00 0b 02 00 d0 16  35 02 02 00 0d 04 04 00   .E...... 5.......
0070  00 12 93 05 01 80 06 01  01 07 01 01 09 20 02 02   ........ ..... ..
0080  00 11 03 04 18 22 f1 17  05 04 0d 8e 02 03 07 02   ....."..  ........
0090  d9 b0 08 02 d9 b0 09 02  17 9a 0a 02 17 9a 17 2d   ........ .......-
00a0  02 02 00 0e 04 04 00 00  12 94 05 01 80 06 01 01   ........ ........
00b0  07 01 01 09 18 02 02 00  11 03 04 0d 8e 02 03 05   ........ ........
00c0  04 18 22 f1 17 09 02 d9  b0 0a 02 d9 b0 1a 47 02   .."..... ......G.
00d0  02 00 0d 04 04 00 00 12  93 07 2a 00 00 00 00 00   ........ ..*.....
00e0  00 00 00 ca ad 75 3c 08  00 45 00 00 00 00 00 00   .....u<. .E......
00f0  00 ff 11 00 00 18 22 f1  17 00 00 00 00 d9 b0 00   ......". ........
0100  00 00 00 00 00 09 06 03  fe 0f 3c 30 c0 0a 01 2a   ........ ..<0...*
0110  0b 01 00 05 01 00 1a 39  02 02 00 0e 04 04 00 00   .......9 ........
0120  12 94 07 1e 08 00 45 00  00 00 00 00 00 00 00 11   ......E. ........
0130  00 00 00 00 00 00 18 22  f1 17 00 00 d9 b0 00 00   ......." ........
0140  00 00 09 04 e0 10 3c c0  0a 01 1e 0b 01 00 05 01   ......<. ........
0150  00 1e 08 01 06 01 04 00  07 8a b8 1f 01 01 1b 14   ........ ........
0160  8a 5d ce 0c 7e ba 3a 0a  b2 ed 3c c5 7f ed 67 b9   .]..~.:. ..<...g.
0170  15 c7 af 01 5f a5 74 30                             ...._.t0
```

The first set of PHS encodings is for voice upstream service flow 4755 using classifier 13. Because this is an upstream service flow the PHSF is 42-bytes long, note this matches the PHSS. Remember the PHSF for an upstream service flow begins with the Destination MAC address and continues to the end of the UDP header. The PHS Mask field, 0x03FE0F3C30C0, indicates which of these bytes are to be suppressed. Notice, the PHSM is 6 bytes or 48 bits long. Each bit corresponds to 1 byte in the PHSF, which is 42 bytes long, so the last 6 bits are just zero padding. The second PHSF field shown in the Figure 12-13 output is a bug; this field is actually the PHSV field. A value of “0” means that the sending entity must verify that the values of the fields to be suppressed in the actual packets match the corresponding values in the PHSF field. If this verification fails, then no suppression occurs.

The impact of PHS on an upstream packet is illustrated in Figure 12-14. The fields that are shaded are the ones that are going to be suppressed. To give you a little help, look at the first 12 bits of the PHSM (0x03F = 0000 0011 1111). The first 6 bits of 0 indicate the first 6 bytes of the PHSF (Destination MAC Address) are not to be suppressed. The second 6 bits of 1 indicate the next 6 bytes of the PHSF (Source MAC Address) are to be suppressed.

Figure 12-14 *Resulting Packet Header after PHS*



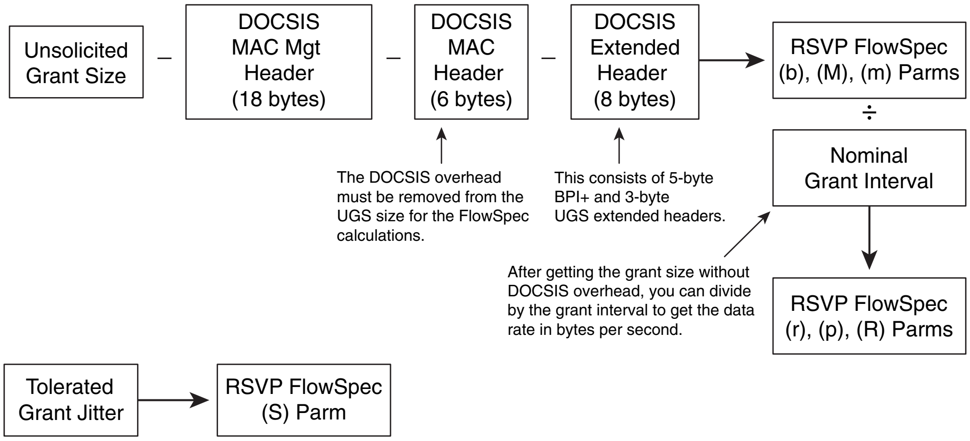
<table>
<tr><th>1 byte</th><th>1 byte</th><th>1 byte</th><th>1 byte</th></tr>
<tr><td>DOCSIS Frame Control</td><td>DOCSIS MAC PARM</td><td colspan="2">DOCSIS Length (After HCS + Extended Header)</td></tr>
<tr><td colspan="4">DOCSIS Baseline Privacy + Extended Header</td></tr>
<tr><td>DOCSIS BPI+ EH Contd</td><td colspan="3">DOCSIS UGS Extended Header</td></tr>
<tr><td colspan="2">DOCSIS Header Check Sequence</td><td colspan="2">CMTS MAC Address (0x00 00)</td></tr>
<tr><td colspan="4">CMTS MAC Address Continued (0x00 00 00 00)</td></tr>
<tr><td colspan="4">MTA MAC Address (0x00 00 ca ad)</td></tr>
<tr><td colspan="2">MTA MAC Address Continued (0x75 3c)</td><td colspan="2">Protocol Type (0x0800 for IP)</td></tr>
<tr><td>0x45</td><td>IP ToS (0x00)</td><td colspan="2">IP Packet Length (0x00 00)</td></tr>
<tr><td colspan="2">IP Identification (0x00 00)</td><td colspan="2">IP Fragmentation Flags & Offset (0x00 00)</td></tr>
<tr><td>IP Time to Live (0xff)</td><td>IP Protocol (0x11 for UDP)</td><td colspan="2">IP Header Checksum (0x00 00)</td></tr>
<tr><td colspan="4">MTA IP Address (0x18 22 f1 17)</td></tr>
<tr><td colspan="4">Remote Gateway IP Address (0x00 00 00 00)</td></tr>
<tr><td colspan="2">UDP/RTP Source Port (0xd9 b0)</td><td colspan="2">UDP/RTP Destination Port (0x00 00)</td></tr>
<tr><td colspan="2">UDP Length (0x00 00)</td><td colspan="2">UDP Checksum (0x00 00)</td></tr>
</table>

Mapping DOCSIS Service Flows into RSVP Flow Specifications

When the CMTS receives a DSA-REQ or DSC-REQ from an embedded MTA with service flow parameters for an admitted QoS parameter set, it compares the received DOCSIS parameters against the RSVP flow specifications that defined the authorized resource envelope. Remember, the reserved resource envelope must be less than or equal to the authorized resource envelope, or else the request is denied. The gate ID in the authorization block parameter in the DSX-REQ enables the CMTS to correlate DOCSIS requests to gates. To make the comparison the CMTS needs to convert the Layer 2 DOCSIS parameters into the equivalent Layer 3 RSVP flow specification parameters.

Figure 12-15 illustrates the mapping logic for an upstream service flow.

Figure 12-15 *DOCSIS Upstream Service Flows to RSVP Mapping*


Unsolicited Grant Size
–
DOCSIS MAC Mgt Header (18 bytes)
–
DOCSIS MAC Header (6 bytes)
–
DOCSIS Extended Header (8 bytes)
RSVP FlowSpec (b), (M), (m) Parms
÷
Nominal Grant Interval
RSVP FlowSpec (r), (p), (R) Parms
Tolerated Grant Jitter
RSVP FlowSpec (S) Parm

The DOCSIS overhead must be removed from the UGS size for the FlowSpec calculations.

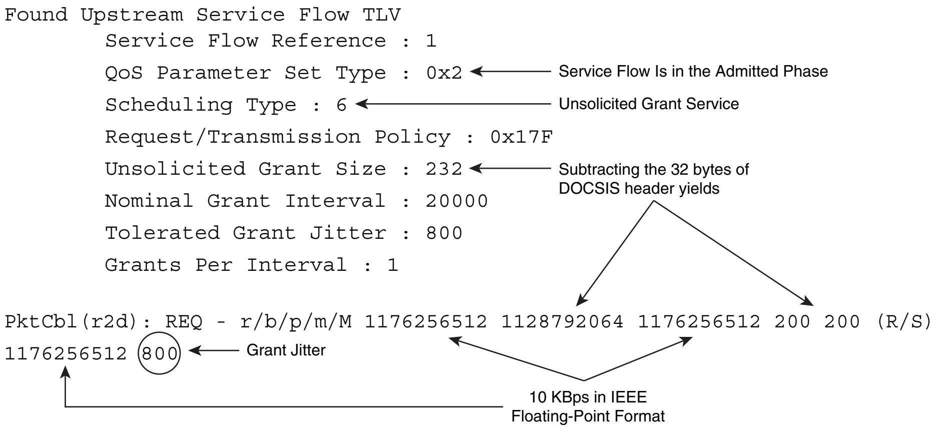
This consists of 5-byte BPI+ and 3-byte UGS extended headers.

After getting the grant size without DOCSIS overhead, you can divide by the grant interval to get the data rate in bytes per second.

The following example further illustrates this conversion. On the Cisco CMTS a couple of debugs were enabled that show the DOCSIS messages and the resulting RSVP flow specifications.

The output from these debugs has been modified just to illustrate the pertinent parts and is shown in Figure 12-16. Debugging DQoS on the Cisco CMTS is covered in detail in Chapter 13.

Figure 12-16 *Example Upstream Mapping*



```
Found Upstream Service Flow TLV
        Service Flow Reference : 1
        QoS Parameter Set Type : 0x2
        Scheduling Type : 6
        Request/Transmission Policy : 0x17F
        Unsolicited Grant Size : 232
        Nominal Grant Interval : 20000
        Tolerated Grant Jitter : 800
        Grants Per Interval : 1

PktCbl(r2d): REQ - r/b/p/m/M 1176256512 1128792064 1176256512 200 200 (R/S)
1176256512 800
```

Service Flow Is in the Admitted Phase

Unsolicited Grant Service

Subtracting the 32 bytes of DOCSIS header yields

Grant Jitter

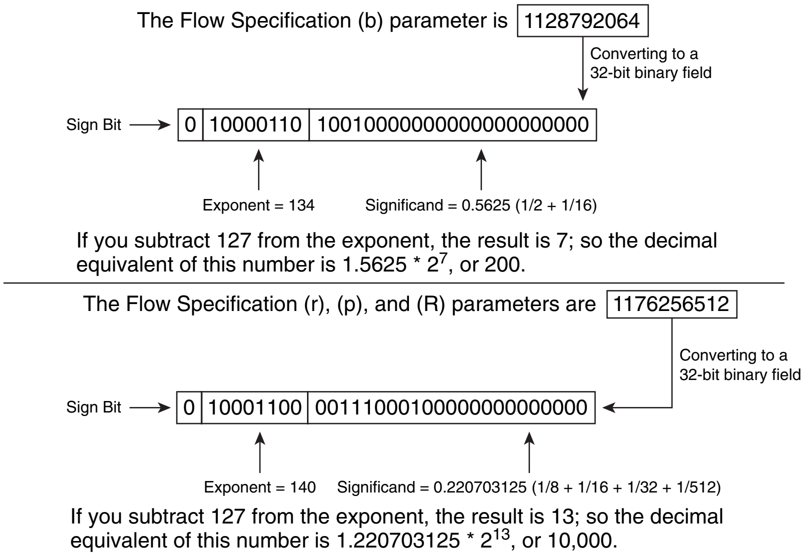
10 KBps in IEEE Floating-Point Format

This upstream service flow is from a DSA-REQ message. The QoS Parameter Set is 0x2, which means the QoS resources are to be placed in the admitted state or reserved state. Scheduling Type 6 is the UGS service. The Request Transmission Policy sets all the bits except for bit 7. This bit allows for Payload Header Suppression; the other bits do things such as disallow the use of contention request and data opportunities, disallow concatenation, and disallow fragmentation for this service flow. The UGS size is 232 bytes, the expected value for the G.711 CODEC with a 20 ms packetization.

So, if you subtract the 32 bytes of DOCSIS overhead from the UGS size, you are left with 200 bytes. You can see from the RSVP flow specification, the (m) and (M) parameters have values of 200. The (b) parameter as well as the (r), (p), and (R) parameters are expressed in single precision IEEE floating-point format, which is a 32-bit field.

NOTE The format of IEEE floating-point numbers is defined in the IEEE-754 specification. In summary, the 32-bit field is broken up into three sub-fields: a Sign field, an Exponent field, and a Significand field. The Sign field is the most significant bit (Bit 31) and is equal to zero (positive) for all the flow specification parameters. The next 8 bits comprise the Exponent field. If you convert this 8-bit field to decimal and subtract by 127, you get the value of the exponent. The remaining bits comprise the Significand. These bits represent fractions of base 2. The most significant bit (MSB) represents the value 1/2, the next MSB represents the value 1/4, the next the value 1/8, the next 1/16, etc. To compute the Significand, add up the values where the bits are set (equal to 1). To get the decimal value of the floating-point number multiply the Significand value plus 1 by the number 2 raised to the exponent value. Figure 12-17 shows some examples. If you do a search on the Internet, you can find several utilities to do this conversion.

Figure 12-17 *Sample Floating-Point-to-Decimal Conversions*



If you subtract 127 from the exponent, the result is 7; so the decimal equivalent of this number is 1.5625 * 2^7, or 200.

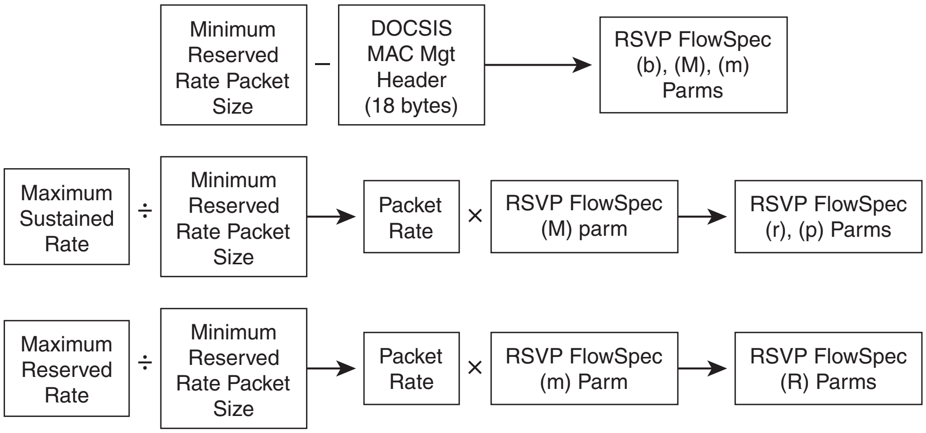
If you subtract 127 from the exponent, the result is 13; so the decimal equivalent of this number is 1.220703125 * 2^{13}, or 10,000.

If you now return to the example, you would see that if you were to do the conversion, the (b) value is also 200 bytes. Taking this value and dividing by the grant interval, which is 20,000 microseconds or 20 milliseconds, yields the data rate. This rate computes to be 10,000 bits per second. Converting this to IEEE floating-point notation yields the value 1,176,256,512 shown in the debug. The (S) parameter is taken from the value of the Tolerated Grant Jitter and is 800 microseconds.

Figure 12-18 illustrates the logic for mapping the downstream service flow parameters.

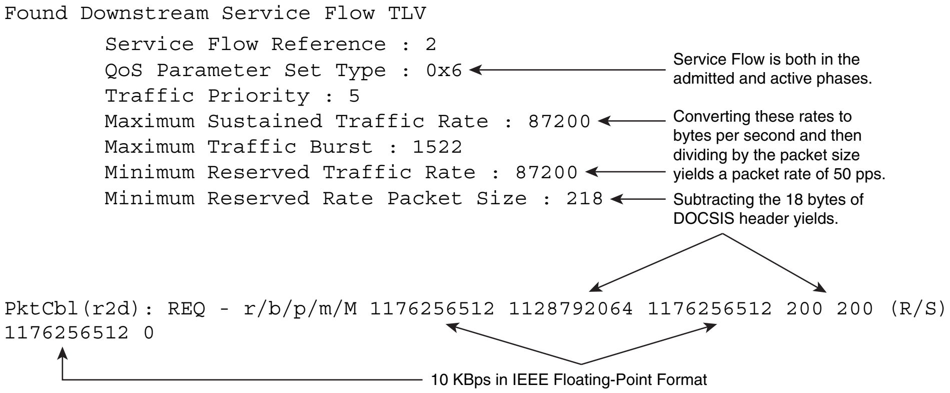
In Figure 12-19, an example is used to help you understand the conversion. This example is from the same call as the previous example, and information from the same debugs is collected.

Figure 12-18 *DOCSIS Downstream Service Flows-to-RSVP Mapping*


Minimum Reserved Rate Packet Size
–
DOCSIS MAC Mgt Header (18 bytes)
RSVP FlowSpec (b), (M), (m) Parms
Maximum Sustained Rate
÷
Minimum Reserved Rate Packet Size
Packet Rate
×
RSVP FlowSpec (M) parm
RSVP FlowSpec (r), (p) Parms
Maximum Reserved Rate
÷
Minimum Reserved Rate Packet Size
Packet Rate
×
RSVP FlowSpec (m) Parm
RSVP FlowSpec (R) Parms

Typically, the Maximum Sustained Rate is the same as the Maximum Reserved Rate, so both Packet Rates equate to the same value. Also because the (M) and (m) parameters are equal, the resulting RSVP FlowSpec parameters are also equal.

Figure 12-19 *Sample Downstream Mapping*



```
Found Downstream Service Flow TLV
        Service Flow Reference : 2
        QoS Parameter Set Type : 0x6
        Traffic Priority : 5
        Maximum Sustained Traffic Rate : 87200
        Maximum Traffic Burst : 1522
        Minimum Reserved Traffic Rate : 87200
        Minimum Reserved Rate Packet Size : 218


PktCbl(r2d): REQ - r/b/p/m/M 1176256512 1128792064 1176256512 200 200 (R/S)
1176256512 0
```

Service Flow is both in the admitted and active phases.

Converting these rates to bytes per second and then dividing by the packet size yields a packet rate of 50 pps.

Subtracting the 18 bytes of DOCSIS header yields.

10 KBps in IEEE Floating-Point Format

This time, the QoS parameter set is 0x6, meaning QoS resources are to be admitted and activated. The traffic priority is “5,” making it a higher priority than any service flows left at the default of “0”. The Minimum Reserved Rate Packet Size is 218 bytes. Subtracting the 18 bytes of DOCSIS MAC Management header yields 200 bytes, which is the value of the (b), (m), and (M) RSVP flow specification parameters. Remember, the (b) parameter is expressed in 32-bit IEEE floating-point format. The Maximum Sustained Traffic Rate is 87,200 bits per second, dividing that by 8 yields 10,900 bytes per second. If you then divide that by the Minimum Reserved Rate Packet Size, you get a packet rate of 50 packets per second. Taking this value and then multiplying by the previously calculated packet size

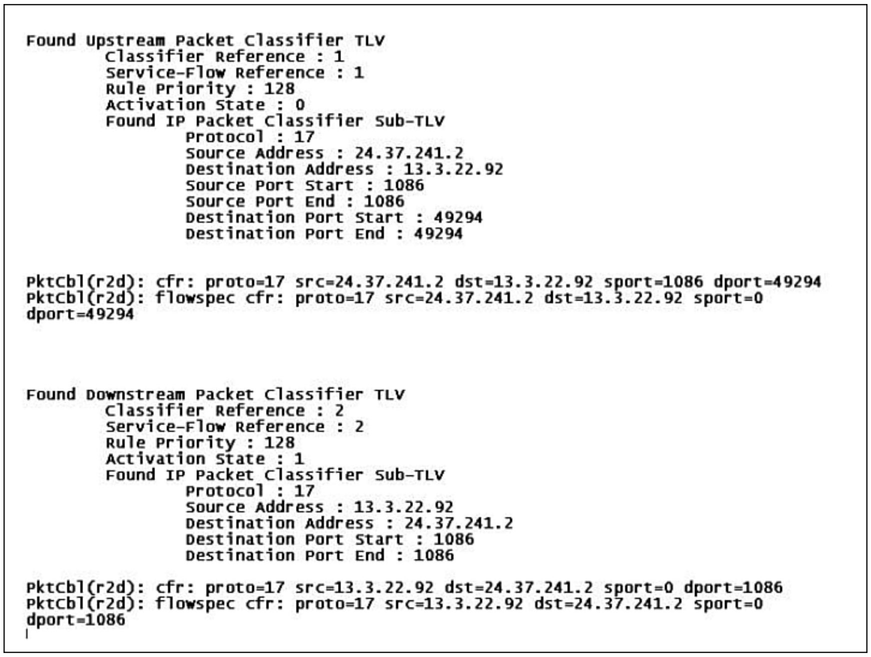
(200 bytes) yields a rate of 10,000 bytes per second; putting this value in 32-bit IEEE floating-point format results in the values shown in the debug.

Mapping DOCSIS Classifiers into Gate Classifiers

The mapping of DOCSIS classifiers into gate classifiers is fairly straightforward. The IP protocol type and source and destination IP address are all direct mappings. The DOCSIS destination port start and end should be equal, and that is mapped to the gate destination port. The gate source port is often "wildcarded" or removed from the classification process by setting it to a value of zero.

Figure 12-20 shows the classifier mappings from the same example using the same debugs.

Figure 12-20 *Mapping DOCSIS Classifiers to Gate Classifiers*



```
Found Upstream Packet Classifier TLV
        Classifier Reference : 1
        Service-Flow Reference : 1
        Rule Priority : 128
        Activation State : 0
        Found IP Packet Classifier Sub-TLV
                Protocol : 17
                Source Address : 24.37.241.2
                Destination Address : 13.3.22.92
                Source Port Start : 1086
                Source Port End : 1086
                Destination Port Start : 49294
                Destination Port End : 49294


PktCbl(r2d): cfr: proto=17 src=24.37.241.2 dst=13.3.22.92 sport=1086 dport=49294
PktCbl(r2d): flowspec cfr: proto=17 src=24.37.241.2 dst=13.3.22.92 sport=0
dport=49294



Found Downstream Packet Classifier TLV
        Classifier Reference : 2
        Service-Flow Reference : 2
        Rule Priority : 128
        Activation State : 1
        Found IP Packet Classifier Sub-TLV
                Protocol : 17
                Source Address : 13.3.22.92
                Destination Address : 24.37.241.2
                Destination Port Start : 1086
                Destination Port End : 1086

PktCbl(r2d): cfr: proto=17 src=13.3.22.92 dst=24.37.241.2 sport=0 dport=1086
PktCbl(r2d): flowspec cfr: proto=17 src=13.3.22.92 dst=24.37.241.2 sport=0
dport=1086
```

As you can see, the IP packet classifier mappings are straightforward. Notice the use of the "activation state" parameter; it has a value of 0 (inactive) for the upstream classifier because this classifier is not yet used because its associated service flow is in the admitted state. The service flow associated with the downstream is admitted and activated at once so this classifier is put into the active state (value is 1) .

Revisiting CMTS Authorization

So after a reserved QoS parameter envelope (indicated by an admitted service flow in a DSA-REQ) or a committed QoS parameter envelope (indicated by an active service flow in a DSC-REQ or DSA-REQ) are converted into RSVP flow specifications, the authorization process can occur.

If the reserved envelope is less than or equal to the authorized envelope and those DOCSIS resources are currently available, the CMTS accepts the DSA-REQ by indicating a positive confirmation code in the DSA-RSP sent back to the MTA. Also included in the DSA-RSP are Service Flow Identifiers assigned to these new service flows. A Service Identifier is also assigned to the upstream service flow. Values for the Timeout for Active parameters and Timeout for Admitted parameter timers can also be included. In this case, only the Timeout for Admitted parameter makes sense at this point and indicates how long the CMTS waits for the CM to activate the resources that have been reserved. The CMTS also assigns Classifier Identifiers to both classifiers. Finally, in the authorization block parameter the DQoS resource identifier, which can be used by the CMS to manage HFC resources and possibly reuse resources, is returned.

Likewise, if the committed envelope is less than or equal to the reserved envelope, the CMTS activates those DOCSIS resources. Notice, the CMTS does not need to check to see if these resources are available as this is assured because of the previous reservation process. The Timeout for Active parameter is included in the response message in this case indicating how long the CMTS allows the service flows to remain idle.

Chapter Summary

PacketCable DQoS enables high-voice quality across the DOCSIS network between a MTA and a CMTS. The embedded MTA, CMTS, CMS, and RKS components all contribute to providing dynamic quality of service. The NCS protocol is used to communicate QoS information between the eMTA and CMS, the COPS protocol is used to communicate QoS information between the CMS and CMTS, and the DOCSIS MAC management protocol is used to communicate QoS information between the eMTA and CMTS.

The fundamental component of PacketCable DQoS is a gate, which is the key to premium DOCSIS resources. A gate contains a policing component for managing resources and a classification component for identifying packets. Other information such as accounting components for tracking and billing resources can also be associated with a gate. PacketCable gates transition through four states: allocation, authorization, reservation, and commitment. The allocation and authorization states occur via COPS-based messaging between the CMS and CMTS. The reservation and commitment states occur via DOCSIS messaging between the embedded MTA and CMTS. These transitions are triggered by NCS call signaling parameters sent from the CMS. Having separate reservation and commit stages provides two-phase activation. This ensures resources are available when needed and that resources are not wasted when not needed.

The CMS communicates DQoS information with the CMTS using the COPS protocol. Additional COPS objects are defined for PacketCable usage. Among these objects are parameters that define gate messages, define gate specifications, identify subscribers, provide billing information, and provide wiretapping information. The COPS protocol uses TCP as its transport layer protocol. DQoS timers minimize the chance resources are wasted or possibly stuck.

DQoS resources are characterized using RSVP flow specifications. The authorized resource envelope is determined by the CMS by interpreting a call's SDP information. Reserved and committed resource envelopes are determined by mapping DOCSIS service flow parameters into RSVP flow specs. The committed resources are always less than or equal to the reserved resources, which in turn are always less than or equal to the authorized resources.

DOCSIS 1.1 provides QoS mechanisms used by PacketCable. Among these mechanisms are dynamic service flows, premium categories of service flows, classifiers, and Payload Header Suppression. The Unsolicited Grant Service (UGS) is the premium type of service flow used by PacketCable. Payload Header Suppression increases DOCSIS bandwidth efficiency by eliminating the repetitive transmission of useless and redundant fields in packet headers.

Chapter Review

1 List the PacketCable components that contribute to providing PacketCable DQoS:

Answer: MTA, CM, CMTS, CMS, RKS

2 What are the four states a gate can pass through? Identify the components involved in putting the gate into each state.

Answer: Allocated (CMS and CMTS), Authorized (CMS and CMTS), Reserved (CMS, eMTA and CMTS), Committed (CMS, eMTA and CMTS)

3 Why is a two-phase activation model important?

Answer: It ensures that resources aren't committed until needed and ensures that resources are available when needed.

4 How is an authorized DQoS resource envelope determined?

Answer: By the SDP information included in the NCS call setup signaling, which is translated into RSVP FlowSpec parameters sent in the COPS signaling to the CMTS

5 How is a reserved DQoS resource envelope determined?

Answer: By DOCSIS service flow parameters

6 How is a committed DQoS resource envelope determined?

Answer: By DOCSIS service flow parameters

7 What DOCSIS extended headers are used in PacketCable?

Answer: BPI+, UGS information, Payload Header Suppression

8 How are the DOCSIS Admitted QoS timeout and Active QoS timeout parameters determined?

Answer: They are set to the DQoS timers T7 and T8, which are set by the CMS in a GATE-SET message.

9 What are the two conditions that must be true for a DOCSIS classifier to be used?

Answer: Classifier state must be active, and associated service flow must be active.

10 What is the expected UGS size if the G.711 codec and a 10-ms packetization period are being used?

Answer: 152 bytes (80 bytes RTP payload, 40 bytes RTP/UDP/IP header, 32 bytes DOCSIS header)

11 What DOCSIS parameter is used to pass DQoS gate information between the eMTA and CMTS?

Answer: Authorization Block

12 What field of a PacketCable COPS gate specification can be used to identify an emergency 911 call?

Answer: Session class

13 How does the CMTS figure out the timer used to send the COPS keepalive message?

Answer: The CMS sends this as a parameter in the COPS Client-Accept (CAT) message during initialization.

14 What is the QI bit in the UGS extended header used for?

Answer: It is used by the MTA to let the CMTS know it has a backlog of voice packets in its transmit queue and needs extra grants.

15 Match the DQoS timers (T0, T1, T5, T7, T8) to their definitions:

A: Time a gate can remain in the authorized state before moving to the reserved state

B: Time a gate can remain in the allocated state before moving to the authorized state

C: Time a gate can remain in the authorized state before moving to the committed state

D: Time between a NCS DLCX is sent to a MTA and the resulting GATE-CLOSE message is received from the CMTS

E: Time committed resources can remain unused before they are released

F: Time reserved resources can remain so before they are released

Answer: T0 - B, T1 - C, T5 - D, T7- F, T8 - E

16 True or False: QoS for signaling packets as well as bearer packets is covered by PacketCable DQoS.

Answer: False. PacketCable DQoS covers only the media stream.

17 True or False: Although a gate ID often refers to two service flows, a DQoS gate is unidirectional.

Answer: True

18 True or False: DOCSIS Packet concatenation and fragmentation are permitted options for voice packets.

Answer: False

19 True or False: The DOCSIS admitted and active states correspond to the DQoS reserved and committed states.

Answer: True

20 True or False: The PHSF field for an upstream packet and a downstream packet are composed of the same header fields.

Answer: False. Upstream PHSF field begins with the Destination MAC address, and downstream PHSF field begins with the Ethernet Type field.

21 True or False: The CMTS can also be referred to as a COPS Policy Decision Point (PDP).

Answer: False. The CMTS is the COPS Policy Enforcement Point (PEP); the CMS is the PDP.

22 True or False: GATE-OPEN and GATE-CLOSE commands are sent from the CMTS to the CMS to ensure proper gate coordination.

Answer: True

23 True or False: PacketCable COPS connections are initiated by the CMTS.

Answer: True