Chapter

3

The Access Link

The KEK is a two-key triple DES encryption key that the CMTS uses to encrypt Traffic Encryption Keys (TEKs) it sends to the modem. Traffic encryption keys are used for encrypting user data traffic. CM and CMTS use message authentication keys to authenticate, via a keyed message digest, the key requests and responses they exchange.

SP-BPI+-I02-990731

In order for a user to access a service offered by a provider—whether that service is telephony or simple data access—there must be a communications link between that user and the service provider's facilities. On a cable network, that link is implemented through the cable modem, or CM, located in the user's residence, and the Cable Modem Termination System, or CMTS, located in the headend. All traffic between the user and the network travels over this CM-CMTS link.

The CM-CMTS link is not symmetric. Cable modems are not at all like the ordinary analog modems commonly used for low-speed access over analog telephone lines. Rather, they are complex devices that act as clients to the CMTS, which in turn directs in real-time exactly how the each CM on the access network is to behave.

The DOCSIS Specifications

Early cable modems were designed according to specifications developed by individual manufacturers using proprietary protocols. Although many of these modems worked well, because each manufacturer adopted a different protocol it was impossible for them to interoperate. Since the cable modem communicates with a CMTS, this meant that once a service provider decided on a particular vendor's CMTS, its customers were immediately locked into using only cable modems from the same vendor. Because several vendors were competing to produce CM-CMTS pairs, the market was fragmented and the hardware was relatively expensive.

In order to standardize the CM-CMTS protocols, a consortium of cable operators was formed. This consortium was called the **Multimedia Cable Network System**, or **MCNS**,[1] and its stated goal was to prepare "a series of interface specifications that will permit the early definition, design, development and deployment of data-over-cable systems on an [*sic*] uniform, consistent, open, non-proprietary, multi-vendor interoperable basis". These specifications are collectively referred to as the **Data-Over-Cable Service Interface Specifications** or "**DOCSIS**".

Note that the original emphasis was on *data* over cable. The first version of DOCSIS was designed to support only ordinary data communication. Telephony

Table 3-1 DOCSIS Specifications

SP-BPI+-I02-990731 Baseline Privacy Plus Interface Specification—Specifies security over the cable access network.
SP-CMTRI-I01-970804 Cable Modem Telephony Return Interface Specification—Specifies the use of telephone lines for upstream information flow.
SP-CMTS-NSII01-960702 Cable Modem Termination System–Network Side Interface Specification—Specifies how the network interfaces with HFC.
TR-DOCS-OSSIW08-961016 Operations Support System Framework for Data Over Cable Services—Specifies a high-level framework for OSS for data services over cable.
SP-RFIv1.1-I02-990731 Radio Frequency Interface Specification—Provides a low-level description of communication between a cable modem and the Cable Modem Termination System.
SP-OSSI-RFI-I03-990113 Operations Support System Interface Specification Radio Frequency Interface—RF Interface MIBs
SP-OSSI-BPI-I01-980331 Operations Support System Interface Specification Baseline Privacy Interface MIB–Privacy MIBs

1. The original members of MCNS were Comcast Cable Communications, Inc., Cox Communications, Tele-Communications, Inc., Time Warner Cable, MediaOne, Inc., Rogers Cablesystems Limited, and Cable Television Laboratories, Inc. (acting on behalf of the CableLabs member companies).

communication has requirements over and above those needed for data (in particular, telephony requires guaranteed Quality of Service and enhanced privacy); support for these requirements was added in version 1.1 of the DOCSIS specifications.

The current version of the DOCSIS specifications may be downloaded from *www.cablemodem.com*. Table 3-1 lists the versions that were current at the time this book was written. The DOCSIS specifications are intended to define the behavior of Cable Modem and Cable Modem Termination System devices in sufficient detail that products conforming to the specifications will be interoperable. The specifications are not designed to imply any particular method of implementing these devices.

PacketCable telephony is designed to run over DOCSIS version 1.1 or later. Theoretically, the entire PacketCable network is independent of the underlying layers and infrastructure. In theory, PacketCable could be implemented to run over a completely different technology such as DSL or wireless. However, when the PacketCable telephony network was being designed, the strengths and weaknesses of DOCSIS 1.1 were taken into account, so that many of the design decisions reflect the assumption that DOCSIS 1.1 is being used over the access network.

The relationship between PacketCable and DOCSIS is shown in Figure 3-1. All current implementations of PacketCable use the Quality of Service "hooks" provided by DOCSIS and discussed later in this chapter. In principle, DOCSIS and PacketCable can be completely decoupled: PacketCable could be built on a completely different access technology. Similarly, a totally different telephony architecture could be constructed on top of DOCSIS. However, as of this writing, there is no

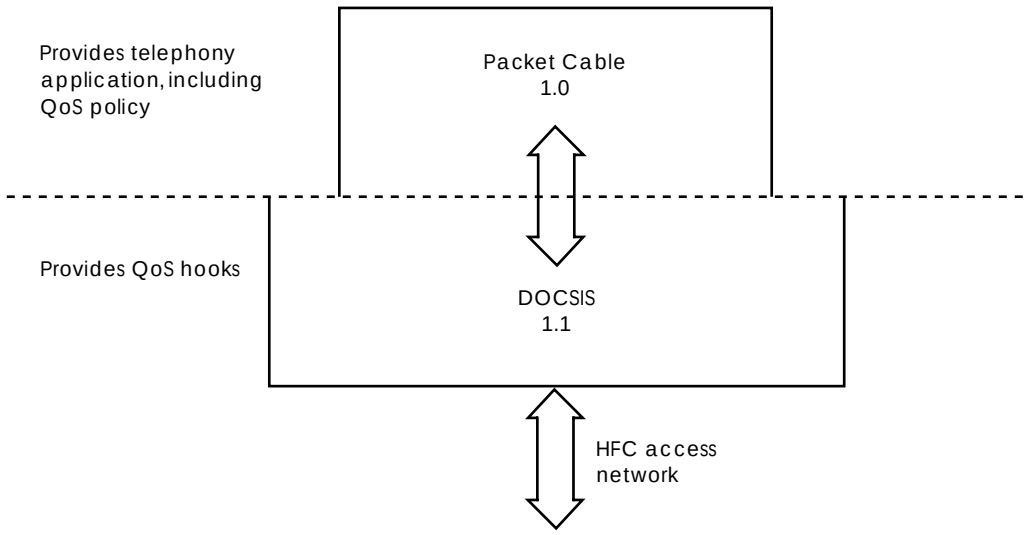


Figure 3-1 DOCSIS and PacketCable

indication that any vendor intends to separate the PacketCable telephony "application" from the underlying DOCSIS transport.

Note that in this chapter, we will be discussing the behavior of cable modems, not of MTAs. In the current release of PacketCable, these are intended to be embedded in the same device, since there is no standard API that allows a cable modem to be driven by an external MTA. This sometimes leads to a certain level of sloppiness when referring to CMs and MTAs. However, in the future MTAs and CMs will migrate to become physically distinct entities and a clear distinction will have to be made. The cable modem will remain the point where the home network interfaces to the cable, and it will continue to function as described in this chapter.

Overview of the Cable Access Network

Before we examine in detail the workings of DOCSIS modems and their corresponding CMTSes, we will look briefly, at a high level, how a DOCSIS access network operates.

DOCSIS cable modems located in homes are clients of Cable Modem Termination Systems, which are located at the other end of the coax/fiber access link, at the MSO's headend. In order that the CMTS is properly able to manage the access network, on which there may be several hundred cable modems all competing for the available upstream and downstream bandwidth, DOCSIS cable modems are required to obtain and obey instructions from the CMTS. This ensures that the resources are allocated fairly and efficiently among the active cable modems on the network.

Initialization

When a cable modem is first connected to the cable and powered up, a complex series of transactions takes place between the modem and its CMTS. The modem is at all times unaware (at least directly) of the presence of other modems on the network; the modem communicates *only* with the CMTS. This is true even if two modems on the same access network wish to communicate—all traffic passes through the CMTS.

The cable modem initialization sequence is as follows.

1. Locate a downstream channel and synchronize operation with the CMTS.
2. Obtain upstream transmit parameters from the CMTS.
3. Perform a ranging operation.
4. Confirm that IP connectivity exists.
5. Synchronize time of day with the CMTS.
6. Transfer operational parameters between CM and CMTS.

7. Register.
8. Initialize Baseline Privacy Plus.

In order to perform this sequence in a reliable and secure manner, two items are placed in the cable modem at the time of manufacture. These items are in non-volatile memory and should never be changed.

- A unique 48-bit MAC address (equivalent to the MAC address in an ordinary Ethernet network interface card)
- An X.509 digital certificate, which is used to authenticate the modem to the CMTS. Typically, this certificate is signed by the modem manufacturer, whose public key the service provider has obtained through other means and that is loaded into the CMTS software so that it can authenticate modems as they attempt to initialize and obtain service.

Downstream Synchronization

The cable modem begins to scan the 6 MHz downstream video channels, looking for a CMTS signal. If the modem has previously been used and is simply restarting after a temporary failure—for example, power-down—the modem first tries to lock on to a CMTS signal in the most recently used downstream channel. It continues to scan until it finds a signal that it can properly detect and with which it can properly synchronize.

Obtaining Upstream Parameters

The CMTS periodically transmits messages called **Upstream Channel Descriptors** (**UCD**s) on all downstream channels. Nominally, UCDs are broadcast every two seconds. UCDs describe the correct parameters that a modem must use to transmit on the various upstream channels to which the CMTS is currently listening.

When the modem receives a UCD containing parameters for a channel that it can use, it stores this information and uses it to determine the transmit parameters for future upstream transmissions.

As well as UCDs, the CMTS periodically transmits **SYNC** messages (nominally every 200 milliseconds). These contain information about the CMTS's notion of time and allow the modem to synchronize properly with the CMTS and the other modems on the network.

Ranging

A number of operational parameters within the modem may need to be adjusted slightly in order to guarantee that all modems on the access link are operating

cooperatively. For example, the transmit power level or the center frequency of the upstream channel might need to be adjusted slightly if the modem is out of alignment. In addition, since cable modems are not all at the same distance from the CMTS, it is insufficient for a CM merely to synchronize its clock with the CMTS. It must also have some notion of the transmission delay between itself and the CMTS, otherwise the transmissions from two modems, one told to transmit at time *t* and another told to transmit at time *t'*, might overlap.[2]

In order to make these adjustments, the cable modem must actively exchange information with the CMTS. It does this through a process known as **ranging**. Cable modem transmissions are sent in one of two modes: contention or noncontention. In the cable system, time is divided into short intervals known as minislots, which are a small multiple of 6.25 microseconds in length. (The precise duration of a minislot depends on the modulation scheme in use. Basically, a minislot is usually the time taken to transmit 16 octets.) Noncontention minislots are allocated by the CMTS in such a way that only one CM is permitted to transmit within the minislot. Transmissions occurring in noncontention minislots have a high probability of being received correctly at the CMTS, since it is guaranteed that there will be no other signal on the line in the same upstream channel at the same time. Contention minislots (which are typically about 25% of the available total) are unallocated, and any CM is permitted to transmit during them. These transmissions may have a low probability of being received correctly if there are many active devices on the access network.

The CMTS manages the ratio of contention to noncontention minislots, just as it manages exactly which modem may transmit during a noncontention minislot. In fact, at the risk of digressing from the point at hand, calculating optimum ratio of noncontention to contention minislots is an interesting problem in network bandwidth management, since it depends on the kind of data that is passing across the network. If most of the traffic flows at relatively constant rates (for example, when the network is handling principally telephony traffic), then there are fewer ad hoc requests for upstream bandwidth, and the need for contention minislots decreases. This in turn allows the CMTS to allocate more bandwidth to noncontention minislots, and thus even more telephony-like traffic may be permitted to flow. If, on the other hand, the traffic is "bursty", such as occurs with Web browsing, then the number of contention minislots typically needs to be increased and the usable bandwidth

2. Upstream bandwidth is an extremely scarce resource on the access network. The modems and the CMTS go to great lengths to use the available bandwidth as efficiently as possible. Part of this process is to ensure that all the devices on the network maintain very closely synchronized clocks to reduce packet collisions on the network.

of the system decreases. A good working average of noncontention:contention mode slots for "typical" traffic is roughly 3:1.

Except for informational messages, transmissions sent in contention mode usually demand an explicit response from the CMTS. If the expected response is not received, the CM will usually retransmit the transmission in another contention-mode minislot, and will continue to do so until a response is received.

Ranging requests are sent in contention mode and so may need to be repeated a number of times before the CM receives the information it desires from the CMTS. In response to a ranging request, the CMTS will instruct the CM to adjust parameters such as clock skew, carrier frequency and transmit power so that they are within acceptable limits.

In addition to the ranging performed during initialization, the CMTS provides specific opportunities for each attached CM to perform subsequent ranging operations to ensure that slight adjustments to the operational parameters may be made as necessary, so that the entire system stays acceptably synchronized.

Establishing IP Connectivity

Once the low-level transmission parameters are properly set, the CM should be able to communicate correctly with the CMTS (and, through it, to the MSO's network on the far side of the CMTS). It now begins communication by transmitting a **Dynamic Host Configuration Protocol** (**DHCP**) "discover" request. In response, a DHCP server provides the modem with an assigned IP address, as well as the address of another DHCP server (possibly the same one) that can provide the modem with more parameters. The initial DHCP response also contains name of a file that contains further, network-specific configuration parameters for the CM. The CM issues a DHCP request to the second server and obtains whatever additional parameters are needed to establish IP connectivity with the network. Note that it does not yet download the configuration file.

Synchronizing Time of Day

As well as a low-level shared notion of time (for the correct synchronization of packet transmissions), the CM and the CMTS need to share a common notion of the approximate time of day, which may be used for logging abnormal events and for key management by the security system (which will typically require that keys be changed periodically).

One of the parameters obtained from the DHCP server is the address of a Time Server (which may be the DHCP server itself). The modem connects to this server on port 37 and obtains the time, using the Time Protocol specified in RFC 868.

Transferring Operational Parameters

The CM now downloads the configuration file whose name was provided by the original DHCP server. This download uses the Trivial File Transfer Protocol specified in RFC 1350. The operational parameters overwrite any default values configured into the modem during manufacture.

A large number of parameters may (but need not) be present in the configuration file. These parameters provide values used by the low-level system, such as upstream and downstream channel frequencies and data rates, as well the addresses of various network servers, timer values, and so on. If explicit values are not provided, the modem adopts sensible default values provided at the time of manufacture.

The configuration file may direct the modem to use an upstream or downstream channel different from the one it is already using, in which case the modem switches to the new channel(s) and performs a new Ranging request.

Registering

Once the modem has obtained and processed the configuration file, it informs its CMTS of the values of its operational parameters in a Registration Request message. The CMTS assigns Service IDs (SIDs), which will be used to identify the various classes of service flowing through this particular modem and informs the modem of the SID values that have been assigned to it.

Initializing Baseline Privacy Plus

A security association between a cable modem and its CMTS allows information to flow between the two without fear that the data can be read or manipulated by a third party. This is an important requirement on a cable access network, since there is at least a theoretical possibility that a neighbor may be eavesdropping on the CM-CMTS communication.

In order to create a security association, the modem now initializes its Baseline Privacy Plus (BPI+) configuration, which effectively secures the link from casual eavesdroppers. (BPI+, however, uses only 56-bit DES to secure the link, which is insufficient to deter a determined attempt to decrypt the traffic.) Once BPI+ is correctly initialized, the modem is a fully fledged member of the network, operating completely under the control of the CMTS.

DOCSIS Protocol Layers

Figure 3-2 shows the protocol layers included in the DOCSIS specifications. They range from the Physical Media Dependent sublayer, which carries modulated **RF** (**Radio Frequency**) energy, to a layer carrying some of the network administration

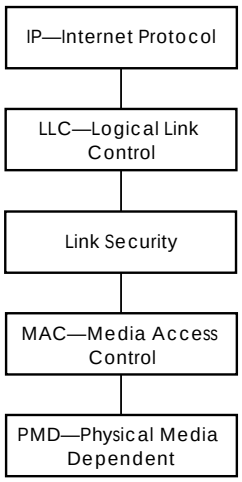


Figure 3-2 DOCSIS Protocol Layers

protocols used in an IP network. In this chapter we will concentrate on discussing the lower layers, since these are specific to cable and adequate descriptions cannot easily be found elsewhere.

Physical Media Dependent Sublayer

At the bottom of the DOCSIS stack is the **Physical Media Dependent** (**PMD**) sublayer. Communications textbooks describe many ways to modulate waveforms to allow information to be transmitted from a source to a destination. Common modulation methods, with which you are probably familiar, include **Amplitude Modulation** (**AM**) and **Frequency Modulation** (**FM**), which are used for ordinary broadcast radio transmission, but there are also many other modulation schemes used for other purposes. Each scheme has characteristics that may make it useful in one set of circumstances, whereas another scheme may be better suited to different circumstances.

Some modulation schemes, for example, are relatively immune to extraneous noise, whereas others may work rather badly in the presence of noise. Others allow for very rapid information flow, whereas some may operate relatively slowly. The schemes used in cable systems are well adapted to that particular environment, and allow relatively high rates of information flow in the cable access network.

Modulation Schemes

All modulation schemes begin with the notion of a pure monochromatic (single frequency) sine wave such as the one depicted in Figure 3-3.

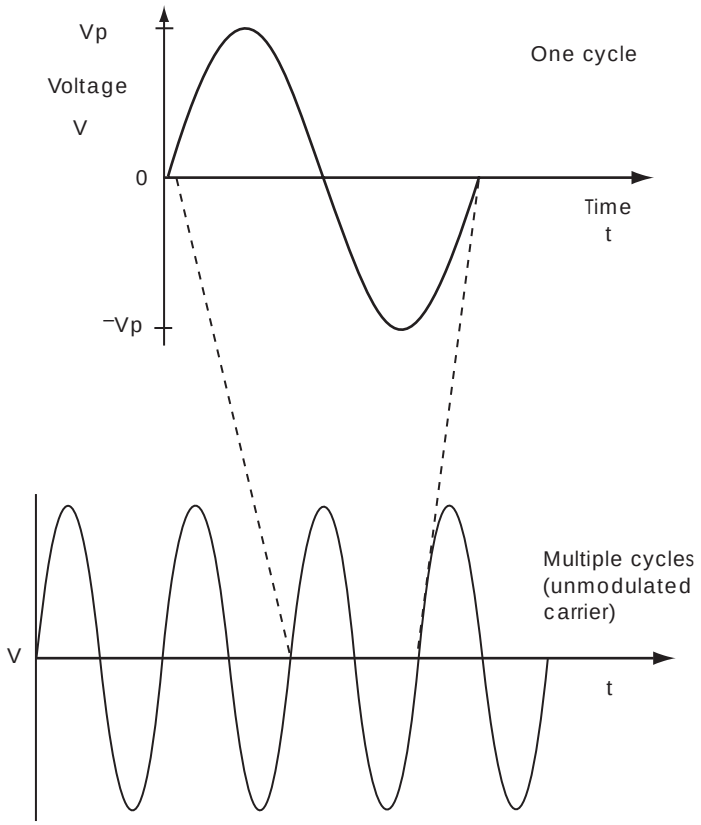


Figure 3-3 Sine Wave

The amplitude of a sine wave (typically measured in volts or millivolts) varies smoothly in time: It starts from zero, crests at a value V_p, decreases again to zero, continues decreasing until it reaches a peak negative value, $-V_p$, and then returns again to zero. This cycle is repeated many times, as shown in Figure 3-3.

The number of complete cycles that occurs in a second is called the frequency of the wave. For the kinds of waves that we will be talking about in this book, the frequency ranges from a few million per second to a few hundreds of million per second. Each cycle per second is called a Hertz, abbreviated as Hz; millions of cycles per second are called megahertz, abbreviated as MHz.[3]

3. The frequency range of naturally occurring waves is extremely wide: Micropulsations, which have their origin in the Earth's magnetosphere, have frequencies below 1 Hz. Ordinary light ranges in frequency from 1.4×10^{14} Hz (red) to 2.5×10^{14} Hz (blue); gamma rays have frequencies as high as 10^{34} Hz. Isn't it wonderful what subjects can turn up in a book about telephony?

A sine wave of a particular frequency carries no information, but it can be modified by a process called **modulation**: The information transmitted is used to deform it from a pure sine wave so that once the wave reaches a distant receiver, the amount of deformation can be measured, and the sender's information can be recovered through **demodulation**. This is shown in Figure 3-4. Because the original sine wave, although it contains no information, is used to *carry* information, it is called a carrier, and the frequency of the carrier is called the carrier frequency.

Most services that utilise carrier waves adopt a channelized bandplan, in which a range of frequencies is broken up into a number of channels, and each channel contains exactly one carrier frequency. For example, the domestic AM broadcast band is

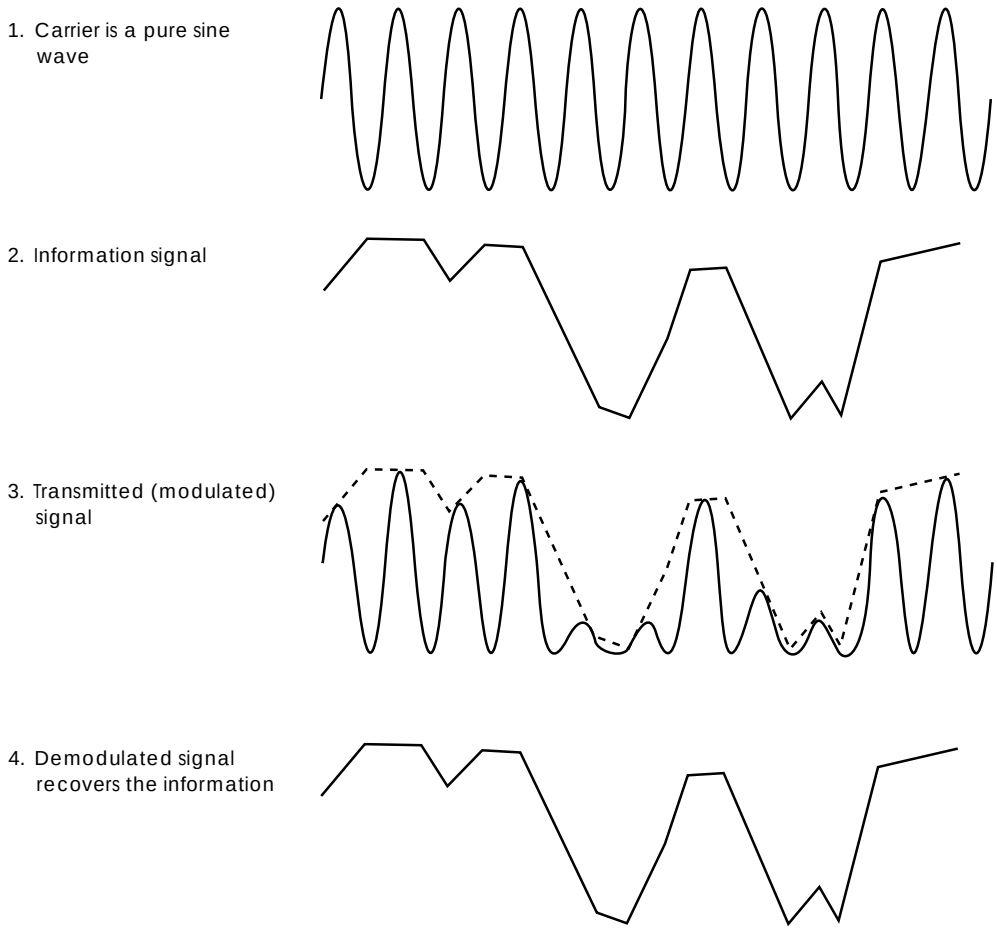


Figure 3-4 Modulating a Carrier

assigned by the **Federal Communications Commission** (**FCC**) to frequencies between 535 and 1605 kHz. However, if stations were to place their carrier frequencies wherever they desired in this range, they would interfere with one another. To prevent this, the FCC mandates a bandplan of fixed channels spaced by 10 kHz and ensures through a licensing system that no two stations in the same geographical area are assigned the same carrier frequency. So, for example, an AM broadcast station may be assigned to the 590 kHz channel (imagine an ugly jingle here, followed by a cheery voice announcing, "KQIQ AM 590"), but, since channels are separated by multiples of 10 kHz, there cannot be a station on 595 kHz.[4]

The FM broadcast band operates under a similar scheme, although in that case the band edges lie at 88 and 108 MHz, and the channels are spaced by 200 kHz. The reason for the greater separation between carrier frequencies is that frequency modulation, although it is more immune to noise than AM—which is why most stations broadcasting music use FM—it also occupies much greater bandwidth than AM. This is an example of different modulation schemes being used for different purposes in different environments.

Broadcast radio is, clearly, quite different from two-way digital data communication over a cable. Although the two share the fundamental characteristic of modulating a carrier wave to transmit information, the two services have different requirements: Broadcast is analog. It is relatively unconcerned with noise and reliability, and information is transmitted at a relatively low rate. By contrast, data communication is digital. It is important that the transmissions be reliable and untainted by errors induced by noise, and the information flow may be quite high-speed. Therefore it is unsurprising that different (and more complicated) modulation techniques are used to transfer data than the relatively simple techniques used in broadcast radio.

The greatest single difference lies in the digital nature of the information. Digital data may always be reduced to a stream of zeros and ones. Therefore it is necessary to modulate the carrier only in such a way as to be able to distinguish between the "zero" state and the "one" state, and then to be able to switch quickly between these states.

Simple modulation schemes such as AM can be operated in this way. For example, in standard ASCII computer code, the uppercase letter A is represented by the 8-bit string 01000001. In a simple digital amplitude modulated system, the waveform corresponding to the letter A might look as in Figure 3-5.

A more common modulation method used for transmission of digital data is **Phase Shift Keying**, or **PSK**. Instead of varying the amplitude of the frequency or the carrier, its phase is shifted according to some well-defined scheme. Figure 3-6

4. The channel spacing in other countries may be different. Since signals in the so-called "medium wave" broadcast band—535 to 1605 kHz—typically do not travel very far, it is not necessary for every country to adopt the same channelized bandplan. This is not true for all frequencies.

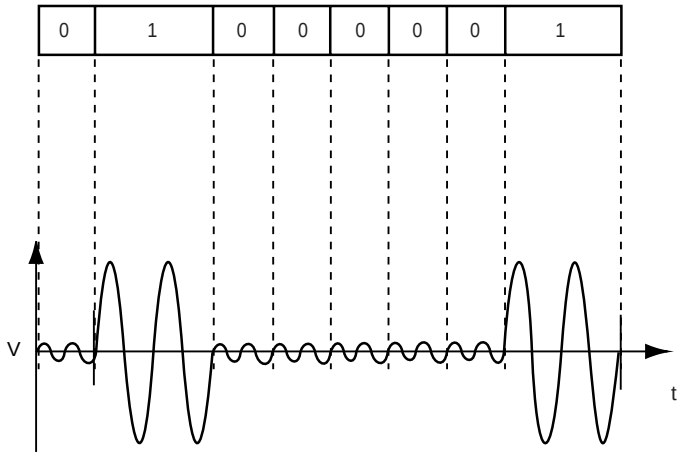


Figure 3-5 Representation of the ASCII Character "A" in Digital AM

shows the letter A in a PSK scheme in which a phase shift of 180° represents a binary one and a phase shift of 0° represents a zero.

A relatively simple variant of ordinary PSK, called **Quadrature Phase Shift Keying** or **QPSK**, is often used in modems. QPSK is a modulation scheme that, although rather inefficient, is also more robust (that is, less prone to errors on noisy channels) than more complicated schemes.

In QPSK, a single carrier is split into two components phased 90° apart. (Two signals with the same frequency but 90° apart in phase are said to be *in quadrature*.) The two components are known as the I-channel and the Q-channel. Dividing the carrier in this way allows us to send twice as much information in the same amount of time, since the two channels are independent of each other.

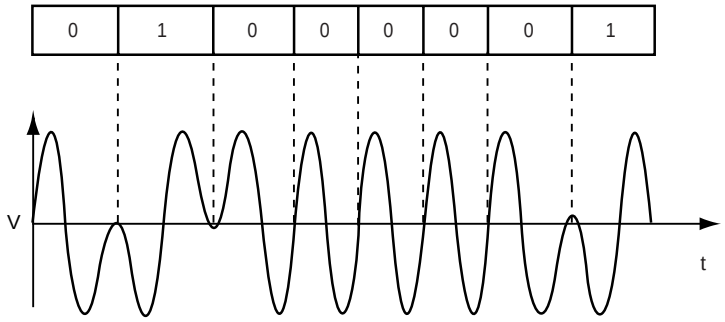


Figure 3-6 Representation of the ASCII Character "A" in Digital PSK

Consider first just the I-channel. We can modulate this exactly the same way as we did in ordinary PSK: A phase shift of 180° represents a binary 1, and a phase shift of 0° represents a binary 0. But we can also do the same for the Q-channel: Adding a phase shift of 180° to the Q-channel (which is already shifted +90° relative to the I-channel) represents a binary 1, and a phase shift of 0° represents a binary 0.

Thus we can transmit bits in pairs, allowing the first bit of the pair (for example) to modulate the I-channel, and the second (for example) to modulate the Q-channel. The I-channel and the Q-channel signals are combined before transmission and are transmitted simultaneously. At the receiving end, a demodulator splits the incoming signal into I-channel and Q-channel, and then examines each channel to determine whether it is at phase 0° or phase 180°. From this measurement, the original pair of bits may be recovered.

QPSK demonstrates an important point about digital transmission systems that is often not clearly understood—that is, that the bit rate of a communication is not always equal to the baud rate. The baud rate is defined as the rate at which individual symbols are transmitted. In QPSK, the carrier may be in one of four states, corresponding to the waveform for 00, 01, 10 and 11, respectively; each state is referred to as a symbol. The baud rate in a QPSK system is therefore the number of (00, 01, 10, 11) symbols transmitted per second. Since each symbol represents two independent bits, the bit rate of a QPSK transmission is twice the symbol, or baud, rate.

The different states in QPSK (as well as other quadrature modulation schemes) are often shown in what is called a constellation diagram, in which each possible symbol is marked with a dot (see Figure 3-7).

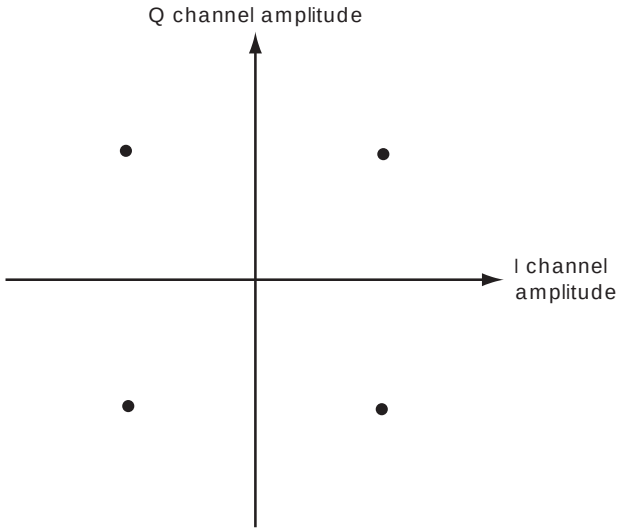


Figure 3-7 QPSK Constellation Diagram

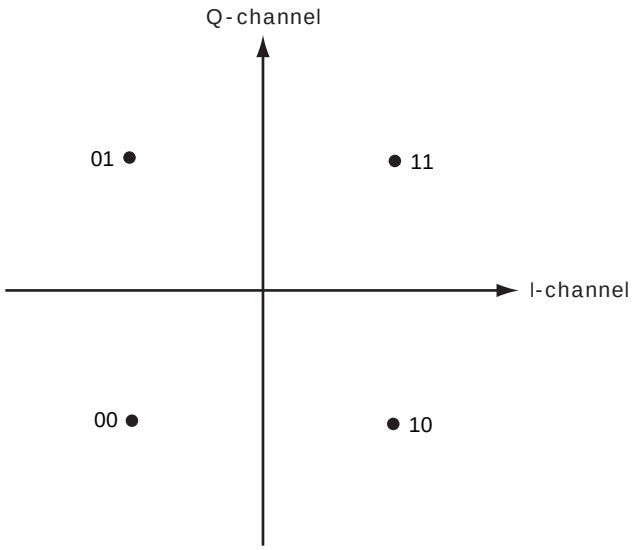


Figure 3-8 DOCSIS QPSK Symbol Mapping (I, Q)

Naturally, both the transmitter and the receiver must agree on the meaning of each symbol. DOCSIS mandates the symbol mapping shown in Figure 3-8, where the I-channel bit precedes the Q-channel bit.

Another common modulation scheme is known as 16-QAM. (**QAM**, as you may be able to guess, stands for **Quadrature Amplitude Modulation**.) This uses the same basic mechanism of splitting the carrier into an I-channel and a Q-channel and shifting the phase of the two channels, but in addition it applies amplitude modulation independently to the two channels. Suppose that the channel is such that we can reliably detect two different levels of amplitude, then the corresponding constellation diagram looks like Figure 3-9.

The total number of different symbols in this system, N, is given by:

```
N = number of phase axes *
       number of distinguishable phases per axis *
       (number of amplitude states)^2
```

or, putting in the numbers:

```
N = 2 * 2 * 2^2
  = 16; hence, 16-QAM.
```

Even more efficient modulation schemes are possible if the communication channel is noise-free and the modulator and demodulator are capable of reliably

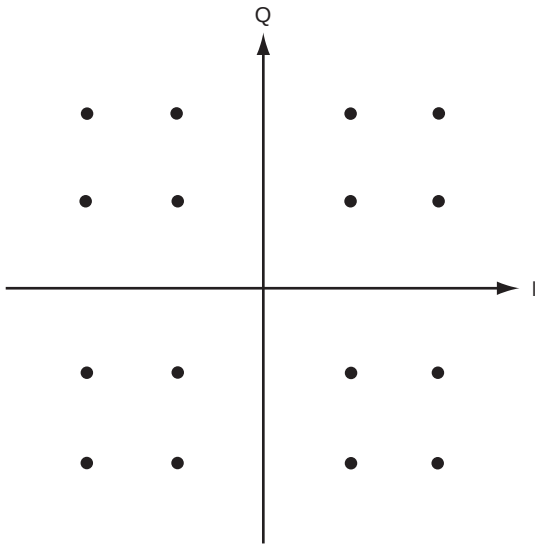


Figure 3-9 Constellation Diagram for Two-Amplitude Quadrature PSK (16-QAM)

distinguishing a greater number of amplitude levels. Both 64-QAM (four amplitude states) and 256-QAM (eight amplitude states) are commonly used where circumstances permit.

Just as the bit rate is twice the baud rate in ordinary QPSK (which is sometimes called 4-QAM), the following relationships hold for the more efficient modulation schemes in Table 3-2.

On most cable systems, upstream communication takes place in the frequency range between 5 and 40 MHz. This tends to be a rather noisy part of the electromagnetic spectrum.[5] Because of the relatively noisy nature of the upstream channel, cable modems must be capable of transmitting only QPSK and 16-QAM. The noise level in the range 5 to 40 MHz is generally too great for 64-QAM and 256-QAM to be useful. 64-QAM and 256-QAM are, however, used in the higher-frequency downstream direction, where the noise is typically much less. When operating in 16-QAM mode, DOCSIS-compliant modems are required to follow the symbol diagram given in Figure 3-10.

The 16-QAM mode of DOCSIS modems can also be programmed to operate according to a symbol mapping in which the transmitted symbol depends on the

5. Theoretically, the cable is sufficiently isolated by the braiding that no noise can enter. In practice a small amount of noise does enter, and the amount that enters increases with decreasing frequency.

Table 3-2 Bits per Symbol for Various Modulation Schemes

Modulation Scheme	*Bits per Symbol (Bit Rate ÷ Baud Rate)*
16-QAM	4
64-QAM	6
256-QAM	8

previously transmitted symbol, a method that is called differential symbol mapping, or differential coding. The constellation diagram for DOCSIS 16-QAM differential coding is shown in Figure 3-11.

Table 3-3 shows how the quadrant of the about-to-be-transmitted symbol is derived from the current bits. (The abbreviation **MSB** in the table stands for **Most Significant Bit**; similarly the abbreviation **LSB** is commonly used to mean **Least Significant Bit**.) The combination of Figure 3-11 and Table 3-3 tells us what symbols must be transmitted and how much phase change to apply to the transmission. Note that Table 3-3 shows us that the prior symbol does not affect the phase (although it does affect which bits are transmitted, as Figure 3-11 tells us).

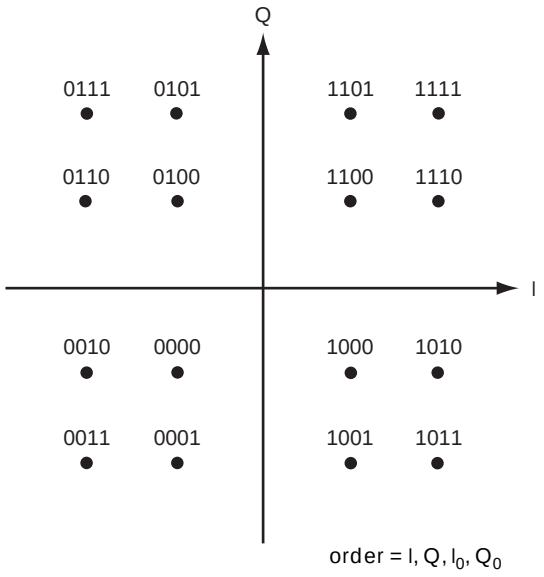


Figure 3-10 16-QAM DOCSIS Symbol Mapping (I_1, Q_1, I_0, Q_0)

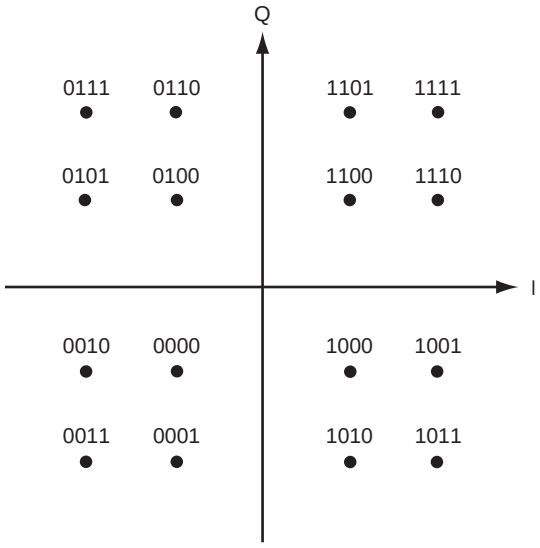


Figure 3-11 16-QAM DOCSIS Differential Symbol Mapping (I_1, Q_1, I_0, Q_0)

As an example, suppose that the last symbol that the modem transmitted was the one marked (1, 0, 1, 0) in Figure 3-11 and that the next symbol will represent the four bits 1110. The current input bits (I_1, Q_1) are 11. The MSBs of the prior symbol were 10. Then Figure 3-11 and Table 3-3 tell us that the MSBs for the currently transmitted symbol must be 01 and the quadrant must change phase by 180°. The LSBs for the currently transmitted symbol are identical to the LSBs of the input signal (that is, differential encoding is used only for the MSBs).

Time Slices

Data transmission in a cable system is synchronous. This means that events are linked to carefully synchronized clocks running within the cable modem and its corresponding cable modem termination system. Time slices are allocated to one or more transmitting modems in a process called **Time Division Multiple Access** (**TDMA**). The unit of temporal granularity in a DOCSIS cable system is 6.25 microseconds.

Data are transmitted and received in indivisible time slices called **minislots**, each of which is an integral power-of-two multiple of 6.25 microseconds (1 × 6.25 microseconds, 2 × 6.25 microseconds, 4 × 6.25 microseconds, 8 × 6.25 microseconds and so on). If the data do not exactly fit within a minislot, the time up to the next available minislot boundary is effectively wasted, since minislots are always synchronized across the network. A single packet of data may occupy several contiguous minislots, depending on its length.

Table 3-3 Derivation of Quadrant in DOCSIS Differential 16-QAM

Input Bits I_1Q_1	*MSBs of Prior Symbol*	*Phase Change*
00	11	0°
00	01	0°
00	00	0°
00	10	0°
01	11	90°
01	01	90°
01	00	90°
01	10	90°
11	11	180°
11	01	180°
11	00	180°
11	10	180°
10	11	270°
10	01	270°
10	00	270°
10	10	270°

When the modem is using QPSK modulation, a total of 64 symbols (corresponding to 16 octets, or 128 bits) can be transmitted in a single minislot.[6] The duration of a minislot on a particular channel, expressed as a multiple of 6.25 microseconds, is present in Upstream Channel Descriptors transmitted periodically by the CMTS.

The CMTS keeps track of time with an internal 32-bit counter (which wraps silently back to zero after reaching $2^{32} - 1$) synchronized to a 10.24 MHz clock. The value of this counter is used to synchronize transmissions by modems throughout the access network.

Both QPSK and 16-QAM are supported on the upstream channel, with symbol rates of 160, 320, 640, 1,280 and 2,560 kilosymbols per second. The highest upstream bit rate that a DOCSIS modem can support is therefore 2,560 kilosymbols times 4 bits per symbol (16-QAM modulation), for a total of 10.24 megabits per second.

6. In the less common case where the modem is using 16-QAM, the modem may transmit 32 octets in a single minislot.

Upstream Transmission

The actual procedure of transmitting a data packet is quite complicated; it is diagrammed in Figure 3-12. We will briefly look at the various values and steps shown in Figure 3-12.

1. Data to be transmitted

   This is simply the string of zeros and ones that the CM wishes to transmit at this time.

2. Block the data

   The data are divided into smaller units called Information Blocks.

3. Apply optional FEC

   In any transmission medium, there is a possibility that bits may be lost or recovered incorrectly in the receiver. To guard against this, CMs may use a system called **Reed-Solomon Forward Error Correction**. This process takes an Information Block and adds bits to it in such a way that if a few consecutive bits are lost or scrambled in transmission, they can be recovered at the receiver.[7] These bits are sometimes called FEC Parity bits. The Information Block plus the FEC Parity is called a codeword.

4. Scramble

   Many electronic circuits at the receiver assume that all DC bias (that is, a "long-term" non-zero voltage) has been removed from a signal. Long strings of zeros or ones in the data stream may result in a short-term apparent DC bias, which can cause receiver circuitry to misinterpret the data. To prevent this, each codeword is scrambled prior to transmission by XORing it with a pseudo-random sequence of bits. The receiver XORs the received data with the same

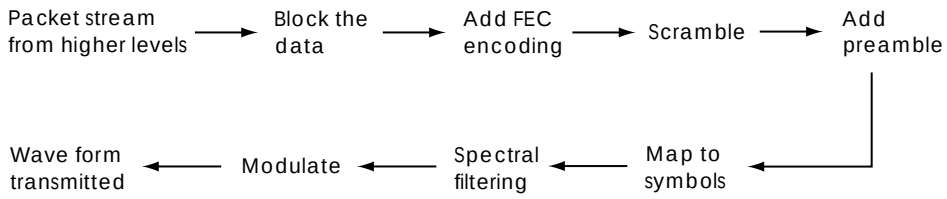


Figure 3-12 Upstream Data Flow Through a Cable Modem

7. The number of bits added and the amount of protection afforded by FEC encoding at the CM are ordered by the CMTS during the initialization sequence. Remember that once it is on the network, the CM can do nothing unless it is explicitly ordered to do so by the CMTS.

pseudorandom sequence, effectively recovering the original codeword. (See Chapter 2 for more details about the XOR operation.)

5. Preamble prepend

   A preamble is placed at the beginning of the packet. This is a sequence that will help the CMTS synchronize correctly so that it recognizes that data are about to be received. The length and contents of the preamble are sent to the CM by the CMTS during initialization.

6. Symbol map

   The bitstream is converted by the process described in Modulation Schemes into the corresponding stream of symbols.

7. Filter

   A stream of bits that changes too rapidly between the zero and one states can cause the bandwidth of the transmitted signal to increase beyond that allocated to the upstream channel. (It is a fundamental law of physics that the more abruptly a signal changes state, the wider the bandwidth that it occupies. To a good approximation, if a signal changes state in *n* microseconds, it will occupy a bandwidth of 1/*n* MHz.) Before transmission, the cable modem smooths the modulating signal so that the transmitted signal will fit within a channel width no greater than 1.25 times the symbol rate.

8. Modulate

   The carrier is modulated with the filtered signal and placed on the coax.

The process is arranged so that the final step, modulation, begins precisely at the start of a minislot.

Downstream Data Flow Through a Cable Modem

Downstream flow is essentially the inverse of the upstream process: The incoming data are demodulated; the bitstream is recovered by applying an inverse symbol map; the preamble is removed; the bits are unscrambled; the FEC encoding is removed (or, if necessary, the information in the FEC parity bits is used to correct for erroneous bits in the data); and finally, the actual data are extracted.

Because the downstream data travel in a less noisy part of the spectrum, they are not subject to the same bandwidth and noise constraints as the upstream data and can be transmitted using more efficient modulation schemes. On the downstream link, a DOCSIS cable modem is required to support 64-QAM with a symbol rate of 5.056941 megasymbols per second (corresponding to 30.341646 megabits per second) and 256-QAM with a rate of 5.360537 megasymbols per second (corresponding to

42.884296 megabits per second). These two rates are often referred to as "30 megabits" and "40 megabits", respectively. Note, however, that these numbers refer to total raw downlink capacity and make no allowance for the overhead that is added to each data packet, nor to the fact that many modems are probably sharing the same downstream channel.

The packet format for downstream data is formatted quite differently from that used for upstream traffic. Instead of using Ethernet-based formatting, the downstream packets are formatted as a continuous stream of 188-octet MPEG[8] packets, each packet comprising a four-octet header followed by 184 octets of data. The MPEG format was chosen for the downstream data because this particular format is well adapted to carrying real-time video data, and the format is already used to deliver certain kinds of digital television down the cable. The details of the MPEG formatting used to carry data packets are provided in the DOCSIS RF specification.

Media Access Control Layer

Above the PMD layer lies the **Media Access Control** (**MAC**) protocol layer. This layer supports the following key properties and features.

- Bandwidth allocation
- Providing upstream minislots
- Contention-based and reservation-based upstream transmission
- Variable-length packets
- Quality of Service, providing bandwidth and latency guarantees and creation, management and deletion of dynamic flows
- Range of data rates

The MAC layer network topology is not confined to a single CM/CMTS pair. Rather it comprises a CMTS and a suite of managed CMs. As we have discussed, the CM-CMTS relationship is not peer-to-peer. CMs act as clients of a CMTS, which instructs them exactly how they are to behave in order that fairness is ensured for all the CMs for which the CMTS is responsible.

---

8. **MPEG** stands for **Moving Picture Experts Group**. The MPEG format, which was originally defined by this group, has been adopted by ITU as recommendation H.222.0.

All transmitted data obey the following rules of ordering.

1. Within an octet, the least significant bit is transmitted first.
2. When the value being transmitted spans more than one octet, the octets are transmitted in order of most significant to least significant. This way of ordering values that span multiple octets is known as **network order**. In the absence of an explicit directive to the contrary, all such values used in all the protocols in this book are transmitted in network order.
3. Signed integer values are transmitted in two's complement format.

   In two's complement format, negative values are transmitted as follows: In order to transmit the negative number –N (where N is positive), calculate +N-1, then invert all the bits in the binary representation of this number. This is the value to be transmitted.

   For example, to transmit a single octet representing the value –15, the octet that is transmitted has the form 11110001 (but remember that the bits are actually transmitted in reverse order, per the first rule).

Figure 3-13 shows the relationship between the MAC layer and the PMD layer.

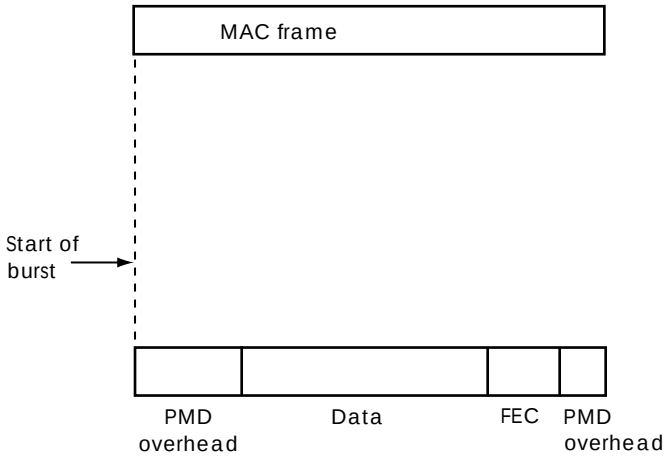


Figure 3-13 Relationship Between MAC Layer and PMD Layer

MAC Header Format

DOCSIS MAC headers have the format shown in Figure 3-14. The Extended Header field, EHDR, is optional. (The DOCSIS security mechanism, Baseline Privacy Interface Plus, requires this field. For details, see the section "BPI + MAC Extended Header".)

The fields in the MAC header are as follows.

- FC

  Frame Control; an octet that identifies the type of the header. The octet is broken down as follows.

  - FC TYPE

    2 bits used to define the type of the rest of the packet

  - FC PARM

    5 bits set to 0; other values are reserved for future use and are currently ignored

  - EHDR_ON

    1 bit, set to 1 if an Extended Header field is present; otherwise set to 0

- MAC_PARM

  Parameters; an octet whose meaning depends on FC

  If EHDR_ON is 1, then MAC_PARM contains the length of the EHDR field.

  Else, if this is part of a concatenated frame, contains the MAC frame count.

  Else indicates the number of minislots requested.

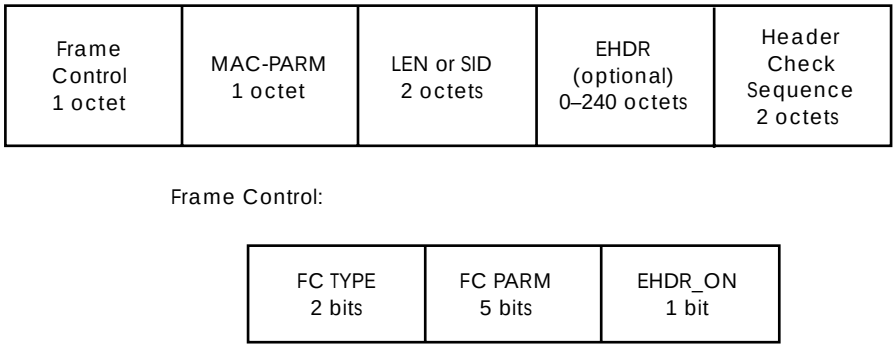


Figure 3-14 DOCSIS MAC Header Format

- LEN (SID[9])

  Length of the MAC frame, or the SID; 2 octets

  If this is a Request header, contains the SID for which the request is being made in the bottom 14 bits.

  Else, contains the length of the MAC frame, defined as the sum of the number of octets in the extended header (which may be zero) and the number of octets following the HCS field.

- EHDR

  Extended header; optional, length 0 to 240 octets

- HCS

  Header Check Sequence; 2 octets. Used to ensure the integrity of the MAC header, this is a 16-bit Cyclic Redundancy Check (CRC) calculated over the rest of the MAC header (including the EHDR, if present). The method of calculating the HCS is CRC-CCITT($x^{16} + x^{12} + x^5 + 1$), described in ISO recommendation 8802-3. See also Appendix B, where the CRC calculation is presented in some detail.

MAC Packet Protocol Data Unit (PDU) Format

DOCSIS modems support variable-length packets that may contain up to 1,500 user octets per packet. The format of DOCSIS MAC data packets is the same as is used in Ethernet, described in ISO 8802-3.

Each data packet begins with a six-octet header as defined in the section "MAC Header Format", with FC TYPE and EHDR_ON both set to 0.

The **Packet Protocol Data Unit** (**PDU**) immediately follows the header and has the format shown in Figure 3-15.

The octets of user data are embedded in the PDU, following 14 octets of routing information and preceding 4 octets of CRC checking, as follows.

- DA

  Destination Address. A 48-bit (6-octet) destination address that identifies the intended recipient.

- SA

  Source Address. A 48-bit (6-octet) source address that identifies the originator of the packet.

9. SID is a DOCSIS term for a Service ID, a concept discussed more fully in the section "MAC Management".

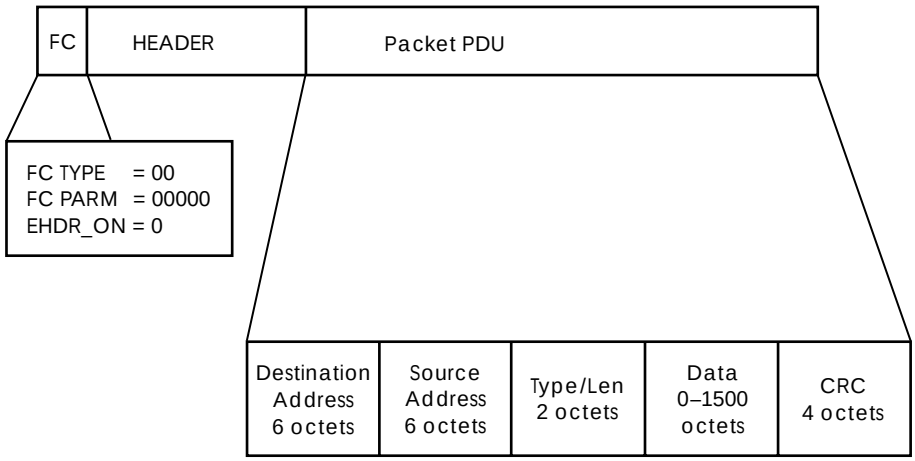


Figure 3-15 DOCSIS MAC Packet PDU Format

- Type/Len

  A 16-bit (2-octet) field that defines either the Ethernet Type or the Length of the data, in conformance with ISO 8802-3.

- CRC

  A 32-bit (4-octet) CRC calculated over the rest of the packet PDU, as specified in ISO 8802-3 and in Appendix B.

Specialized MAC Headers

There are also many specialized MAC headers that are used for specific management functions that maintain synchronization between the CM and the CMTS. These are the headers.

- Timing Header
- MAC Management Header
- Request Frame
- Fragmentation Header
- Concatenation Header
- Fragmentation Extended Header
- Service Flow Extended Header

- Payload Header Suppression Header
- Unsolicited Grant Synchronization Header

For the most part, the details of these headers are beyond the scope of this book. However, the MAC Management header and the Request frame are of interest, since they are used to request and deliver guaranteed Quality of Service for telephony on the upstream link.

The MAC Management header supports MAC management messages (unsurprisingly). It has the following fields.

- FC
  - FC TYPE

    11
  - FC PARM

    00001
  - EHDR_ON

    0
- MAC_PARM

  Reserved for future use
- LEN

  Length of the packet PDU, in octets
- HCS

  Header Check Sequence.

The header is followed by a specific MAC management message.

The request frame is used by the cable modem to request upstream bandwidth for sending information packets. The header is transmitted "bare", without any subsequent PDU. Since it is only 6 octets long, it fits in a single minislot. Here is its format.

- FC
  - FC TYPE

    11

- FC PARM

  0001x

  x = 0 for a minislot request

  (x = 1 for an ATM cell request)

- EHDR_ON

  0

- MAC_PARM

  Total number of minislots requested; this is the actual number of minislots needed to transmit the desired data, including any PMD overhead.

- SID

  The Service ID for the flow requesting bandwidth

- Header Check Sequence

Format of MAC Management Messages

There are 255 possible different types of MAC Management message; of these, 22 are currently defined, as shown in Table 3-4.

The format of MAC Management messages is as shown in Figure 3-16. MAC management messages are encapsulated within an LLC unnumbered information frame, described in ISO 8802-2. The MAC Management Message is preceded by an ordinary MAC header that indicates that the PDU contains a MAC Management Message, formatted according to the section "Specialized MAC Headers".

The fields in the MAC Management Message Header (that is, the header contained in the PDU, not the one that preceeds the Management Message) are as follows.

- DA

  Destination Address (48 bits)

- SA

  Source Address (48 bits)

- msgLen

  The length, in octets, of the MAC message, starting at DSAP and ending at the end of the payload. (This does not include the CRC check.) 2 octets.

- DSAP (Destination Service Access Point)

  Defined to be 0; 1 octet

Table 3-4 MAC Management Messages

Message Type Value	*Message Name*	*Message Description*
1	SYNC	Timing Synchronization
2	UCD	Upstream Channel Descriptor
3	MAP	Upstream Bandwidth Allocation
4	RNG-REQ	Ranging Request
5	RNG-RSP	Ranging Response
6	REG-REQ	Registration Request
7	REG-RSP	Registration Response
8	UCC-REQ	Upstream Channel Change Request
9	UCC-RSP	Upstream Channel Change Response
10	TRI-TCD	Telephony Channel Descriptor
11	TRI-TSI	Termination System Information
12	BPKM-REQ	Privacy Key Management Request
13	BPKM-RSP	Privacy Key Management Response
14	REG-ACK	Registration Acknowledgement
15	DSA-REQ	Dynamic Service Addition Request
16	DSA-RSP	Dynamic Service Addition Response
17	DSA-ACK	Dynamic Service Addition Acknowledgement
18	DSC-REQ	Dynamic Service Change Request
19	DSC-RSP	Dynamic Service Change Response
20	DSC-ACK	Dynamic Service Change Acknowledgement
21	DSD-REQ	Dynamic Service Deletion Request
22	DSD-RSP	Dynamic Service Deletion Response

- SSAP (Source Service Access Point)

  Defined to be 0; 1 octet

- Control

  Defined to be 3 (which marks this as an Ethernet unnumbered information frame); 1 octet

- Version

  Defined to be either 1 or 2, depending on the Type field; 1 octet

- Type

  The type of the message, taken from Table 3-4; version = 1 for messages 1 to 13; version = 2 for messages 14 to 22; 1 octet

- RSVD (reserved, used only for alignment)

  Defined to be 0; 1 octet

The header is followed by the management message payload. Following the payload is a 4-octet CRC check calculated as specified by ISO 8802-3 and Appendix B, which covers the message, beginning with DA and ending at the end of the payload.

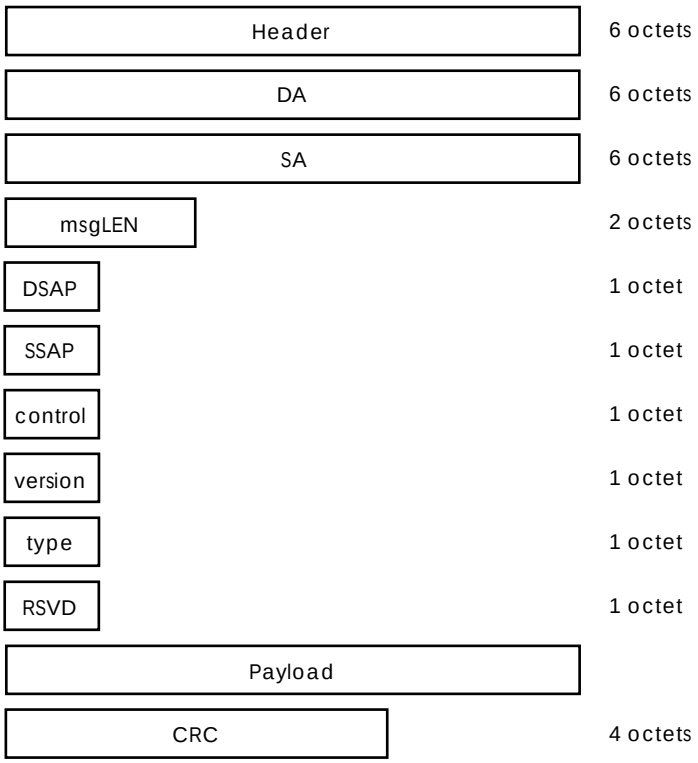


Figure 3-16 Format of MAC Management Messages

MAC Management

We have referred several times to the fact that, unlike on a peer-to-peer analog modem system, a cable modem acts as a client of its termination system. This is necessary for efficient operation in a shared environment where multiple modems are contending for limited bandwidth resources. In order for a CMTS to manage its client modems effectively, DOCSIS defines a number of MAC management messages, which are used to ensure that all the modems served by a single CMTS act in a fair and reasonable manner.

Not all of the MAC management messages listed in Table 3-4 are of equal importance to telephony applications. We will describe only the most important ones, which are useful to understanding how a cable modem establishes its place in the access network and recognizes when it is permitted to transmit upstream data.

Service Identifiers and Service Flow Identifiers

The CMTS manages CM data flows by assigning to each modem two or more 32-bit **Service Flow Identifiers** (**Service Flow ID**s, or **SFID**s), each of which is unique within the set of cable modems for which the CMTS is responsible. At least one SFID represents a data flow in the upstream direction, and at least one represents flow downstream.

In addition to the SFIDs, **upstream** flows are assigned 14-bit **Service ID**s (**SID**s), which the modem uses to request, and the CMTS to grant, upstream bandwidth. SFIDs are explored in greater detail in the section "Quality of Service (QoS)".

Within a cable modem, SFIDs are treated independently and may have different priorities. Call signaling traffic, for example, might be deemed more important than bearer data and so be assigned to a SFID with a higher priority. Similarly, voice conversation may be allocated to a SFID with a guaranteed upstream bandwidth, whereas an e-mail application would more likely be transmitted via an SFID associated with a low-priority, "best-effort" stream.[10]

Time Synchronization Message (SYNC)

It is vital that all the modems in a system (those sharing a common CMTS) have a closely aligned notion of the time. This is accomplished through the periodic transmission of SYNC messages by the CMTS. Nominally, SYNC messages are transmitted every 200 milliseconds.

The CMTS contains a master clock, running at precisely 10.24 MHz. Every time this clock ticks (every 0.09765625 microseconds) a 32-bit counter inside the CMTS

10. A "best effort" channel is one that does not provide a guaranteed amount of bandwidth.

increments. In order to bypass the (perhaps variable) delay in the CMTS protocol stack, the current value of the clock is inserted into the SYNC message at the moment that the message is handed to the PMD Sublayer, so that the most accurate value possible is transmitted.

Note that the resolution of the master clock is such that 64 counts correspond to 6.25 microseconds. Since the minimum duration of a minislot is 12.5 microseconds, the resolution of the time in the SYNC message is sufficient to enable the CM/CMTS combination to synchronize clocks within a tiny fraction of a minislot.

Because electrical signals travel at a finite speed, there is a correspondence between time and distance. Since signals travel through the access network rather more slowly than they do in air, a single tick of the 10.24 MHz clock, or 0.09765625 microseconds, corresponds to a distance of approximately 80 feet. This is the ultimate spatial resolution of the access network.

Upstream Channel Descriptor (UCD)

Upstream Channel Descriptors are transmitted periodically by the CMTS (nominally once every two seconds). Their purpose is to define the characteristics of each upstream channel. Because of the many parameters that may be required to completely define the characteristics of a particular upstream channel, UCDs require on an extensible mechanism for encoding parameters. The mechanism used is known as **TLV** encoding (from "Type-Length-Value") and is used frequently throughout DOCSIS and PacketCable when a protocol calls for passing variable amounts of information.

In TLV encoding, the value of a parameter is encoded using the following.

1. A Type field, of length one octet, which represents the parameter being encoded
2. A Length field, of length one octet, which gives the length, in octets, of the value of the parameter
3. The Value of the parameter

The length of a TLV-encoded parameter is therefore equal to the length of the value of the parameter, plus two. The entire UCD is shown in Figure 3-17.

The non-TLV-encoded fields are interpreted as follows.

- Upstream Channel ID

  Value used to identify the particular upstream channel to which this message refers

- Configuration Change Count

  This value is incremented by one whenever there is a change in any parameter encoded in the UCD. This allows a modem to disregard the rest of the message if

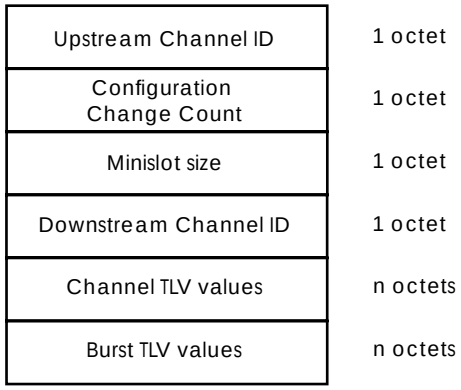


Figure 3-17 Upstream Channel Descriptor

the Configuration Change Count is the same as for a message that has already been processed.

- Minislot size

  The duration of the minislot for this upstream channel, given in units of 6.25 microseconds. Allowable values are: 2, 4, 8, 16, 32, 64, 128.

- Downstream Channel ID

  Value used to identify the downstream channel on which this message is being transmitted.

  DOCSIS defines four different allowable TLV parameters for the channel.

- Symbol Rate

  Type = 1; Length = 1. Gives the symbol rate, in units of the base rate of 160 kilo-symbols per second. Allowable values are: 1, 2, 4, 8, 16.

- Frequency

  Type = 2; Length = 4. Gives the carrier frequency of the upstream channel, in Hertz.

- Preamble Pattern

  Type = 3; Length = 1 to 128. Preamble superstring. See below.

- Burst Descriptor

  Type = 4; Length = total length of descriptor.

The last two of these require further explanation.

Preambles are required in DOCSIS in order to help the receiver synchronize properly (see Upstream Transmission). A Type 3 TLV parameter provides a "super" preamble bit stream, from which actual preambles are chosen (in the burst descriptor).

Burst Descriptors are unordered compound encodings of further TLV-encoded quantities that may be used to define a number of physical-layer characteristics. For further details of these, consult the DOCSIS specifications.

Ranging

The ranging process was briefly described in Initialization. Cable modems need to perform ranging during initialization and periodically during operation, in order to ensure that power levels, carrier frequencies and clocks do not drift out of alignment with the other CMs on the network. The mechanism for this is the Ranging Request (RNG-REQ) and Ranging Response (RNG-RSP) pair of messages.

The most difficult adjustment to understand concerns clock synchronization. Because of the finite speed at which information flows in the cable network, it is not trivial to maintain highly accurate clock synchronization.[11] Depending on the ratio of fiber to coax in the access network, as well as the number and quality of the amplifiers, the speed at which information flows between a modem and its CMTS is typically between 67% and 80% of the speed of light in vacuo (which is approximately 186,000 miles per second or roughly 1 foot per nanosecond).

Since an access network may cover distances of several tens of miles, and cable modems must cooperate in transmitting data within an accuracy of a few miscroseconds, each modem must establish quite accurately its location within the network. This makes it possible to time its transmissions so that they do not collide with transmissions from other modems and so that they arrive at the CMTS at the correct time.

The CMTS defines both the master time and the master location within the network. All timing is performed on the basis of the 10.24 MHz clock located within the CMTS. If the CMTS instructs a modem to transmit at a particular time *t*, *t* references the arrival time of the information at the CMTS. In other words, all time corrections must be performed by the individual modems on the network. The CMTS itself makes no allowances for transmission delays.

Ranging Request (RNG-REQ)

Ranging Requests are transmitted by cable modems at initialization and thereafter when requested to do so by the CMTS. Ranging requests are used to determine the

11. Deviating slightly from the subject at hand, the difficulty of synchronizing clocks at a distance was one of the problems that led to Einstein's Special Theory of Relativity.

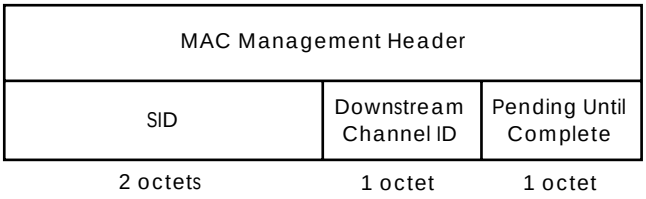


Figure 3-18 Format of a CM Ranging Request (RNG-REQ) Message

packet-delay time on the network between the CM and the CMTS, as well as to ensure that the carrier frequency and power levels are within reasonable limits. The format of an RNG-REQ is given in Figure 3-18.

- SID

  An Initialization SID, a Temporary SID, or a Registration SID if this is an "Initial Maintenance" Request. If this is a "Station Maintenance" request, then an Assigned SID.

  The SID is transmitted in the lower 14 bits.

- Downstream Channel ID

  Identifier of the downstream channel on which the CM received the UCD that provided the parameters for this upstream channel. This allows the CMTS to know on which downstream channel the CM is listening for a response.

- Pending Till Complete

  Indicates whether all received Ranging Responses have been processed. If zero, processing is complete; if non-zero then this field contains an estimate of the amount of time needed (in hundredths of a second) by the CM to complete processing of a received Ranging Response.

Ranging Response (RNG-RSP)

Unsurprisingly, Ranging Responses are transmitted to cable modems in response to Ranging Request messages. However, they may also be transmitted at other times, and a cable modem must be prepared to receive and process a Ranging Response message at any time. (For example, a CMTS may notice that the transmissions from a particular CM are in danger of exceeeding tolerable limits in regards to power, frequency or time, and may choose to unilaterally transmit an RNG-RSP rather than requesting a CM to perform an explicit ranging operation.) The format of an RNG-RSP message is given in Figure 3-19.

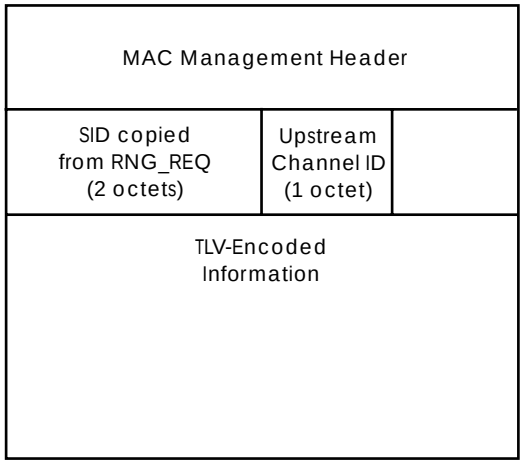


Figure 3-19 Format of a CMTS Ranging Response (RNG-RSP) Message

- SID

  Either an initialization SID, if this response is instructing the CM to change channels, or the SID contained in the corresponding RNG-REQ, or the temporary SID if the RNG-REQ was part of the initialization sequence.

- Upstream Channel ID

  Identifier of the channel on which the CMTS received the RNG-REQ.

- Ranging Status

  Indicates whether the received messages from the CM are within tolerable limits. (If they are not, the CM should perform ranging until they do fall within the correct limits, or, if that does not happen after several attempts, the modem should shut itself down and indicate a fault.)

Other values are TLV-encoded, according to Table 3-5. *Note*: The decibel is a unit that measures the ratio of two power levels, P_1 and P_2. P_1 and P_2 are separated by N db where $N = 10 \log_{10}(P_2/P_1)$. If P_2 is twice P_1, then the signals are separated by almost exactly 3 dB. If P_2 is ten times P_1, then the signals are separated by precisely 10 dB.

Upstream Bandwidth Allocation Map (MAP)

Upstream data may be transmitted in either contention minislots or noncontention minislots. Generally, user data—especially telephony data—are sent in noncontention

Table 3-5 TLV-Encoded Ranging Response Parameters

Adjustment	*Type (1 octet)*	*Length (1 octet)*	*Meaning*
Timing	1	4	Signed transmission timing offset; units of 6.25/64 microseconds (that is, units of the number of ticks of the 10.24 MHz clock)
Power	2	1	Signed power offset; units of 1/4 dB
Frequency	3	2	Signed frequency offset, units of Hz
Equalization	4	n	Equalization data; see DOCSIS specifications for details.
Ranging Status	5	1	1 = continue; 2 = abort; 3 = success
Downstream Frequency	6	4	Center frequency of new downstream channel (Hz)
Upstream Channel ID	7	1	ID of new upstream channel

minislots. The CMTS periodically broadcasts Upstream Bandwidth Allocation Map (**MAP**) messages, which describe the detailed allocation of upcoming timeslots for each upstream channel. The format of a MAP message is shown in Figure 3-20 and described below.

The measurement of time as used in MAP messages can be somewhat confusing. MAP messsages measure time in units of minislots, using a 32-bit counter that wraps silently at 2^{32}–1. This is akin to the counter in the SYNC messages broadcast by the CMTS, but the unit of time is different in the two cases.

In the SYNC message, the unit is the number of ticks of the 10.24 MHz clock, which ticks many times per minislot. The number of ticks per minislot is given in the UCD message and is always a power of two. Time as measured in the MAP message must agree with time as measured in the SYNC messages, except that the latter is more accurate. The least-significant bits in the MAP message must match the corresponding (more significant) bits in the SYNC message.

For example, if the time as measured in the SYNC message has the value 1234567890, which corresponds to the bit pattern 01001001 10010110 00000010 11010010, and if there are 128 ticks per minislot, then the corresponding time in the MAP message must have the lowest 25 bits equal to 0 10010011 00101100 00000101. See Figure 3-20.

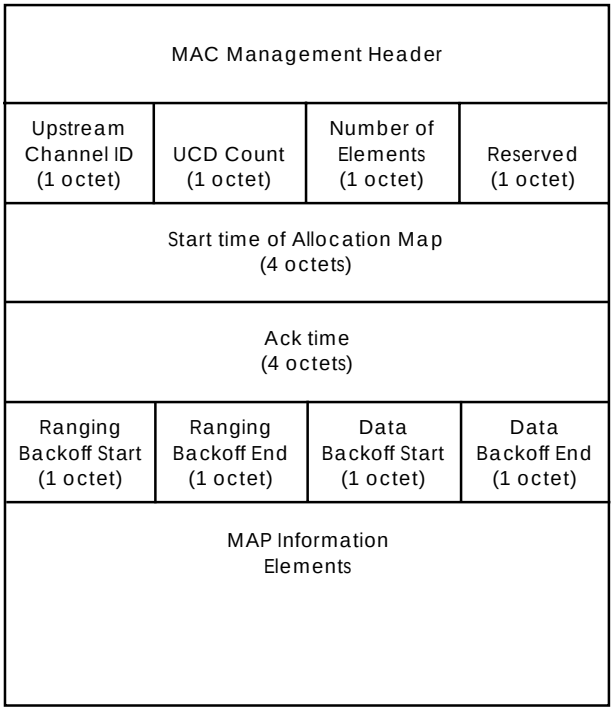


Figure 3-20 DOCSIS MAP Message Format

- Upstream Channel ID

  Value used to identify the particular upstream channel to which this message refers

- UCD Count

  Matches the value of the Configuration Change Count field of the UCD, which contains the Burst Descriptor that applies to this map. This ensures that any modem transmitting data according to this map will do so with the correct physical parameters.

- Number Elements

  Number of information elements that appear in the map

- Reserved

  Used for alignment

- Alloc Start Time

  Effective start time, in minislots, of assignments within this map

- Ack Time

  Latest time, in minislots, that was processed in the upstream direction before this map was generated. That is, requests or other upstream signals from cable modems timestamped with values subsequent to this time were unprocessed by the CMTS when this map was generated.

- Ranging Backoff Start

  Initial backoff window, used for ranging contention, expressed as a power of two. Possible values range from zero to 15.

- Ranging Backoff End

  Final backoff window, used for ranging contention, expressed as a power of two. Possible values range from zero to 15.

- Data Backoff Start

  Initial backoff window, used for contention data and requests, expressed as a power of two. Possible values range from zero to 15.

- Data Backoff End

  Final backoff window, used for contention data and requests, expressed as a power of two. Possible values range from zero to 15.

- MAP Information Elements

MAP Information Elements

When providing an upstream bandwidth allocation map, the CMTS uses Information Elements (IEs) to encode the details of the allocation. For each granted interval described in a MAP, the CMTS transmits a 32-bit quantity, divided into three fields that encode the upstream SID, the precise time of the minislots granted, and the use to which these minislots are to be put. For more details, see the next section.

Example Upstream Bandwidth Allocation

As we have seen, the process by which a cable modem obtains permission to transmit data upstream is far from trivial. In this section we will attempt to walk through the process slowly to be sure that we understand how the mechanism works.

The essential problem is that a number of modems share a single upstream channel (there are a number of upstream channels, and each channel can be treated independently, but even so, each upstream channel is shared by several modems). Therefore there has to be an arbitrated mechanism by which each modem can be

assured of opportunities to transmit. The fact that all the modems share a notion of time, with themselves and their controlling CMTS, makes cooperation possible.

The upstream channel is treated as a sequence of contiguous minislots. The CMTS transmits (on the downstream channel) a MAC management message, the MAP message, that describes exactly how an upcoming series of minislots is to be used. Note that a malfunctioning cable modem that does not honor these commands will be quickly discovered and will be effectively shut down by the CMTS. (Unless, of course, the malfunction is so great that it fails to obey these commands as well. In such a circumstance, the CMTS should recognize what has happened and move traffic on to other channels; it should also notify the network operator of the problem so that, if nothing else can be done, the malfunctioning CM can be shut down manually.) However, it is possible for an individual to create a CM-like device that, when connected to a cable, either renders the cable useless or at least severely degrades performance (for example, by rapidly sweeping a powerful carrier in the frequency domain). This is a consequence of a shared pipe, and there is little that can be done to prevent such attacks.[12]

A typical MAP might grant some minislots for the exclusive use of particular modems that have indicated in prior Request frames that they have data ready to transmit requiring a number of minislots to transmit. It might also set aside some minislots for modems to use in contention mode and yet others that may be used only by new modems signaling that they wish to join the network. The scheduling algorithm is controlled entirely by the CMTS, and, in most cases, the CMTS will contain intelligence that allows the detailed scheduling to change as a function of the kind of traffic currently on the network. The exchange that occurs between modem and CMTS is shown in Figure 3-21. Figure 3-22 shows an example of how a MAP might allocate an upcoming series of minislots.

Contention Resolution

The timing of all upstream signaling is under the control of the CMTS. In particular, the CMTS decides which minislots are allocated to a particular CM and which are subject to contention.

Data transmitted in a contention minislot are not guaranteed to arrive at the CMTS, since several CMs may simultaneously transmit within the minislot, in which case all their data will be lost. Data transmitted in a noncontention minislot are almost certain to arrive at the CM (barring problems with the low-level link), since only one CM is permitted to transmit in a noncontention minislot.

12. How an MSO might go about detecting the source of such attacks in a timely manner is an interesting problem that we leave as an exercise for the reader.

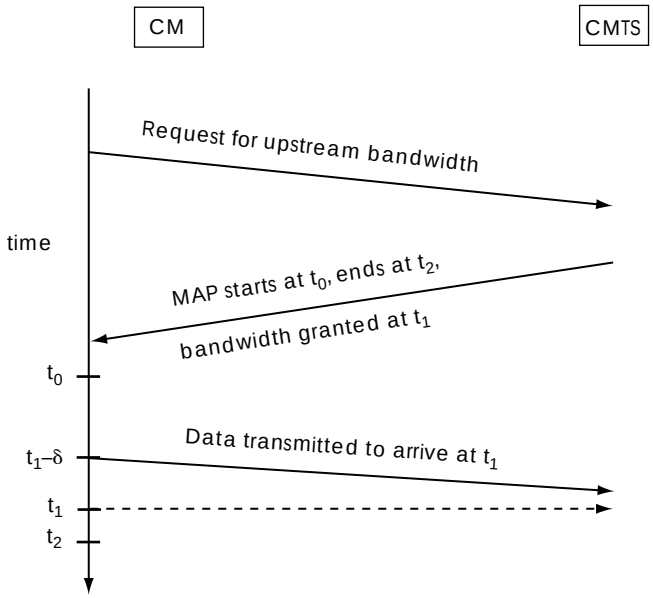


Figure 3-21 CM-CMTS Interaction Granting Upstream *Bandwidth*

Contention minislots are, in a sense, "wasted" upstream bandwidth, since they (generally) are not used to send useful data. However, they are necessary since the only way of reserving guaranteed bandwidth for user data is by requesting noncontention minislots—and the mechanism for doing this is via messaging that is carried in contention minislots.

The optimum ratio of contention minislots to noncontention minislots is a function of the type of traffic being carried. If the upstream flows on a cable are mostly carrying telephony traffic, then the number of contention minislots may be decreased

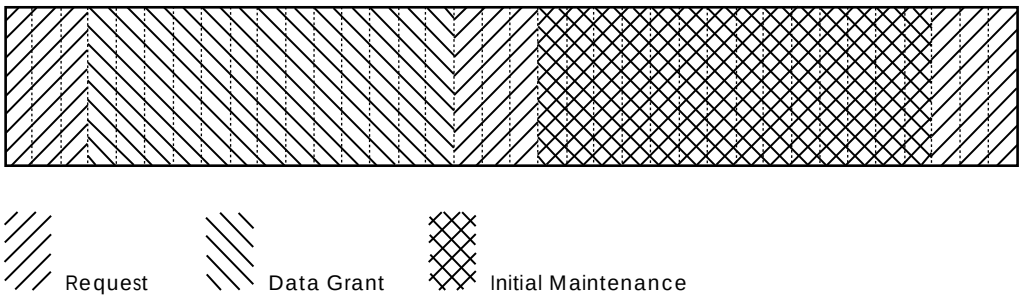


Figure 3-22 Minislot Allocation in a DOCSIS MAP Message

because the CMTS typically allocates fixed-bandwidth flows using an Unsolicited Grant mechanism, which requires very few contention minislots to operate (see Quality of Service).

On the other hand, if the flows are mostly best-effort data flows, then a relatively large number of contention minislots are needed because a relatively larger number of requests for upstream bandwidth need to be made because of the unpredictable nature of the traffic. The shorter and more bursty the data, the more contention mode minislots are needed. A well-designed CMTS will monitor the upstream traffic flow and frequently adjust the percentage of contention minislots so as to optimize the way that the upstream bandwidth is being used.

Since there is no guarantee that transmissions made in a contention minislot will be received (because of collisions), there has to be some mechanism to allow for CMs to retransmit contention data. The mechanism must also ensure that any two CMs, once they have transmitted into the same minislot and caused a collision, do not continue to do so. This is called contention resolution.

In every MAP message, the CMTS supplies a pair of values corresponding to an initial back-off window and a maximum back-off window to be used for contention resolution in the time period covered by the MAP (see Upstream Bandwidth Allocation Map). The values are presented as powers of two, such that a value of 5, for example, would indicate a back-off window of width 32.

When a CM transmits a packet in a contention minislot, it sets the width of a window to the value corresponding to the value of Data Backoff Start in the current MAP. At some time later it receives a MAP message from the CMTS (see the section "The MAP Message" for more details about this exchange). If the MAP indicates that the CMTS did not receive the modem's packet, the modem assumes that the packet was lost and begins its retransmission strategy.

The modem selects a random value in the range (0, window width – 1). It then allows this number of retransmission opportunities to pass before it retransmits its request. For example, suppose that a CM has a current value for Data Backoff Start of 5. This means that the back-off window runs from zero to 31. It transmits a request, but suppose that the CM receives no response. The CM now randomly selects a number in the range (0, 31). Suppose that it selects 13. This means that the CM must allow 13 retransmission opportunities to pass before it retransmits its request. Assume that the first Request IE[13] in the MAP is for 5 requests. It must allow these to pass, and it still has 8 more to go. The next Request IE might be for 7 requests. It must allow these to pass as well. The next Request IE might be for 2 requests. It must remain silent for the first but will retransmit on the second.

13. A "Request IE" is a list of opportunities provided by the CMTS in which requests for upstream bandwidth may be made. For details of this and other Information Elements (IEs) see the next section.

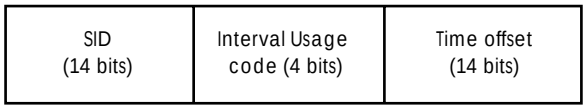
If this transmission also fails to elicit a response, the CM doubles the length of the back-off window (to a maximum value controlled by the maximum back-off window allowed in the currently applicable MAP), generates a new random number within this window, lets that number of opportunities pass, and then retransmits.

The MAP Message

A MAP message contains an ordinary MAC management header, followed by a variable number of **Information Elements** (**IEs**) in the format given in Figure 3-23. IEs within a MAP message are ordered strictly according to time, as decribed by the time counter in the CMTS. Except for the first IE, the start time of an IE is (usually) inferred from the start time and duration of the prior IE. A null IE terminates the list of IEs.

Each IE contains a 14-bit SID, a 4-bit type code and a 14-bit time offset. A SID of 0x3FFF indicates a broadcast intended for all CMs. Ordinary unicast SIDs are in the range 0x0001 to 0x1FFF and are used to describe a particular CM or a particular service within a particular CM. SIDs in the range 0x2000 to 0x3FF0 are used for multicast messages, which are used only for administrative purposes. SIDs in the range 0x3FF1 to 0x3FFE are used to describe contention minislots of various lengths, as shown in Table 3-6. The transmissions sent in a single contention-mode burst may not exceed 14 minislots in length. Therefore if a modem desires to transmit information that exceeds 14 minislots, it must do so in noncontention mode.

Note that the there is no practical difference between a “broadcast” message and a “multicast” message in this context. We preserve the difference in terminology merely because the DOCSIS specification does so. The basic idea is simple: The



Interval Usage Code	Information Element
1	Request
2	Request/Data
3	Initial Maintenance
4	Station Maintenance
5	Short Data Grant
6	Long Data Grant
7	Null
8	Data Acknowledgement
9–14	Reserved
15	Expansion

Figure 3-23 MAP Information Elements

CMTS makes the list of available minislots known to all the cable modems. The following IEs are defined:

Request IE Corresponds to intervals in which CMs may make requests for upstream bandwidth. The Request IE is usually broadcast, indicating that the marked minislots are considered to be contention minislots. If it is unicast to a particular modem, then only that modem may use the marked minislots to request upstream bandwidth.

Request/Data IE Marks minislots that may be used either for requests for upstream bandwidth or to transmit short data packets that fit entirely within the allocated minislots. The value of the SID, which is typically a multicast SID, indicates exactly how the data may be sent, according to Table 3-6.

Since these minislots are contention minislots available for any CM to use, if a CM uses them to transmit data (as opposed to Requests), the transmitted data packets should request a data acknowledgement; otherwise the CM has no way to determine whether the information reached the CMTS.

Initial Maintenance IE Provides an opportunity for new devices to join the network. Initial Maintenance grants are for relatively large numbers of minislots, as

Table 3-6 Mapping of Multicast SIDs to Data Transmission Algorithms

SID	*Meaning*
0x0000	Broadcast
0x3FFF	Broadcast
0x3FF1	Multicast; a CM may transmit at any minislot, but the transmission must fit entirely within a single minislot.
0x3FF2	Multicast; a CM may start to transmit at minislot number 1, 3, 5 and so on, and the transmission must fit entirely within two minislots.
0x3FF3	Multicast; a CM may start to transmit at minislot number 1, 4, 7 and so on, and the transmission must fit entirely within three minislots.
...	...
0x3FFE	Multicast; a CM may start to transmit at minislot number 1, 15, 29 and so on, and the transmission must fit entirely within 14 minislots.

they must allow for the maximum possible round-trip delay, plus the time to transmit a Ranging Request message.

Station Maintenance IE Allows CMs to perform periodic station maintenance, such as ranging or adjustments to power or frequency. Station Maintenance IEs may be either broadcast or unicast, depending on the policy of the network operator.

Null IE The Null IE is used to mark the end of the list of allocated minislots.

Data Grant IEs There are two kinds of Data Grant IEs: the Short Data Grant IE and, as you might guess, the Long Data Grant IE. Both of these IEs are used to allocate minislots for a CM to transmit data for which it has indicated (via a Request) a desire to transmit. The difference between a Short Data Grant and a Long Data Grant pertains to the physical layer: Short Data Grants are used for bursts whose length is less than the maximum burst size indicated in the Upstream Channel Descriptor (UCD); Long Data Grants are used for data that will exceed this length. Short Data Grants and Long Data Grants are effectively identical insofar as allocating upstream bandwidth is concerned.

A Data Grant IE may allocate a length of zero minislots, in which case it is a Data Grant Pending IE, and indicates to the CM that its request has been received but that no minislots have yet been allocated to it. Reception of a Data Grant Pending IE informs the CM that there is no need to repeat its request for upstream bandwidth. (Data Grant Pending IEs are placed after the Null IE that indicates the end of the allocated minislots.)

The Data Grant Pending IE is obviously useful when a large number of modems make more or less simultaneous requests for upstream bandwidth; the CMTS can acknowledge receipt of the requests without actually granting bandwidth.

Data Acknowledge IE This simply indicates that a particular data PDU was received from a CM. Typically, this is issued in response to a request for acknowledgement that was contained in an upstream packet, and upstream packets typically only request acknowledgements when they are transmitted in contention minislots. Like Data Grant Pending IEs, Data Acknowledge IEs are placed after the Null IE.

Expansion IE This is currently unused and is present merely to allow for extensibility.

A CM may make a request for upstream bandwidth during a minislot associated in the MAP with any one of the following: Request IE, Request/Data IE, Data Grant IE. The request indicates the SID for the flow that desires to transmit and the number or minislots being requested.

When a CMTS transmits a MAP containing upcoming minislot usage, it must do so sufficiently in advance of the first minislot mapped in the message to allow for the most distant CM in the network to receive and process the message before the map becomes operative. Typically, this requirement means that the CMTS must allow something of the order of a millisecond between the MAP transmission and the earliest minislot mapped in the message. Figure 3-24 shows the process of obtaining upstream bandwidth.

Suppose that a CMTS transmits, at time T_1, a MAP whose first minislot is for time T_2 and whose last minislot is at time T_3. A particular CM whose propagation delay from the CMTS is δ, receives this MAP message at some time T_1' (= T_1 + δ), prior to T_2. Suppose now that sometime shortly after T_2 the CM assembles a packet that it wishes to transmit. The CM does not immediately transmit the Request. In order to decrease the probability of collisions, it uses a strategy similar to that used for contention resolution. Using the value of the Data Backoff Start in the MAP, it generates a random number, *n*, of transmit opportunities that it must let pass (see Contention Resolution). It then scans the currently applicable MAP, looking for minislots in which it is permitted to transmit a request for sufficient upstream bandwidth to transmit the data packet. It allows *n* of them to pass and settles on the *n+1*th, for time T_5 in which to transmit its request.

The CM issues a request for the number of minislots needed to transmit the data at T_4, where $T_4 = T_5 - \delta$. However, since the Request IE was not directed at a particular CM, other modems may also transmit during the same minislot. Therefore our CM calculates a back-off timer as described in Contention Resolution, in case the CMTS does not receive the request.

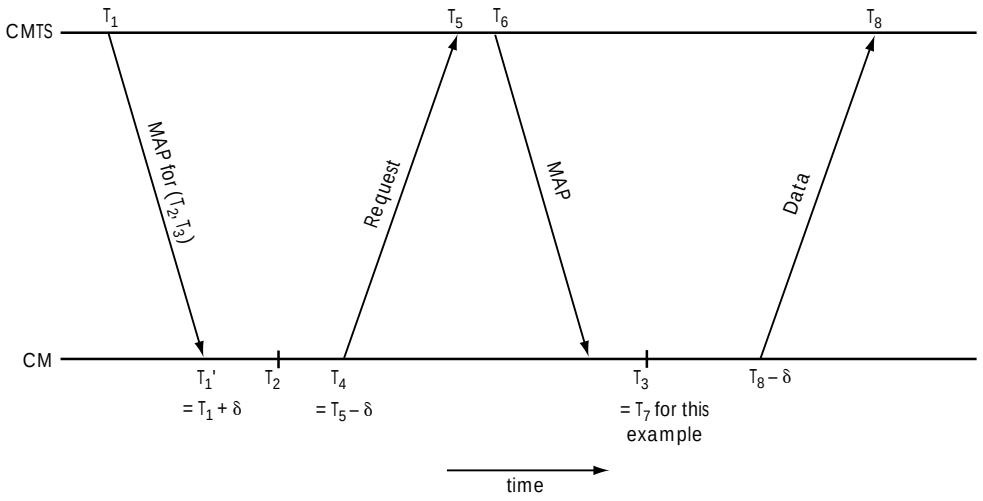


Figure 3-24 Example of a CM Obtaining Upstream Bandwidth

To keep things simple, we will assume that the CMTS receives the request at T_5 (the section "Contention Resolution" describes the back-off strategy if the CMTS does not receive it). The CMTS processes the request and (normally) will find time in the next MAP to schedule our CM's transmission. It allocates the correct number of minislots in the next MAP, whose starting time is T_7; it transmits the map at time T_6. Our CM receives the MAP at $T_6 + \delta$, scans it, and determines that it has been granted a transmit opportunity at T_8. At $T_8 - \delta$, it transmits the data, which is received at the CMTS exactly at time T_8.

Quality of Service (QoS)

In a telephony network, it is vital that users be guaranteed access to sufficient bandwidth in both upstream and downstream directions to ensure that their conversations will not be interrupted by pauses, gaps or other annoying artifacts caused by a lack of necessary bandwidth. Because of the nature of the HFC network, bandwidth management is especially important in the upstream direction, where noise is greater and available bandwidth much less than in the downstream direction.

DOCSIS contains mechanisms designed to provide CMs with guaranteed bandwidth. If the bandwidth cannot be guaranteed, a signal may be passed to higher level protocols so that they may take appropriate action (which may be, for example, to deny setup for the call). The guarantee is made *at the beginning of the call*, so that once a call has been allowed to start the users can be sure that the needed bandwidth will be available for the duration of the call.

DOCSIS modems provide QoS through the notion of Service Flows: A **Service Flow** is a unidirectional flow of packets that are guaranteed a particular bandwidth, which the flow requested at the time it was set up. Service Flows are identified by a 32-bit Service Flow Identifier (SFID) assigned by the CMTS. Each active *upstream* Service Flow also has a unique 14-bit SID.

At least two Service Flows are defined in the configuration file that the modem downloads during initialization: a Primary Upstream Service Flow and a Primary Downstream Service Flow. These flows are used for subsequent unclassified traffic and all MAC messages.

Conceptually, the resources[14] required by Service Flows belong to a three-level hierarchy. When a CM attempts to create a Service Flow, the requested resources are tested (by the CMTS) against a provisioned authorization envelope to ensure that the request can be allowed. If so, the Service Flow is *authorized*.

The CMTS then checks to ensure that sufficient resources are actually available to grant the request. If so, the Service Flow is *admitted*. Admission ensures that the

14. The resources may not be limited to bandwidth: Memory, DSP processing power or other resources may be involved. However, in practice, bandwidth is almost always the resource in shortest supply.

resources are available for use, and it reserves the necessary resources. It does not yet, however, grant the CM the right to use them. To do so, the Service Flow must be *activated*.

This notion of three levels of envelopes of resource control is an important one that we shall examine in more detail when we discuss PacketCable QoS in Chapter 6.

The particular QoS attributes of a Service Flow may be specified either by an explicit definition or by using a Service Class Name in a request. A Service Class Name is a string that the CMTS recognizes as a shorthand to refer to a particular set of QoS parameters. Service Class Names provide a useful level of indirection. For example, they allow higher-level applications to construct flows with sensible QoS values simply by using a particular name. Also, the set of QoS parameters to which a name refers may change dynamically in response to traffic patterns, allowing the CMTS to more effectively manage the various traffic flows passing through it.

Using Service Flows frees higher-level applications from the need to manage the Request frames and MAP messages we have discussed earlier. In particular, depending on the kind of flow granted, transmission opportunities may be presented to a CM without an explicit packet-by-packet request from the modem. Conceptually, a higher-level application may simply request bandwidth for a particular codec without worrying about any of the low-level details about how the packets are actually transmitted from the modem to its CMTS.

The creation of a Service Flow may be initiated either by the CM or by the CMTS. The mechanism used is a three-way handshake of MAC messages known as a Dynamic Service Addition: DSA-Request, DSA-Response and DSA-Acknowledge. Changes to an existing Service Flow are made through a similar series of Dynamic Service Change messages, and deletions of Service Flows occur through a two-way handshake of DSD-Request and DSD-Response.

The DSA and DSC messages allow for very fine-grained control of the bandwidth allocated to a Service Flow. We will talk in rather more detail about these messages in the section "Dynamic Service Flows."

When upstream bandwidth is requested, there are several mechanisms that may be used to fulfil the request. Which mechanism is chosen depends on the policy of the network operator, as well as the amount of intelligence in the CMTS and the traffic load on the upstream access network.

Unsolicited Grant Service (UGS)

An **Unsolicited Grant Service** Flow (**UGS**) is a flow to which the CMTS allocates a fixed number of minislots periodically to allow for a constant-bit-rate flow of information. If the packetized output from a voice codec consists of constant-sized packets, produced at a constant rate (as is usually the case), UGS is a very efficient method of allocating upstream bandwidth. Typically, UGS is used for telephony traffic because

it incurs very little maintenance traffic. Essentially, the CM says, "Give me *n* minislots every *m* milliseconds", and the CMTS then grants these minislots (they appear in the MAP messages) without the need for the CM to explicitly request them.

Real-Time Polling Service

Real-Time Polling is similar to UGS, except that the CMTS periodically gives the modem an opportunity to request upstream minislots to transmit queued data. If the CM has no data to transmit, it issues no request and therefore the CMTS is free to reallocate those minislots to another modem.

UGS with Activity Detection (AD)

Combining elements of UGS and real-time polling, a Service Flow operating under UGS/AD is monitored by the CMTS. When the CMTS detects a number of unused minislots, it reverts to real-time polling until such time as the CM begins transmitting traffic on the flow; at that time the Service Flow reverts to UGS.

Non-Real-Time Polling Service

A Service Flow operating under non-real-time polling is guaranteed some transmit opportunities even when the network is congested. This service is of limited use in telephony systems. Essentially, this is a means to ensure that even in a congested network each modem is sure of at least some transmission opportunities.

Best Effort Service

In Best Effort Service Flows, the modem and CMTS simply do their best to send data when possible, with no guaranteed noncontention minislots. In a highly congested network, effective data rates may be very low in Best Effort service flows. Typically, data services use Best Effort Service Flows; unless the access network is very lightly loaded, it is not useful for transporting telephony packets.

Committed Information Rate

A Committed Information Rate Service Flow is usually configured as one that is delivered Best Effort but with some reserved non-real-time polling to ensure that at least some information will flow, even on a fully loaded network. Typically, current telephony services use the UGS mechanism to provide upstream bandwidth, although there is no specific requirement that they do so. With the deployment of more advanced non-constant-bit-rate codecs, Real-Time Polling or UGS/AD will likely become more widespread. The way in which PacketCable networks use the hooks provided by DOCSIS Dynamic Service Flows is more fully described in Chapter 6.

Dynamic Service Flows

Because of their importance to providing real-time telephony, it is worthwhile spending some time examining DOCSIS Dynamic Service Flows. Dynamic Service Flows are Service Flows that can be created, modified or deleted at will. In a typical implementation, each telephone conversation will be assigned to two Service Flows, one in the upstream direction and one downstream. We will look at how Dynamic Service Flows are created; the process for modifying (through DSC messages) and deleting (through DSD messages) Service Flows is very similar.

A Dynamic Service Flow is created by a Dynamic Service Add Request (DSA-Req); either the CM or the CMTS may initiate creation of a Dynamic Service Flow by transmitting a DSA-Req to the other device. A single DSA-Req can create at most two Service Flows, one in each direction. Whichever side initiates the request, a three way handshake takes place before the Dynamic Service Flow is in place and usable (see Figure 3-25).

A Service Flow has associated with it as many as three so-called QosParameterSets, each of which defines characteristics such as jitter, latency and details of the bandwidth allocation for the Service Flow. These are known as the ProvisionedQosParameterSet, the AdmittedQosParameterSet and the ActiveQosParameterSet.

ProvisionedQosParameterSet This is a static set of QoS parameters obtained during initialization. The parameters represent a maximal set of resources that the modem may consume for a single Service Flow. For example, if the subscriber pays only for low-bandwidth access using low-bit-rate codecs, that fact will be reflected in the values in the ProvisionedQosParameterSet.

AdmittedQosParameterSet This represents a set of QoS parameters for which the CM and/or CMTS has reserved resources. However, although the resources are guaranteed to be made available immediately on request, they cannot actually be used.

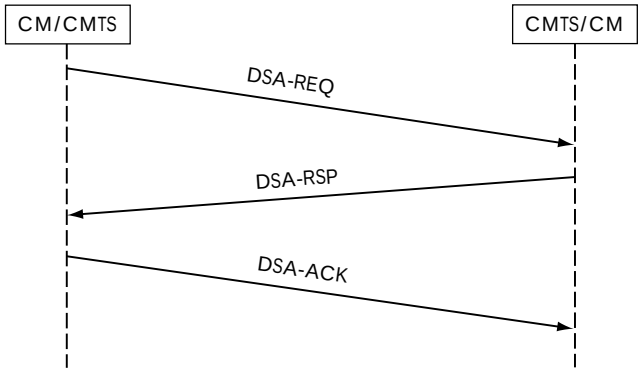


Figure 3-25 Three-Way DSA Handshake

ActiveQosParameterSet This set of parameters represents resources that are available for use. Only a Service Flow with a non-null ActiveQosParameterSet may actually carry packets.

When a DSA-Req is made, the CMTS will check that the requested resources do not exceed the ProvisionedQosParameterSet. Assuming that they lie within these boundaries, the CMTS will then check whether sufficient resources exist to admit the Service Flow and, if possible, it will do so. Typically, a DSA-Req contains a null ActiveQosParameterSet, so the Service Flow is merely admitted, not made active. A subsequent DSC command is used to convert the AdmittedQosParameterSet to an ActiveQosParameterSet—and hence to allow traffic to be carried on the Service Flow. Sometimes, however, the DSA-Req specifies that the flow is to be made immediately active (it does this by containing a non-null ActiveQosParameterSet), in which case traffic is immediately permitted to pass through the new Service Flow.

The flows for CM-initiated and CMTS-initiated Dynamic Service Additions are shown in Figures 3-26 and 3-27. As always, it is the CMTS that does the bulk of the work. The CM's role is limited to confirming that it can support the new Service Flow, configuring itself to do so and signaling that it is ready to use the new flow.

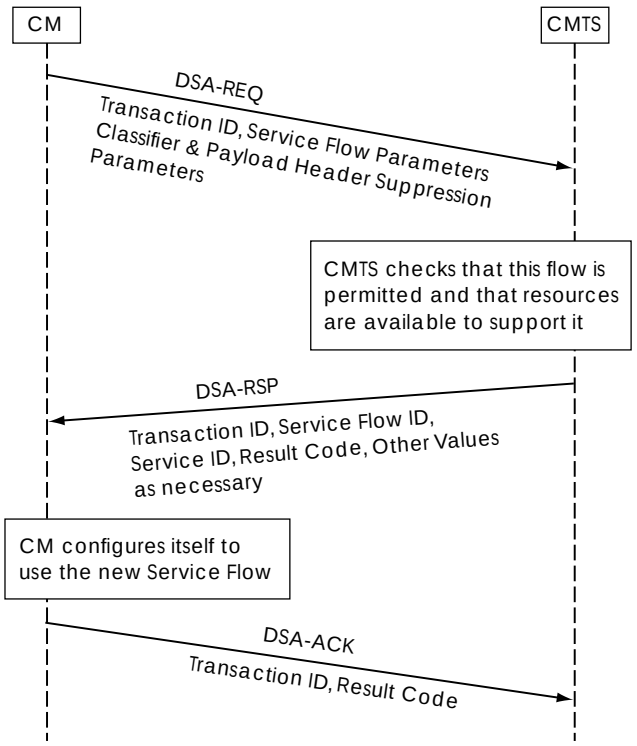


Figure 3-26 CM-Initiated Dynamic Service Addition

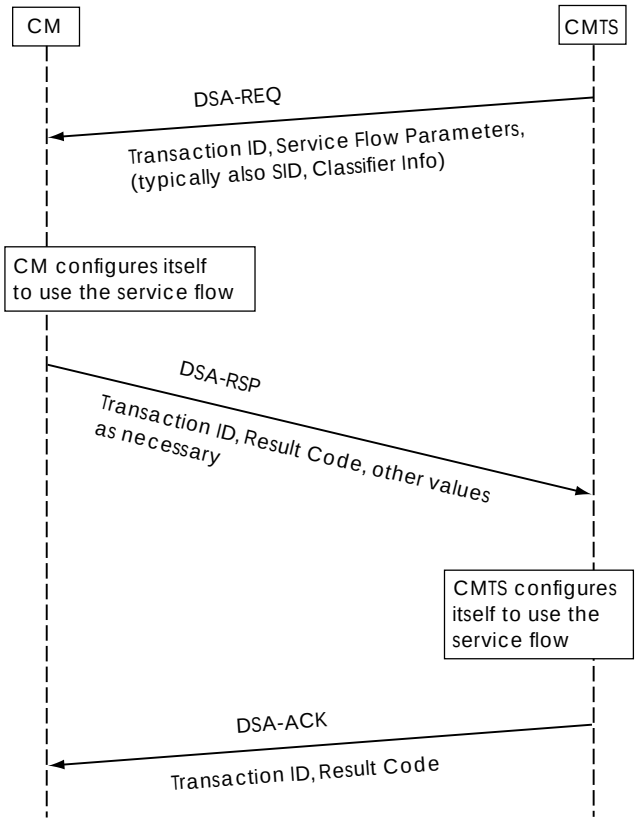


Figure 3-27 CMTS-Initiated Dynamic Service Addition

Baseline Privacy Interface Plus

On a shared-access medium such as the one provided by cable companies, it is important that each user's data—both upstream and downstream—be protected from eavesdropping and alteration by other users of the same cable. In principle, there is nothing to stop a device attached to cable from eavesdropping on, and potentially capturing, passing traffic. At least in theory, it is possible for a malfeasant to attach a device to the cable in his own home and capture the digital traffic being transmitted and received by cable modems in nearby houses.[15]

15. In practice, and as of this writing, this is quite hard because of the limited number of DOCSIS chip sets available for interfacing to cable systems, none of which were designed to support such monitoring. This situation, however, is likely to change in the near future. Earlier schemes were typically much simpler than DOCSIS and were therefore prone to being monitored.

In early, proprietary cable-based data systems there was sometimes no protection against such eavesdropping, leaving sophisticated computer users free to access information on other users' machines and also to examine the traffic passing along the coax. The current version of DOCSIS prevents this by implementing a mechanism called **Baseline Privacy Interface Plus**, **BPI+**. (An earlier incarnation was called the Baseline Privacy Interface. BPI+ is considerably more secure than BPI. It is mandatory to implement BPI+ on DOCSIS 1.1 compliant modems.)

BPI+ does not discriminate among the types of data flowing over the cable. All packets of user data transmitted over the cable are protected equally by the BPI+ security protocols. The mechanism used to secure communications between a CM and its corresponding CMTS is encryption of the traffic flows between the two devices.

BPI+ comprises two protocols.

- An encapsulation protocol, used for encrypting and decrypting the packets. This protocol defines the format for encrypted packets, the set of supported ciphersuites and the rules for applying the cryptographic algorithms to the packetized data.
- A key management protocol (Baseline Privacy Key Management, BPKM) that provides a secure method for distributing keying material between the cable modem and its CMTS.

The only encryption/decryption algorithm supported by the current version of BPI+ is the CBC mode of DES (see Chapter 2). Note that this is ordinary DES, not 3DES, and so is restricted to a 56-bit key.[16] While adequate for ordinary use, this is insufficient to confound a well-funded, determined attacker. For this reason (as well as others) PacketCable does not rely merely on BPI+ to ensure the security of telephone conversations. Instead, PacketCable places another layer of stronger encryption on top of the BPI+ encryption.

BPI+ provides only encryption; it does not use any authentication algorithms. While this may be adequate for some services, it may not be so for a telephony service. Consequently PacketCable also adds authentication (as needed) to some of the flows associated with telephony traffic.

BPI+ encrypts only MAC frame data (user data). It does not encrypt MAC frame headers. Also, it is not used to protect MAC management messages; these always travel in the clear.

16. An even weaker variant, with a 40-bit key, is also supported. The United States government for many years allowed automatic export of devices only if the device supported a maximum key length of 40 bits. Since any company that deploys a system whose security is based on 40-bit DES does not deserve to remain in business very long—and possibly because it finally dawned on the governmental authorities that the only people being penalized were those who chose to comply with the law—the government finally eased these restrictions early in 2000.

Security Associations in BPI+

BPI+ recognizes three kinds of **security associations** (**SA**s) that may exist between a CM and its CMTS.

- A Primary SA is established during MAC registration. It is an association that remains in place between the CM and the CMTS as long as the CM retains power and is unique to the CM/CMTS pair.
- Static SAs are preprovisioned within the CMTS. Multiple CMs may share the same static SA with a single CMTS.
- Dynamic SAs are created and destroyed on the fly in response to the creation and termination of specific *downstream* traffic flows. Multiple CMs may share the same dynamic SA with a single CMTS.

At any given time, apart from the single primary SA, a modem may share several static and dynamic SAs with its CMTS, each pertaining to a particular traffic flow. Each security association is identified by a 14-bit **Security Association Identifier** (**SAID**) that is unique within the universe of SAs maintained by a single CMTS. The SAID for a modem's Primary SA is numerically equal to the value of its Primary Service ID (SID).

An SA has associated with it three parameters: traffic encryption keys, CBC initialization vectors and a **ciphersuite** identifier (currently limited to whether the encryption/decryption algorithm is 40-bit or 56-bit DES).

The Primary SA is used to carry all upstream traffic. Downstream flows may use any of the three types of security association. Typically, however, telephony traffic is also carried over the Primary SA, since it would defeat the purpose of encrypting the traffic if multiple CMs had access to the keying material.

Baseline Privacy Key Management (BPKM)

In most cryptographically based security systems, key management is the most complicated part of the system. Ensuring that keys are generated randomly and shared in a secure manner is usually a complex problem, and the solutions are likewise complicated. Key management in BPI+ is no exception.

Baseline Privacy Key Management (**BPKM**) uses X.509 certificates, the RSA public key encryption algorithm, and two-key 3DES to secure the exchange of keys between a CM and its CMTS.

BPKM uses a two-tiered approach to key management, in which computationally intensive public key cryptography is used to establish a shared secret, the **Authorization Key** (**AK**), between the devices. The Authorization Key is then used to secure the exchange of the keys used for securing traffic, which are known as **Traffic**

Encryption Keys (**TEK**s). This allows the TEKs to be changed frequently without the need for expensive public key operations. Figure 3-28 shows this diagramatically.

Each modem contains an X.509 certificate, emplaced at the time of manufacture. The certificate contains the modem's public key, its 48-bit address, a manufacturer ID, and a device serial number. The certificate is signed by the modem manufacturer. This allows the CMTS, when it is presented with the certificate, to authenticate the modem. The initialization process is reasonably straightforward:

1. The CM sends the certificate to the CMTS, and the CMTS authenticates the CM.
2. The CMTS generates an Authorization Key and returns it to the CM, encrypted by the CM's public key (which was obtained from the X.509 certificate).
3. In addition to the Authorization Key, the CMTS identifies the SAID and the corresponding properties of the primary SA shared by the CM and the CMTS.
4. The CM and CMTS jointly derive a Key Encryption Key (KEK) and authentication keys from the Authorization Key.
5. The CM requests Traffic Encryption Keys for its traffic flows; the CMTS responds, encrypting the TEKs with the KEK.

The rest of this section discusses this exchange in more detail.

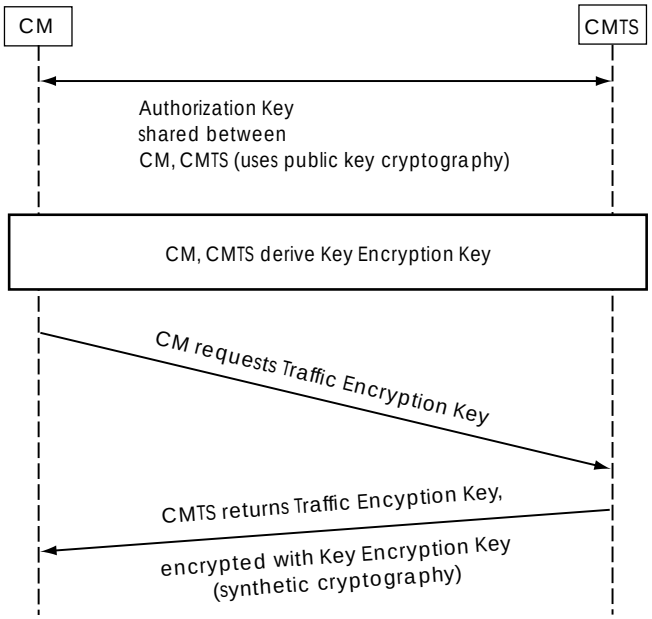


Figure 3-28 Basic BPKM Mechanism

Authenticating the CM

The BPI+ initialization sequence begins when a CM sends an Authentication Information message to the CMTS. The detailed format of this message is contained in the DOCSIS specifications. For our purposes, the important thing to know is that the message contains the modem's X.509 certificate.

The CMTS does not respond to the Authentication Information message, which is purely informative. It does, however, allow the CMTS to authenticate the modem in advance of an explicit request for security information.

Immediately after sending the Authentication Information message, the CM transmits an Authorization Request. This, as its name suggests, is an explicit request to generate security parameters for communication between the modem and the CMTS. The Authorization Request message contains the following.

- A CM-Identification attribute, which itself contains, at a minimum the following
  - —The modem's manufacturer and its serial number
  - —The modem's 48-bit address
  - —The modem's public key
- The modem's X.509 certificate
- The list of ciphersuites supported by the modem
- The modem's Primary SID (which was the first static SID assigned to the modem during MAC registration). The value of a cable modem's Primary SAID is always equal to the value of its Primary SID, so this implicitly generates the first SAID value for this CM at the CMTS.

If it has not already done so, the CMTS authenticates the X.509 certificate. It then returns an Authorization Reply message, containing the following.

- An Authorization Key (of length 160 bits) encrypted with the modem's RSA public key as obtained from the X.509 certificate. The public key must be 1,024 bits in length.[17] The AK is used to derive two message authentication keys (one each for upstream and downstream messages) and a **Key Encryption Key** (**KEK**). The algorithm used to derive these keys is described later, in Key Derivation.

17. Earlier versions of DOCSIS specified a key length of 768 bits. The CMTS must be able to process public keys of both lengths.

- A 4-bit value that acts as a sequence number to distinguish among generations of Authorization Keys
- The lifetime of the Authorization Key
- The SAID of the CM's Primary SA, as well as the SAIDs of any Static SAs for which the modem is authorized to obtain keying information

The Authorization Key

The Authorization Key is a 160-bit key that is encrypted by the CM's 1,024-bit public RSA key. The precise algorithm used for this encryption is the RSAES-OAEP encryption scheme described in version 2 of the PKCS#1 standard, obtainable from *http://www.rsasecurity.com/rsalabs/pkcs/*.

Obtaining TEKs

Each SAID in a cable modem is independent of any other SAIDs that the modem might have. Whenever a key is needed for a particular SAID, the modem sends a Key Request for that SAID to the CMTS. The purpose of a Key Request is to obtain a TEK. The TEK, as the name Traffic Encryption Key implies, is the key that is actually used to encrypt traffic flowing through this SAID. The Key Request contains the following.

- A CM-Identification attribute
- The value of the SAID for which the request is being made
- An HMAC, allowing the CMTS to authenticate the Key Request message. The key for this HMAC is derived from the Authorization Key, using the algorithm described in Key Derivation.

The keying material for the referenced SAID is returned in a Key Reply message containing the following.

- The TEK for this SAID; the key is 3DES EDE encrypted, using a two-key 3DES Key Encryption Key derived from the Authorization Key.
- A CBC initialization vector
- A sequence number for this key
- The remaining lifetime for this key
- An HMAC, which allows the CM to authenticate this message

Figure 3-29 shows this diagrammatically.

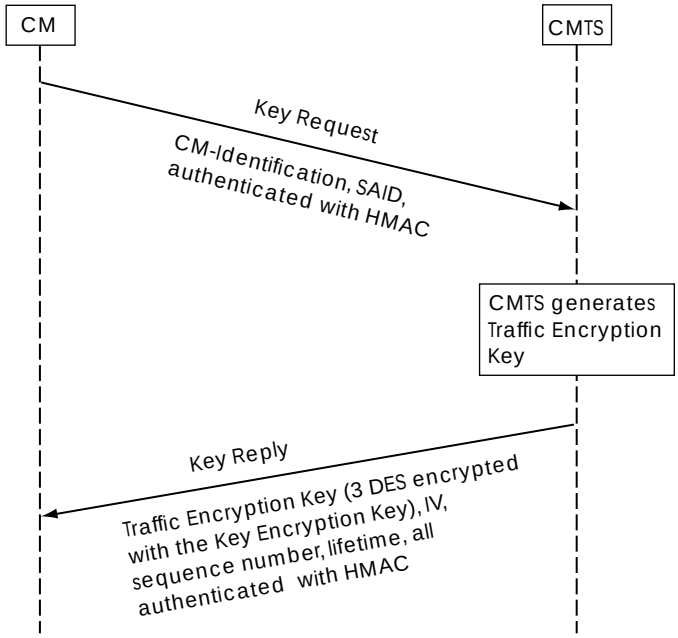


Figure 3-29 Keying Messages in BPKM

So we can summarize the process as follows.

1. The modem authenticates itself to the CMTS and asks for an Authorization Key.
2. The CMTS returns the Authorization Key, encrypted by the modem's public key.
3. Both the modem and the CMTS derive two authentication keys and a single Key Encryption Key from the Authorization Key.
4. The modem requests a Traffic Encryption Key for a particular SAID.
5. The CMTS returns the TEK, encrypted with the KEK, along with other keying material.
6. The TEK is used to encrypt traffic flowing through this SAID.

Key Derivation

The various keys used in BPI+ are derived from a 160-bit Authorization Key (AK) that is generated by the CMTS and passed to the CM, encrypted by the CM's public key. Three keys are derived from AK: the KEK, used to encrypt TEKs as they are passed from the CMTS to the CM; HMAC_KEY_U, the message authentication key

used in upstream Key Requests and HMAC_KEY_D, the message authentication key used in downstream Key Replies and error messages.

The keys are generated as follows.

```
KEK = Truncate(SHA1(K_PAD + AK), 128)
HMAC_KEY_U = SHA1(H_PAD_U + AK)
HMAC_KEY_D = SHA1(H_PAD_D + AK)
```

where:

Truncate(*x*, *n*) means to truncate *x* to its left-most *n* bits;

x + *y* means to concatenate *x* and *y* (in that order)

SHA1(*x*) means to calculate the SHA-1 hash of *x*

K_PAD is a 512-bit string formed by repeating the value 0x53 63 times.

H_PAD_U is a 512-bit string formed by repeating the value 0x5C 63 times.

H_PAD_D is a 512-bit string formed by repeating the value 0x3A 63 times.

TEK Encryption

The TEK that is passed in a Key Reply message is encrypted with the KEK. Traffic is encrypted using 56-bit DES, so a 56-bit TEK is necessary. The CMTS actually generates a 64-bit TEK. The least significant bit of each octet in the generated TEK is ignored, resulting in a 56-bit key suitable for encrypting the traffic.[18]

The mechanism used to encrypt the TEK as it is passed from CMTS to the modem is two-key EDE ECB 3DES, applied as follows.

$$C = Ek_1[Dk_2[Ek_1[P]]]$$
$$P = Dk_1[Ek_2[Dk_1[C]]]$$

where:

P = plaintext 64-bit TEK

C = ciphertext 64-bit TEK

k_1 = 64 leftmost bits of KEK

18. This behavior is nonstandard. Normally, the least significant bit of each octet in a 64-bit DES key is a parity bit, and each octet of the key must be of odd parity. In DOCSIS, the least significant bit is simply ignored.

k_2 = 64 rightmost bits of KEK

E[M] = 56-bit DES encryption of M in ECB mode

D[M´] = 56-bit DES decryption of M´ in ECB mode

Lifetime of Keying Material

In practice, a CMTS maintains two valid sets of keying material for each SAID, staggered so that the lifetime of one set expires midway through the life of the other set. It places both sets into a single Key Reply whenever a modem sends a Key Request for that SAID. This ensures that there is always at least one valid set of keying material for the SAID. Since the CM knows when the keys expire, it can schedule Key Requests appropriately.

As well as periodic Key Requests, the CM also sends periodic (but much less frequent) Authorization Requests. The CMTS generates Authorization Keys with overlapping lifetimes, so that at least one Authorization Key is always valid.

Packet Formats

BPKM messages are carried in the Management Message Payload Field of DOCSIS MAC management messages. The details of the many different kinds of messages are beyond our scope, but they are readily available in the DOCSIS documentation. The generic format of a BPKM message is shown in Figure 3-30. The meanings of the various fields are as follows.

- Code—One octet. Identifies the type of the BPKM packet according to Table 3-7.
- Identifier—One octet. The CM increments the Identifier field whenever it transmits a new BPKM message. When sending a response, the CMTS places the value of the Identifier field in the matching request into its response.

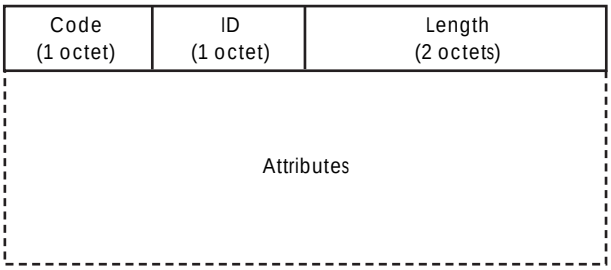


Figure 3-30 BPKM Message Format

Table 3-7 BPKM Message Codes

BPKM Code Octet	*BPKM Message Name*	*MAC Management Equivalent Message Name*
0–3	Reserved	
4	Auth Request	BPKM-REQ
5	Auth Reply	BPKM-RSP
6	Auth Reject	BPKM-RSP
7	Key Request	BPKM-REQ
8	Key Reply	BPKM-RSP
9	Key Reject	BPKM-RSP
10	Auth Invalid	BPKM-RSP
11	TEK Invalid	BPKM-RSP
12	Authent Info	BPKM-REQ
13	Map Request	BPKM-REQ
14	Map Reply	BPKM-RSP
15	Map Reject	BPKM-RSP
16–255	Reserved	

- Length—Two octets. Gives the number of octets in the Attribute field.
- Attributes—Variable length. Carries the specific request or response information as defined by the DOCSIS specifications. Attributes are TLV encoded.

The CM's X.509 Certificate

Ultimately, the security of BPI+ rests in the X.509 certificate inserted into the modem during manufacture. If a CMTS is presented with what it believes to be a valid certificate, it will proceed with BPI+ initialization. In order to authenticate the validity of a CM certificate, BPI+ utilizes the hierarchy shown in Figure 3-31 (see Chapter 2 for more about certificate hierarchies).

The CMTS is preprovisioned with the DOCSIS root RSA public key. This enables it to authenticate certificates claiming to be signed by the DOCSIS root. Manufacturer certificates are such certificates. A manufacturer certificate contains the manufacturer's RSA public key. Since it is signed by the DOCSIS root, which the CMTS

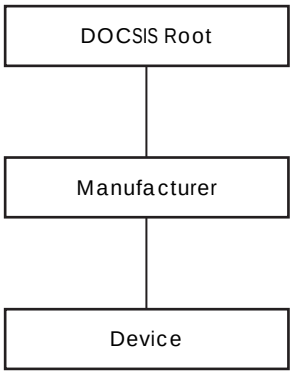


Figure 3-31 DOCSIS Certificate Hierarchy

can verify, then the CMTS can trust that the manufacturer's public key as handed to it by the CM is indeed correct.

The CM certificate, which contains the modem's RSA public key, is signed by the manufacturer. Since the CMTS now has the manufacturer's public key, it can validate this certificate in turn, thereby authenticating the cable modem's public key.

Once it is certain that it has the cable modem's public key, BPI+ initialization can continue with the generation of an Authorization Key on request, and the AK can be delivered securely, encrypted with the modem's public key.

BPI+ MAC Extended Header

BPI+ operates to encrypt the data PDU portion of MAC frames. It signals this by using the Extended Header (EHDR) field of the MAC header.

As we saw earlier, the Extended Header is an optional, variable length field that occurs in the MAC header, immediately following the LEN field. The BPI+ Extended Header is five octets in length, divided as in Figure 3-32.

Type

A 4-bit field that defines the direction of flow of the frame. The value may be either BPI_UP (defined to be 3) or BPI_DOWN (defined to be 4).

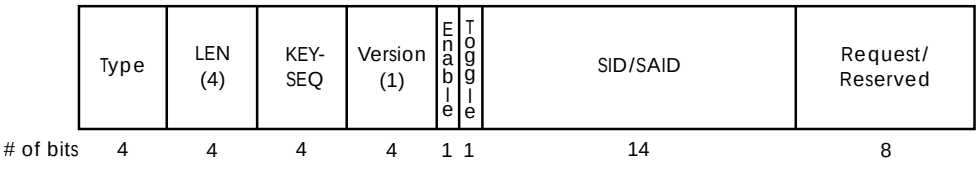


Figure 3-32 DOCSIS BPI+ Extended Header Format

LEN

A 4-bit field containing the value 4

KEY_SEQ

A 4-bit field. This field contains the number of the key that is currently in use for the Security Association ID associated with this frame, starting with zero and incrementing by one every time that the CMTS generates new keying material for this SAID. The value in the field wraps around with every *n*th key generation, where *n* mod 16 is zero.

Version

A 4-bit field containing the value 1

ENABLE

A 1-bit field identifying whether the PDU is actually encrypted. A one indicates encryption, a zero indicates no encryption.

TOGGLE

A 1-bit field that matches the LSB of the KEY_SEQ field

SID/SAID

A 14-bit field. In the upstream, this field contains the SID of the frame; in the downstream, it contains the Security Association ID.

REQUEST/Reserved

A 1-octet field. In the upstream, this field contains a number of minislots that may optionally be requested for upstream bandwidth. In the downstream, the field is set to zero and is ignored.

Where Do We Go from Here?

In this chapter we have looked at the essential facilities provided by DOCSIS modems operating on an HFC access network.

These facilities are available to all the traffic being handled by the modems. You have probably noticed that there was little specific mention of telephony in this chapter. However, most of the concepts we have examined—in particular the difficulty of obtaining guaranteed upstream bandwidth and the security applied to DOCSIS transmissions—are important grounding for understanding the design of the PacketCable telephony network.

In the next several chapters we will examine how the facilities provided by DOCSIS modems are used as an integral part of a fully fledged telephony network capable of providing high quality and advanced services to large numbers of subscribers.