5 Standards for multimedia communications

5.1 Introduction

In Chapter 1 we identified the different types of communication networks that are used to provide multimedia services. We also described a range of multimedia applications that use these networks and, although these are many and varied, we showed that they can be classified into one of three categories: interpersonal communications, interactive applications over the Internet, and entertainment applications.

In Chapter 2 we described the way the different types of media that are used in multimedia applications – text, images, speech, audio, and video – are represented in a digital form and we concluded that a number of the network types used can only support multimedia applications as a result of the technological advances that have taken place in the field of compression. Hence in Chapters 3 and 4 we described a selection of the different algorithms and standards that are used for the compression of the different media types.

In general, however, most of the multimedia applications we described in Chapter 1 involve not just a single type of medium but rather a number of media types that are integrated together in some way. For example, an

interpersonal application such as videoconferencing – and most entertainment applications – involve speech and video integrated together while a typical interactive application over the Internet involves text and images integrated together. Hence in addition to the standards that we described for the compression of the different types of media, a range of application-level standards have been defined that are concerned with how the integrated information streams associated with the various applications are structured. And since the different types of network that are used to support multimedia applications operate in different ways, there is often a range of standards associated with a particular application, each intended for use with a specific type of network.

Standards are necessary because it is essential that the two or more items of equipment that are used for the application interpret the integrated information stream in the same way. They are also necessary at the networking level. For instance, with all types of network there is a finite probability that the information bitstream received from the network will contain bit/transmission errors and hence it is necessary also to ensure that both communicating parties utilize the same standards for detecting the presence of bit errors in the received information stream and, in some instances, requesting that another (hopefully error-free) copy of the information is sent. This is just one aspect of what is called a **communications protocol**. Other aspects include the initiation and clearing of a communications session between the two communicating applications and, in some instances, the setting up and clearing of a connection through the particular network being used.

In this chapter we present an overview of the standards that have been defined for use with multimedia communications. In practice, these are many and varied and relate both to the operation of the various networks that are used and also to the hardware and software in the computers (and other types of terminal equipment) that are connected to these networks to provide their users with access to multimedia communication services. A common framework known as a **reference model** is used for defining the various standards. We shall describe the structure of this in the next section and a selection of the standards that have been defined for use with specific multimedia applications in the following sections.

5.2 Reference models

Although the range of multimedia applications (and the different network types that are used to support them) are numerous and varied, the standards associated with the three types of application mentioned in the introduction have a common structure, as we show in Figure 5.1.

As we can see, standards are required at both the application level and the networking level, the latter including those for interfacing equipment to the network – network interface standards – and those relating to the

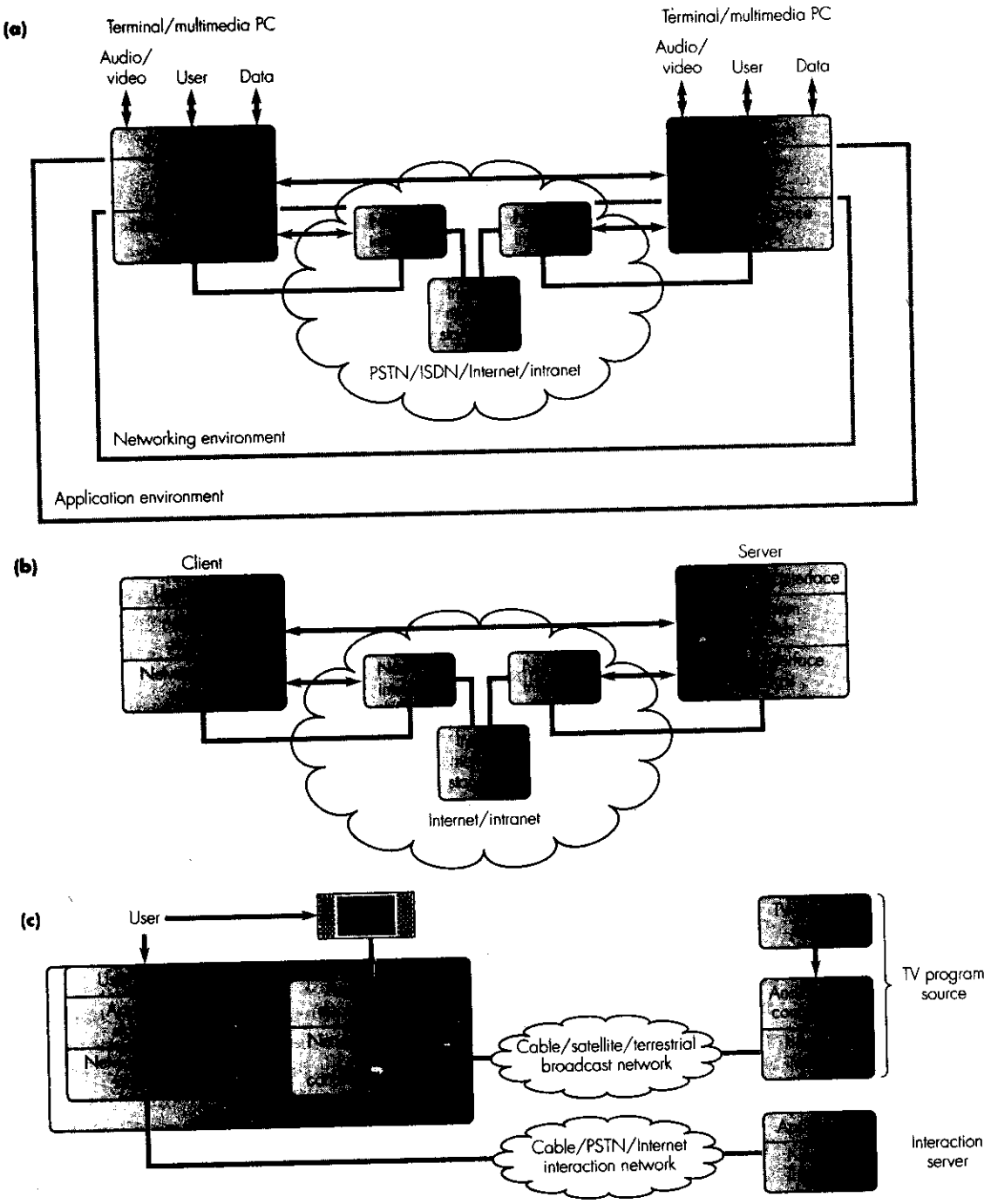


Figure 5.1 Standards requirements for multimedia applications: (a) interpersonal; (b) interactions over the Internet; (c) entertainment.

internal operation of the networks – internal networking standards. The functionality of each set of standards is as follows:

- **application standards:** these provide users, through an appropriate interface, with access to a range of multimedia communication applications;
- **network interface standards:** as we explained in section 1.5.3, different types of network operate in different modes – circuit-switched or packet-switched, connection-oriented or connectionless – and hence each network type has a different set of standards for interfacing to it;
- **internal network standards:** these are concerned with the internal operation of the network and again differ from one type of network to another.

As we show in the figure, the last two standards are concerned solely with networking issues and are said to be within the **networking environment**. Also, in the case of network interface standards, since these operate over the access circuit to the network, they are said to have only **local significance**. Application standards, however, are network-independent and relate to communications between the two (or more) terminals/computers involved in the application. Normally, the latter are referred to as **end systems** and these communications are said to have **end-to-end significance**. The application standards build on the set of networking standards to create what is known as the **application environment**.

In practice, associated with each standard is the set of procedures that are to be used to perform the particular function. Examples include, the content and structure of the source information stream associated with an application, how the information stream is formatted prior to its transmission over the network, the way transmission errors are detected, the procedure that is to be followed to obtain another copy of a corrupted block of information, and so on. Clearly, for each function, both communicating parties must adhere to the same set of procedures and collectively these form the communications protocol relating to that function.

The communication subsystem that is required in each end system associated with an application is a complex piece of hardware and software. Early implementations of the software for such subsystems were often based on a single, complex, unstructured program – normally written in an assembly language – with many interactions between the different parts. As a result, the software was difficult to test and often very difficult to modify. To overcome these deficiencies, later implementations were based on a **layered architecture**. This means that the complete communication subsystem is broken down into a number of protocol layers each of which performs a well-defined function.

The actual protocol layers that are used for each type of application differ and are influenced strongly by the type of equipment that is used to provide the application. In the case of interpersonal applications, for exam-

ple, the user equipment can be either a terminal that is dedicated to providing this type of application or a multimedia PC/workstation that can be used to support both interpersonal and interactive applications. With a multimedia PC/workstation, the protocol layers that are normally used are based on what is called the **TCP/IP reference model** and hence we shall use this in order to identify and describe the function of the different protocol layers present. In general, the protocols that are used with a dedicated terminal are a subset of these.

5.2.1 TCP/IP reference model

As we indicated in the introduction, a reference model is simply a common framework for defining the specific set of protocols to be used with a particular application/network combination. The resulting set of protocols are then known as the **protocol stack** for that application/network combination. In the case of the TCP/IP reference model, the name is derived from two of the protocols – rather than layer functions – that form the core of the protocol stack used with the Internet. More recently, however, the choice of layer functions that were used for the TCP/IP protocol stack have been used in a more general way for other multimedia applications. This has been achieved by making the network interface layers applicable to different types of network rather than just the Internet. The general structure of the modified model in relation to the original model is shown in Figure 5.2(a) and part (b) shows its application in relation to the various standards we identified in Figure 5.1. We shall describe the role of each layer in the modified model separately.

Physical layer

The physical layer is concerned with how the binary information stream associated with an application is transmitted over the access circuit to the network interface. As we shall expand upon in Chapter 6, the signals that are used to represent each binary 1 and 0 in the stream vary for different types of access circuit. Hence depending on the choice of access circuit, it is also necessary to select a specific way of representing each binary 1 and 0 as well as the physical dimensions of the plug and socket to be used to connect the terminal/computer (end system) to the access circuit termination. Typically, this is in the home or office and again a number of alternatives are possible.

Link layer

Although selected applications such as telephony over a PSTN or an ISDN generate a constant bit rate stream that is transmitted transparently over the total network, most multimedia applications involve multiple media types integrated together in some way. Hence, as we shall expand upon in the following sections, the more usual form of representing the source information stream is in the form of a contiguous stream of blocks with each block containing the integrated media stream associated with the application. Also, as

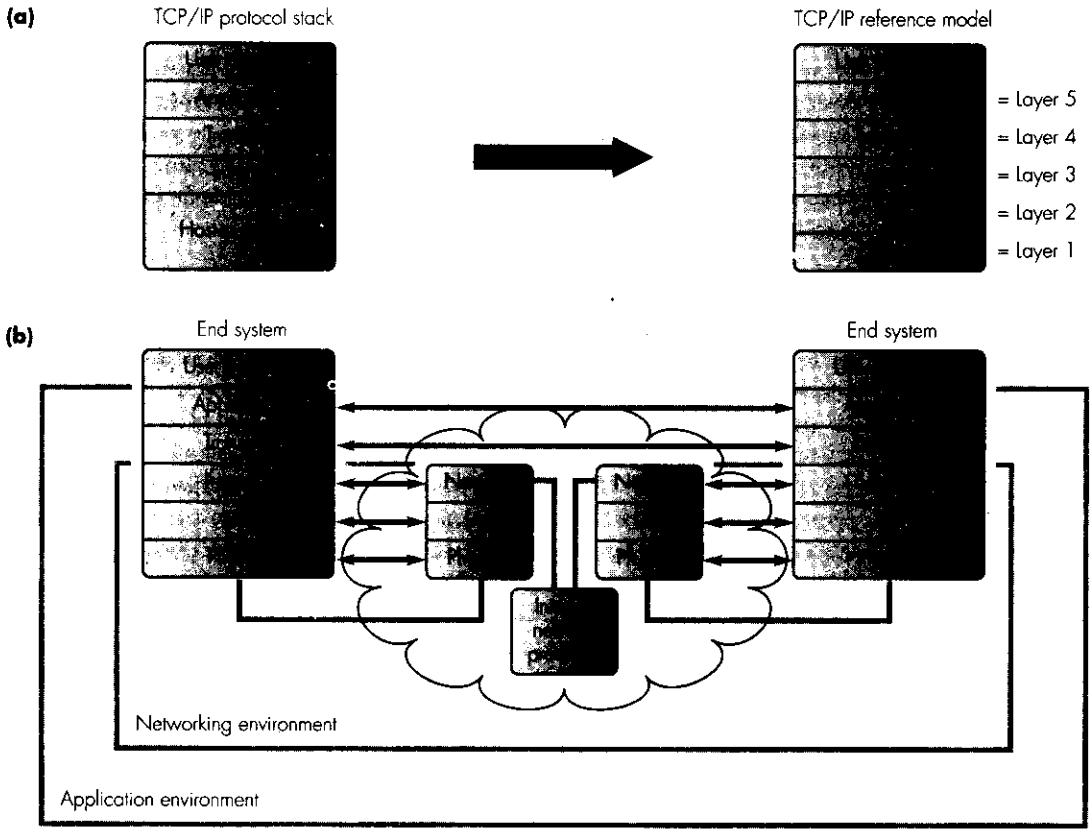


Figure 5.2 TCP/IP reference model: (a) evolution; (b) application.

we described earlier in Section 1.5.3, in a packet-switched network, as each packet is received by a packet-switching exchange, it is checked for the presence of transmission errors. It is the role of the link layer to indicate the start and end of each block within the source bitstream and, in a packet-switched network, to add error check bits to the information bitstream for error detection – and in some instances error correction – purposes.

Network layer

The network layer is concerned with how the source information stream gets from one end system to another across the total network. With a connection-oriented network, this involves the setting up of a network connection, the exchange of the information relating to the call/session, and the subsequent clearing of the connection. In the case of a connectionless network, it involves formatting the source information into packets, each with the unique network-wide address of the two (source and destination) communi-

cating end systems at its head. As we can see from this, there are a number of different network layer protocols, each concerned with a particular type of network. Also, as with the link and physical layers, the network protocol is not concerned with the content of the information stream being transferred/exchanged but simply how it is transferred.

Transport layer

It is the role of the transport layer to mask the differences between the service offered by the various network types from the application layer and instead, to provide the application with a network-independent information interchange service. In addition, since there can be multiple applications running in the same end system concurrently, the transport layer is also responsible for directing each information flow to and from the related application.

As we indicated in Section 1.5.5, all of the networks that are used for multimedia applications provide a best-effort service. This means that with a circuit-switched network, bit errors may be present in the constant bit rate information stream after its passage through the network and, with a packet-switched network, some packets may be missing from the received stream as those which incur bit errors will have been discarded by the network. For applications such as telephony, this is acceptable. For other applications, however, it is essential that only error-free information is received by the destination application. Hence there are two types of service provided by the transport layer, the first known as an **unreliable transport service** and the second a **reliable transport service**. Thus for a particular application, the appropriate service – and hence protocol – is chosen.

Application layer

The application layer provides the user, through a suitable interface, with access to a range of multimedia communication services. Examples include email, Web access, telephony, videoconferencing, and so on. Associated with each application is a specific application protocol which provides the user with the corresponding service. Typically, therefore, the application layer in an end system contains a selection of application protocols, each providing a particular service. In the case of interpersonal communications, these include application protocols for email, telephony, and videoconferencing while in interactive applications, they include application protocols for access to remote information servers and servers containing stored videos/movies.

5.2.2 Protocol basics

Each layer performs a well-defined function in the context of the overall communication subsystem. The protocol to be used at each layer is chosen to meet the needs of a particular application/network combination. The two communicating protocols within each layer operate according to a defined set of rules in order to implement the desired function of the layer. Normally,

this is achieved by adding appropriate **protocol control information (PCI)** to the head of the information being transferred. The complete block – information plus PCI – is known as a **protocol data unit (PDU)** and this is then sent to the corresponding protocol in the remote system. The two procotols are said to operate at the same **peer layer** within the protocol stack.

In practice, although the two protocols communicate by exchanging protocol control information at a peer level, as we show in Figure 5.3(a), the actual PDUs containing the PCI are transferred using the services provided by the protocol layer immediately below it in the protocol stack. Each layer provides a defined set of services to the layer immediately above it. The selected protocol at that layer then implements these services by communicating with the peer protocol in the remote system according to the defined protocol. Hence as we show in part (b) of the figure, as the information to be transferred is passed down from one layer to the next, the protocol at each layer adds its own PCI at the head of what it receives and, once the link layer protocol has added its own PCI – including the error check bits at the tail – it is this that is encoded and transmitted over the network to the remote system. Conversely, as the received information stream is passed up from one layer to the next in the remote system, each layer protocol reads and removes its own PCI from the head and, after interpreting this according to the defined protocol for that layer, passes the remaining information up to the protocol layer immediately above it.

5.3 Standards relating to interpersonal communications

As we described in Section 1.4.1, interpersonal communications such as telephony, video telephony, data conferencing, and videoconferencing are provided both by circuit-mode networks such as a PSTN or an ISDN and packet-mode networks such as a LAN, an intranet, or the Internet. Most of the standards relating to these applications have been defined by the ITU-T and there are separate standards relating to both types of network. Before describing these standards, it will be helpful if we first identify the use of the different constituent parts that they contain.

To do this, consider an application involving two design engineers working jointly on a project and having a meeting relating to the project over a network. Typically, each will be using a multimedia PC or workstation and an example communication session may comprise multiple phases along the following lines.

First, the session may start with a telephone conversation and, during this, the design/image on each engineer's screen – normally referred to as **user data** – is sent to the other party. They then decide to bring in a third person who is working (remotely) on the project and, since one of the engineers has not met this person, they convert to a videoconferencing call and, during this phase, all three members start to discuss the design jointly.

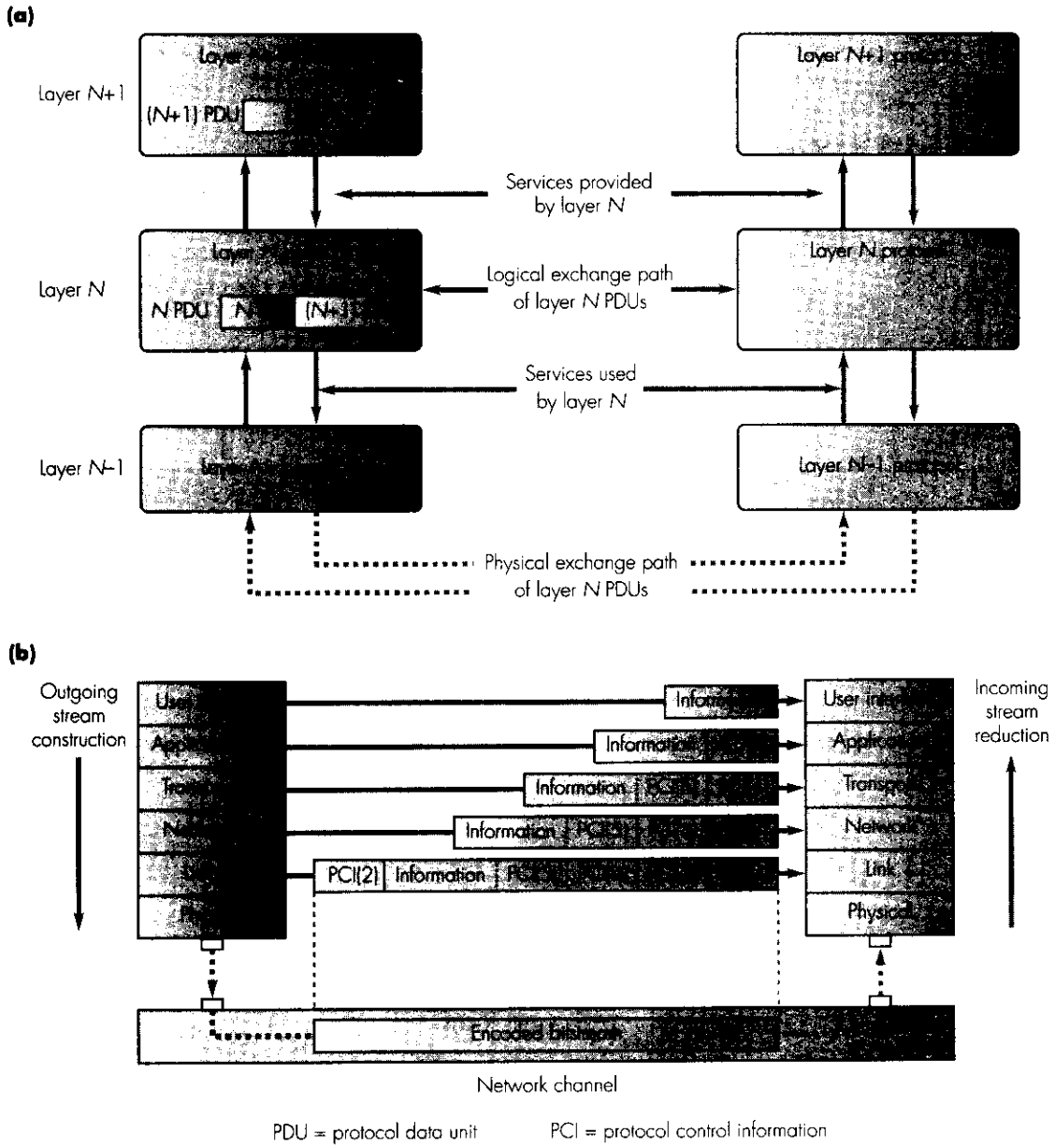


Figure 5.3 Protocol basics: (a) layer interactions and terminology; (b) end-to-end transfer.

As we can deduce from this example, the content of the information stream being transferred varies. It starts with speech only, then speech with user data, then speech with video, and finally speech, video, and user data. In the case of an ISDN, each media type is allocated a fixed portion of the

channel bandwidth and hence each can be used as and when required. With a PSTN and most packet-switched networks, the appropriate amount of bandwidth is allocated on-demand as the session progresses.

5.3.1 Circuit-mode networks

The functionality of the various standards that are used in end systems connected to a circuit-mode network is shown in Figure 5.4.

As we explained earlier in Section 1.5.3, with networks such as a PSTN (plus modem) and an ISDN, once a connection through the network has

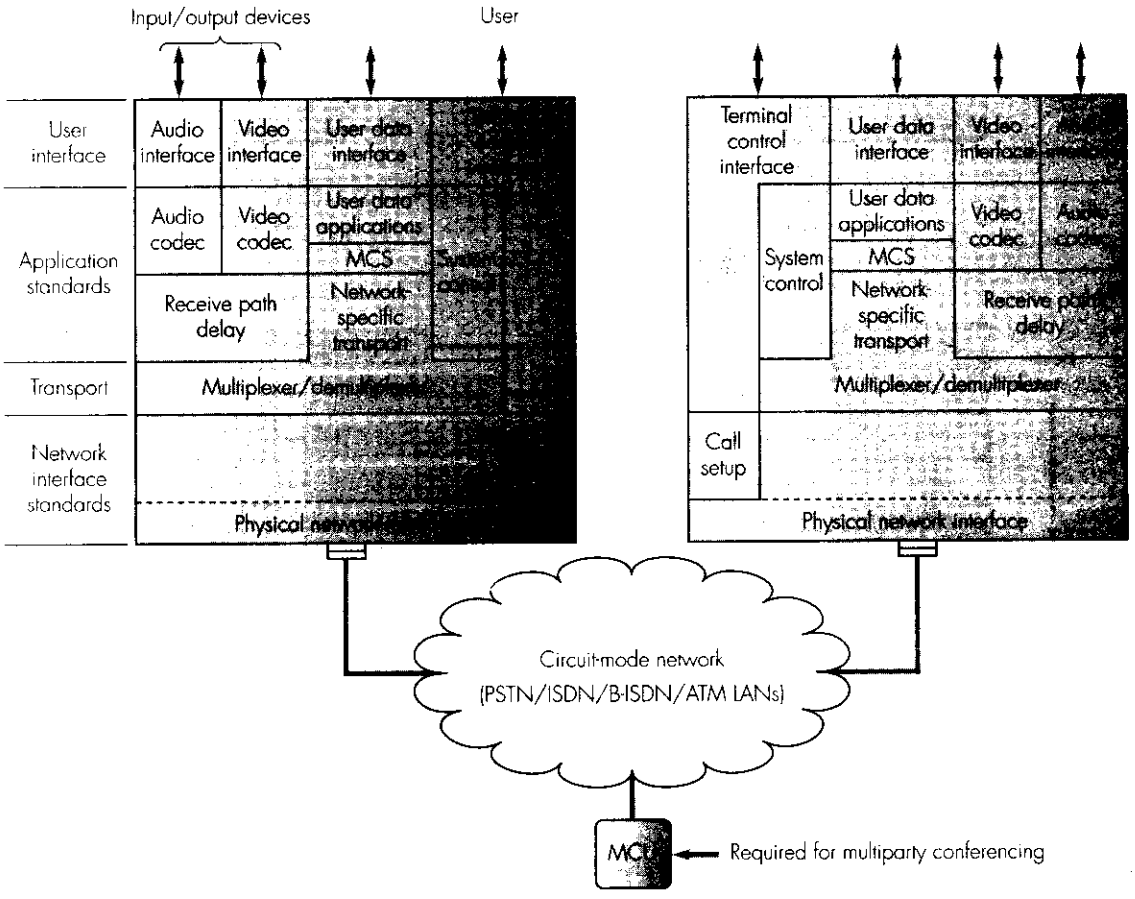


Figure 5.4 Structure of interpersonal communication standards for circuit-mode networks.

been set up, the connection provides a constant bit rate channel over which the user can transfer any type of digital information. Hence in relation to the TCP/IP reference model we described in Section 5.2.1, the network interface standards relate primarily to the physical connection to the network termination and with the procedures followed to set up and clear/tear-down a connection. The basic transport layer function is provided by the multiplexer/demultiplexer. Essentially, the multiplexer merges the source information from the three application streams – audio, video, and user data – and the system control application into a single stream for transmission over the constant bit rate channel provided by the connection and, on receipt of the merged stream, the demultiplexer routes the constituent streams to the corresponding application.

The system control application is concerned with negotiating and agreeing on the operational parameters to be used with the call/session. These are based on the **capabilities** of the end systems involved in the call and enable, for example, a simple terminal to communicate with a more sophisticated terminal/computer. With a PSTN, the system control function includes the management of the available transmission bandwidth during a call. Both of these functions are achieved by the two system control application protocols exchanging appropriate messages – protocol data units – over the network connection. In the case of a two-party call, this involves the applications in the two end systems communicating directly with each other while with a multiparty conference call, it involves each end system communicating with a multipoint control unit (MCU) as we explained in Section 1.5.4.

The audio and video codecs each use a particular compression algorithm which is appropriate for the application and within the bandwidth limits provided by the network. Also, in order to synchronize the audio and video streams – to achieve **lip-sync** for example – a delay is often introduced at the receiver into one of the streams. As we explained in the previous section, the user data application is concerned with the transfer of data such as a digitized image or the contents of a file. Normally, therefore, the data is sent in the form of individual blocks/packets and the application uses an appropriate reliable transport protocol to overcome lost/corrupted packets. In addition, if the user data is to be shared between the various members of a conference, the application uses the services provided by a protocol known as a **multipoint communications service (MCS)**. A copy of this is present in each end system and this, in turn, relays a copy of all transmitted data to the other members of the conference via the MCU.

The actual standards that have been defined for use with the different types of circuit-mode network – including a B-ISDN and local area networks that provide a guaranteed bandwidth such as an ATM LAN – are summarized in Table 5.1. The standard/recommendation at the head of each column is a system-level standard and embraces a number of additional standards for the various component functions such as audio and video compression, and so on. We shall describe the main features of each standard separately.

Table 5.1 Summary of the standards used with the different types of circuit-mode network.

Standard	H.320	H.324	H.321	H.310	H.322
			B-ISDN (ATM)	B-ISDN (ATM)	Guaranteed bandwidth LANs
	G.711* G.722 G.728	G.723.1*	G.711* G.722 G.728	G.711* G.722 G.728 MPEG-1*	G.711* G.722 G.728
		H.261* H.263*	H.261	H.261* MPEG-2*	H.261
	T.120		T.120	T.120	T.120
	H.223		H.221	H.221 H.222	H.221
System control	H.242	H.245	H.242	H.245	H.242
	V.25		Q.931	Q.2931	Q.931

* = mandatory

H.320

The H.320 standard is intended for use in end systems that support a range of multimedia applications over an ISDN. Hence, as we described in Section 1.3.4, the usable bandwidth of a connection is $p \times 64$ kbps where p can be 1 through 30. For video telephony, for example, p is either 1 or 2 while for videoconferencing p is 2 or greater.

Audio The choice of audio/speech compression standard can be selected from one of three ITU-T recommendations: G.711, G.722, and G.728. The choice is determined primarily by the amount of transmission bandwidth available for the audio. For example, as we described in Section 2.5.1, G.711 relates to standard PCM and requires 64 kbps of bandwidth. Alternatively, as we described in Section 4.2.2, the G.722 standard gives an improved performance with the same bit rate but at the expense of added complexity in the codec. Clearly, however, since both standards require 64 kbps, they can only be used in multimedia applications when multiple 64 kbps channels are being used. If only a single 64 kbps channel is available, then the G.728 standard

described in Section 4.2.5 must be used and, although the perceptual quality of the speech is inferior to standard PCM, the bit rate required is only 16 kbps. This leaves 48 kbps for the other media types which is typical of many video telephony applications. In general, the other two standards are selected primarily for videoconferencing applications as these need the higher quality speech to discriminate between the voices of the different members of the conference.

Video The video compression standard is H.261 and, as we described in Section 4.3.2, a constant output bit rate from the encoder (of the allocated rate) is obtained by varying the quantization threshold that is used dynamically. The video resolution can be either quarter-screen (QCIF) or full-screen (CIF), the actual resolution used being negotiated at the start of the conference.

User data The user data applications are based on the **T.120 standard** which, in practice, consists of a set of recommendations. As we show in Table 5.1, the same standard is used with all the different types of circuit-mode network. There is a set of application-specific recommendations that support the sharing of various media types:

- T.124: sharing of text for what is known as text chat,
- T.126: still-image and whiteboard sharing,
- T.127: sharing of file contents(text and binary),
- T.128: sharing of text documents and spreadsheets,

and a series of communications-related recommendations:

- T.122: multipoint control unit (MCU) procedures,
- T.125: multipoint communication services (MCS) procedures,
- T.123: a series of network-specific transport protocols all of which provide a reliable transport service.

The standard also includes extensions to allow the use of non-standard protocols to be negotiated.

System control/call setup The call setup (signaling) procedure associated with an ISDN is defined in **Recommendation Q.931** and, as we shall expand upon in Section 7.4.1, this involves the exchange of messages over a separate 16 kbps channel known as the **signaling channel**. Also, with an ISDN the bandwidth associated with the audio, video, and data streams are negotiated and fixed at the start of a conference. Hence the system control standard – **Recommendation H.242** – is concerned primarily with the negotiation of the bandwidth/bit rate to be used for each stream. Once an end system has set up a connection to the MCU, the system control within the end system informs

the multipoint controller part of the MCU of its capabilities. These include the video format (QCIF or CIF) and compression standards it supports, the audio compression standards, the T.120 user data applications, and the proposed bit rate of each channel. The MCU then negotiates and agrees a minimum set of capabilities so that all members of the conference can participate.

Multiplexing The multiplexer/demultiplexer layer is defined in **Recommendation H.221** and describes how the audio, video, and data streams are multiplexed together for transmission over the network. Normally, the fixed portions of the available bandwidth are allocated using a technique known as **time division multiplexing (TDM)** which we will describe in more detail in Section 7.2.3. Hence the role of H.221 is to ensure that each input stream is placed into its allocated position in the output bitstream and, at the receiver, to pass each stream to the appropriate application.

H.324

The H.324 standard is intended for use in end systems that support a range of interpersonal communication applications over low bit rate switched networks such as a PSTN. In general, therefore, the network interface is a modem which, in the case of a PSTN, provides a bit rate of up to 56 kbps. Normally, as we shall expand upon in Section 7.2.2, the modem contains auto dial and auto answer facilities for the establishment of a call/network connection using the standard dialing, ringing, and answering procedure.

Video and audio The video compression standard can be either H.261 or the H.263 standard which we described in Section 4.3.3 of the last chapter. As we explained, the H.263 standard uses the same compression technique as H.261 but contains a number of more advanced coding features in order to operate over lower bit rate channels such as those provided by a PSTN. The audio compression standard is either G.723.1 or G.729, both of which we described in Section 4.2.5. The G.723.1 standard is the most common and, as we explained, operates at a bit rate of either 5.3 or 6.3 kbps. Also, although there is an algorithmic (codec) delay associated with both codecs, this is normally less than the video codec delay. Hence in order to obtain lip-sync, a delay has to be added to the audio stream at the receiver. The actual delay is measured at the sending side and, as part of the H.245 system control protocol, a message containing the required delay is sent to the receiver.

User data The user data application standard is T.120 and basically this contains the same set of protocols as are used in an H.320-compliant terminal except for the network-specific transport protocol, T.123. The H.223 multiplexing standard, however, is different. Because of the relatively low – and possibly variable – bit rate that is available, the audio, video, and user data streams are not allocated fixed portions of the available bandwidth but rather these are negotiated using the H.245 system control protocol. This occurs

both prior to a call commencing and as the call is in progress and, at any point in time, each stream may be present or not. For example, when a large file is to be transferred, it is possible to temporarily suspend the video stream until the file contents have been sent.

Multiplexing The total channel bandwidth is divided into a number of separate **logical channels** each of which is identified by means of a **logical channel number (LCN)**. The first – LCN0 – is used to carry the control stream and each of the remaining channels carries a separate media stream. The allocation of LCNs is controlled by the transmitter and, when it wishes to open a new channel, it sends an H.245 control message which includes the media type and the type of codec being used. The role of the multiplexer is then to merge those streams that are currently present into the available bandwidth. This is achieved by using what is known as a **bit-oriented protocol,** the principles of which are shown in Figure 5.5

We shall describe the bit-oriented mode of transmission in more detail in Section 6.5.3. Essentially, however, the transmitted bitstream is treated as a string of bytes and, as we show in the figure, this is divided into a number of separate information fields. Each field comprises a variable number of bytes and is separated by one or more **flag bytes**. These have the bit pattern 01111110 and a technique known as **zero bit insertion and deletion** is used to ensure that this pattern cannot be present in the information field. At the start of each information field is a header (byte) and the combined header plus information field is known as a **multiplex protocol data unit** or **M-PDU**. The header byte includes a **multiplex code** which specifies a particular mix of media and control logical channels, and, since each M-PDU may contain a different code, a different mix of logical channels.

As we show in Figure 5.5, the multiplex code is 4 bits in length and forms the index to a table known as the **multiplex table**. A copy of the table is held by both the transmitter and receiver and each entry in the table specifies a particular sequence of logical channels and the number of bytes in each channel. Hence the transmitter can readily change the usage of the available bandwidth (by the various sources) very quickly by simply changing the multiplex code in the header byte. Although in theory there can be a large number of different mixes of logical channels, at any point in time, the table contains just 16 entries. For a particular application, this is normally sufficient but, as part of the system control function, a new set of entries can be sent by the transmitter should this be required.

Adaptation In order to allow for the possibility of transmission errors being present in the received byte stream associated with each logical channel, additional bytes are added by the transmitter for error detection purposes to enable the receiver to detect the presence of errors and, if necessary, to request another copy of the affected bytes. This function is also part of the H.223 standard and is known as the **adaptation layer**. In general, the different types of media – audio, video, and user data – require different levels of

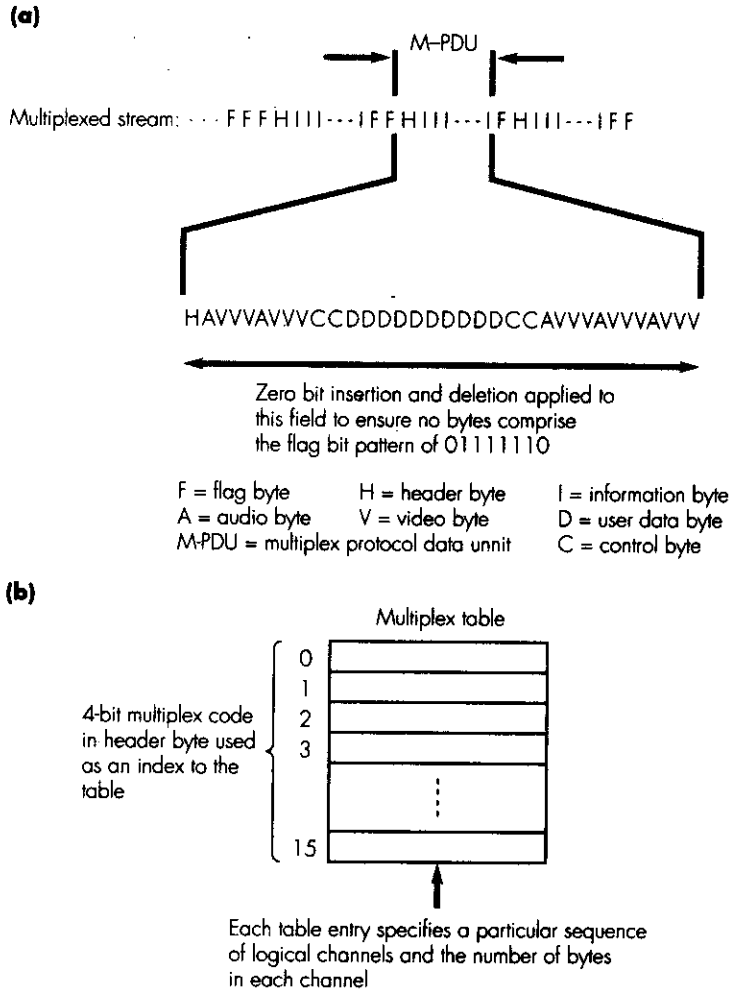


Figure 5.5 H.223 multiplex principles: (a) structure of the multiplexed byte stream; (b) multiplex table usage.

protection against transmission errors. Hence the adaptation layer supports three different schemes – AL1, 2, and 3 – each of which provides a different error-handling capability. The user then selects the most appropriate scheme to meet the requirements of the application. For example, AL1 is intended for use with user data applications since it has features to support the retransmission of any corrupted blocks of data. AL2 is intended for use with the audio and video streams as it supports error detection but retransmission is optional. AL3 is intended for use with video applications that require a higher level of protection. It again supports error detection and also the retransmission of corrupted blocks.

Multipoint conferencing The H.324 standard also supports multipoint conferencing via an MCU. However, since the modems associated with each end system may be operating at different bit rates, during the establishment of the conference the MCU negotiates an agreed minimum bit rate with all the participants by the exchange of system control messages. In addition, interworking between an H.324 terminal and an H.320 terminal is supported either by a device known as an **H.324/H.320 gateway** or an MCU and **dual-mode terminals** which support both interfaces. In the case of a gateway, this communicates with an H.324 terminal using the H.223 multiplex standard and with an H.320 terminal using the H.221 standard. The role of the gateway is then to convert the content of the audio, data, and control streams into/from the appropriate format. This procedure is known as **transcoding** and, in the case of the audio stream, this must be carried out in real time as the audio stream is relayed by the gateway. In practice, this requires a significant amount of processing as it converts from one audio format into another. In the case of the video stream, the processing required is, in general, too great and normally, therefore, the video stream is not transcoded and instead the H.261/QCIF combination is negotiated prior to the conference commencing.

System control The H.245 system control standard is concerned with the overall control of the end system and, as we can see from the above, this involves many functions. These include the exchange of messages for the negotiation of capabilities, the opening and closing of logical channels, the transmission of the contents of the multiplex table, and the choice of adaptation layers. All the messages are defined in a standard syntax known as **Abstract Syntax Notation One (ASN.1)** and an associated encoding scheme is used to ensure that the exchanged messages are interpreted in an unambiguous way by all types of end system. We describe these in more detail in Section 13.2. Because of their importance, a separate error control scheme is applied to all system control messages before they are multiplexed with the three application media streams.

H.321/H.310

Both the H.321 and the H.310 standards are intended for use with terminals that provide a range of multimedia applications over a B-ISDN which, as we explained in Section 1.3.5, is also known as an ATM network.

The H.321 standard relates to the provision of interpersonal communication applications over a B-ISDN and, in practice, is an adaptation of the H.320 standard that is used with an ISDN. Hence, as we can see from Table 5.1, all the standards associated with H.321 are the same as those used with H.320. This simplifies the interworking across both types of network and the only difference is that the network interface layers associated with H.321 relate to interfacing the end system to a B-ISDN rather than an ISDN.

The H.310 standard is intended for use with end systems that support not only interpersonal applications but also interactive and entertainment

applications. Hence, as we can see from Table 5.1, the audio and video standards include the H.321 set which are intended for interpersonal communications and also additional standards – MPEG-1 audio and MPEG-2 video plus their associated multiplex standard H.222 – for use in interactive and entertainment applications. Also, in order to support the wider range of applications, the more comprehensive system control standard H.245 is used. We shall defer the description of the standards associated with interactive and entertainment applications until Sections 5.4 and 5.5 respectively.

H.322

The H.322 standard is intended for use with end systems that support interpersonal communication applications over a local area network (LAN) that provides communication channels of a guaranteed bandwidth. Examples include ATM LANs, the operation of which we shall explain in Section 10.5. In general, the communication channels associated with these networks are able to support multiples of 64 kbps and hence, as we can see from Table 5.1, the same set of standards are used with H.322 as are used with an H.320 end system in order to simplify interworking across both types of network.

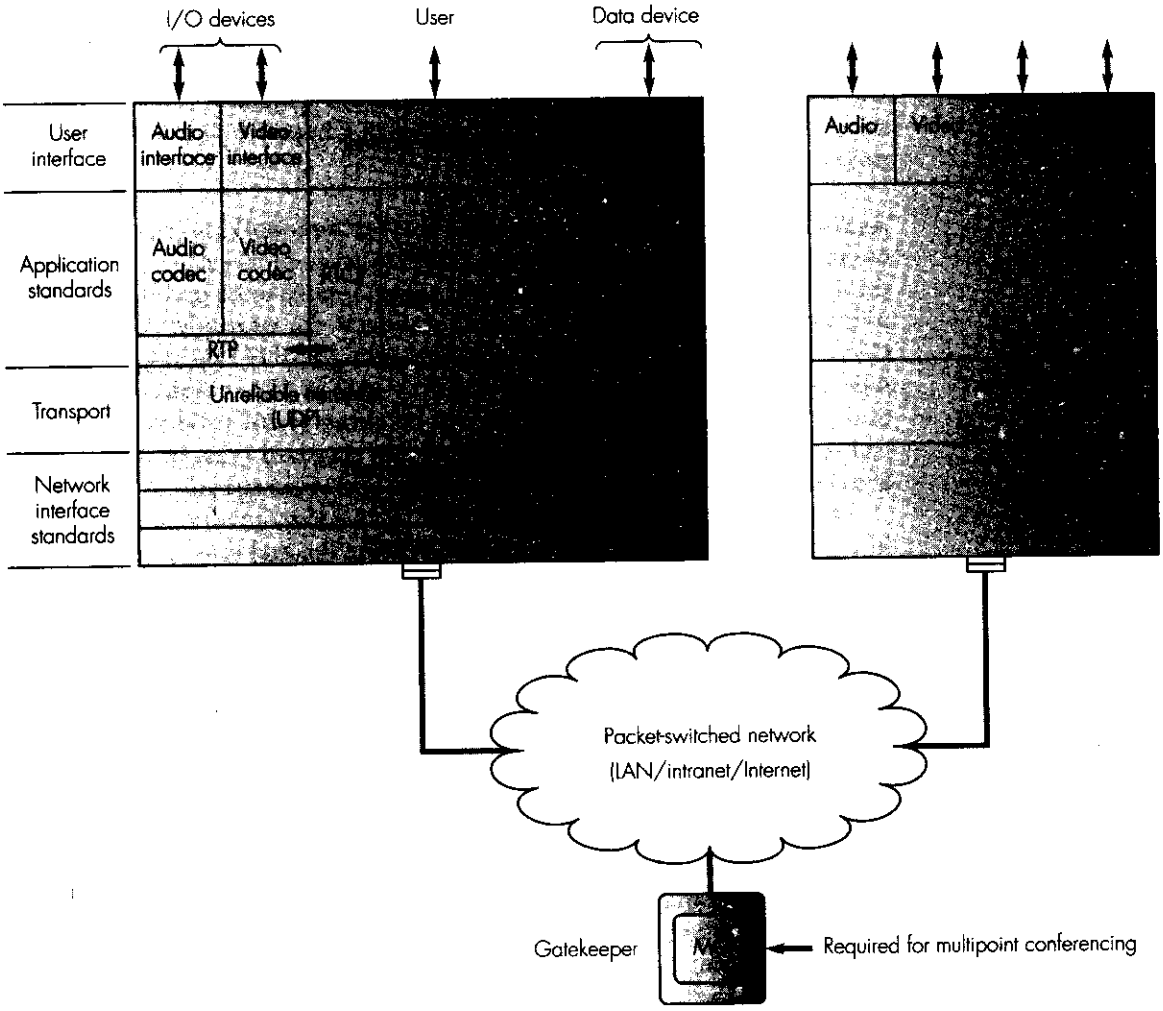
5.3.2 Packet-switched networks

Two alternative sets of protocols have been defined for providing interpersonal communication services over packet-switched networks, one defined by the ITU in Recommendation H.323 and the other by the IETF. In this section we shall discuss both standards.

H.323

As we described earlier in Section 1.3.2, normally, the access network used with both an intranet and the Internet is a campus/site LAN. Hence the H.323 standard pertaining to packet-switched networks relates primarily to how interpersonal communications are achieved between end systems that are attached either to the same LAN or to different LANs that are interconnected together in some way. Unlike the H.322 standard which relates to LANs that offer a guaranteed bandwidth/QoS, the H.323 standard is intended for use with LANs that provide a non-guaranteed QoS which, in practice, as we shall expand upon in Section 8.2, applies to the majority of LANs.

As we show in Figure 5.6, the standard comprises components for the packetization and synchronization of the audio and video streams, an admission control procedure for end systems to join a conference, multipoint conference control, and interworking with terminals that are connected to the different types of circuit-switched networks. The standard is independent of the underlying transport and network interface protocols and hence can be used with any type of LAN. It is assumed, however, that the transport layer provides both an unreliable (best-effort) service and a reliable service which, in practice, is the case for most LANs. With an end system connected to a



MCS = multipoint communication services
MC = multipoint controller

RTP = real-time transport protocol
RTCP = real-time transport control protocol
RAS = request access service

UDP = user datagram protocol
TCP = transmission control protocol
IP = internet protocol
} examples only

Audio codec options: G.711/722/723.1/728/729
Video codec options: H.261/263

Figure 5.6 Structure of the H.323 interpersonal communication standards for packet-switched networks.

LAN and communicating over an intranet or the Internet, for example, the network layer protocol is the **internet protocol (IP)**, the unreliable transport service is provided by the **user datagram protocol (UDP)**, and the reliable transport service by the **transmission control protocol (TCP)**. We shall describe the operation of the IP protocol in Chapter 9 and both transport protocols in Chapter 12.

Audio and video coding In order to simplify interworking with terminals/computers that are attached to the different types of circuit-mode networks, the H.323 standard allows a variety of coding options to be used for the audio and video streams. The audio codec standard, for example, can be either G.711 or G.728 in order to simplify interworking with H.320-compliant terminals or G.723.1 or G.729 for interworking with H.324-compliant terminals. Similarly, the video codec standard can be either H.261 or H.263. Prior to a call commencing, however, an agreed coding standard must be negotiated to avoid the necessity of transcoding the audio and video streams.

The output streams of both the audio and video codecs are formatted into packets for transfer over the network using the **real-time transport protocol (RTP)**. As we shall explain in Section 12.5.1, this is used for the transfer of real-time information and, at the head of each RTP packet is a format specification which defines how the packet contents/payload are structured. There are standardized formats for the information streams produced by all the different audio and video codecs. In addition, as part of the **real-time transport control protocol (RTCP)**, the sending end system sends information to enable the receiving end system to synchronize the audio and video streams. Other information transferred as part of the RTCP includes the transmitted packet rate, the packet transmission delay (sender to receiver), the percentage of packets lost/corrupted, and the interarrival jitter (receiver to sender). This information can then be used to optimize the number and size of receiver buffers and to determine if the retransmission of lost packets is feasible. For example, if the transmission delay is below a defined threshold – for example if all the end systems are attached to the same LAN – it may be feasible to request the retransmission of corrupted packets. Conversely, if the delay is greater than the threshold then retransmissions are not possible.

Call setup As we shall expand upon in Chapter 8, LANs that do not provide a guaranteed QoS have no procedures to limit the number of calls/sessions that are using the LAN concurrently. Although this is acceptable with applications such as text-based file transfers which require only short bursts of bandwidth, with applications that involve audio and video, this approach is often not acceptable since the potentially large bandwidth that would be required to support many concurrent calls/sessions could exceed the total bandwidth available with the LAN. In order to limit the number of concurrent calls that involve multimedia, a device called an **H.323 gatekeeper** can (optionally) be used.

Essentially, during the setting up of a multimedia conferencing call, each end system involved in the conference must first obtain permission from the gatekeeper. Then, depending on the current level of usage of the LAN, the gatekeeper decides whether the call can take place. If an increase in the allocated bandwidth is required during a call, then again prior permission must be obtained from the gatekeeper. In addition, as we described in Section 1.5.4, since with a LAN the use of an MCU is optional, if an MCU is not being used, then the functions of the multipoint controller (MC) part of the MCU relating to the setting up of a multipoint call are often incorporated into the gatekeeper. The messages that are exchanged to set up a call are shown in Figure 5.7.

As we show in the figure, the setting up of a call is carried out in two stages. First the end system initiating the call obtains permission from the gatekeeper to set up a call by sending an *access request (ARQ)* message to the gatekeeper (1) and the gatekeeper responding with either an *access confirm (ACF)* or an *access reject (ARJ)* message (2). Assuming permission is received, for a two-party call the initiating terminal then sends a *setup* request message directly to the called end system (3). The latter first acknowledges receipt of the setup request by returning a *call proceeding* message directly to the initiating end system (4) and then proceeds to obtain permission from the

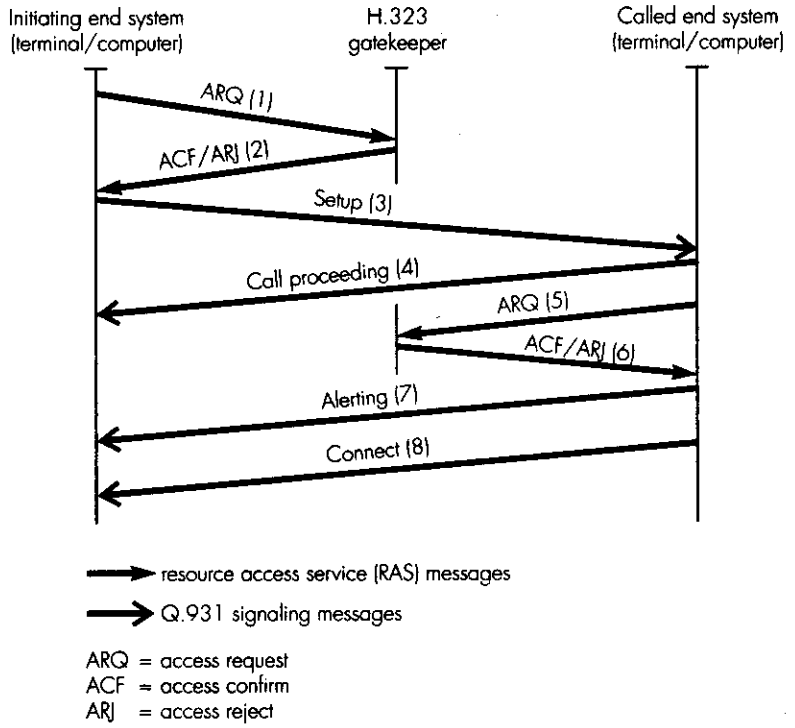


Figure 5.7 Two-party call setup procedure using an H.323 gatekeeper.

gatekeeper to take part in the call by means of the exchange of *ARQ* (5) and *ACF* (6) messages. Assuming permission is granted, the called end system sends an *alerting* message directly to the initiating end system (7) which is equivalent to the ringing tone heard when setting up a telephone call over a PSTN. Finally, if the user accepts the call, then the called end system returns a *connect* message directly to the initiating end system (8).

The messages exchanged with the gatekeeper concerned with the two end systems obtaining permission to set up a call are part of the **resource access service (RAS) protocol** and the example messages concerned with call setup which are exchanged directly by the two end systems are part of the Q.931 signaling protocol. In practice, both the RAS and Q.931 protocols are part of recommendation H.225. A similar procedure is followed for a multiparty conference call except all the messages are exchanged via the gatekeeper since this also contains the multipoint controller part of an MCU.

Once a call has been set up, the exchange of system control messages using the H.245 control protocol can then start. These include the negotiation of capabilities and the opening of logical channels for the audio, video, and user data streams. As we shall expand upon in Section 12.2, associated with both the UDP and the TCP protocols is a **port number**. This is carried in the header of the protocol data unit (PDU) associated with each protocol and is used to identify the application protocol to which the PDU contents relate. Also, as we shall expand upon in Section 12.2, in the header of the PDUs associated with the IP is the identity of the protocol (UDP or TCP) that created the packet to be transferred. Hence, as we show in Figure 5.8, on receipt of each packet from the network, the IP uses the **protocol identifier** to route the packet contents to either the UDP or the TCP. The latter then uses

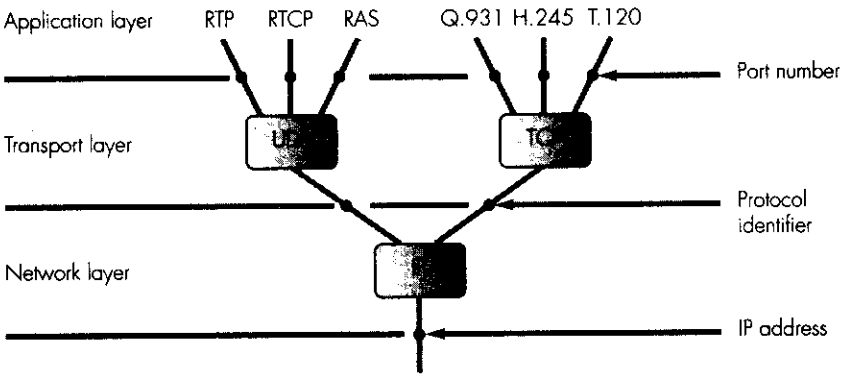


Figure 5.8 H.323 multiplexing/demultiplexing.

the port number at the head of the packet to relay the packet contents – the application protocol data unit – to the appropriate application protocol. Thus the multiplexing and demultiplexing operations associated with H.323 are carried out by the IP/UDP/TCP combination.

Interworking In addition to the operation of the end systems, the H.323 standard also defines how interworking with end systems that are attached to a circuit-mode network is achieved. This is through a device known as an **H.323 gateway** and the general scheme is shown in Figure 5.9.

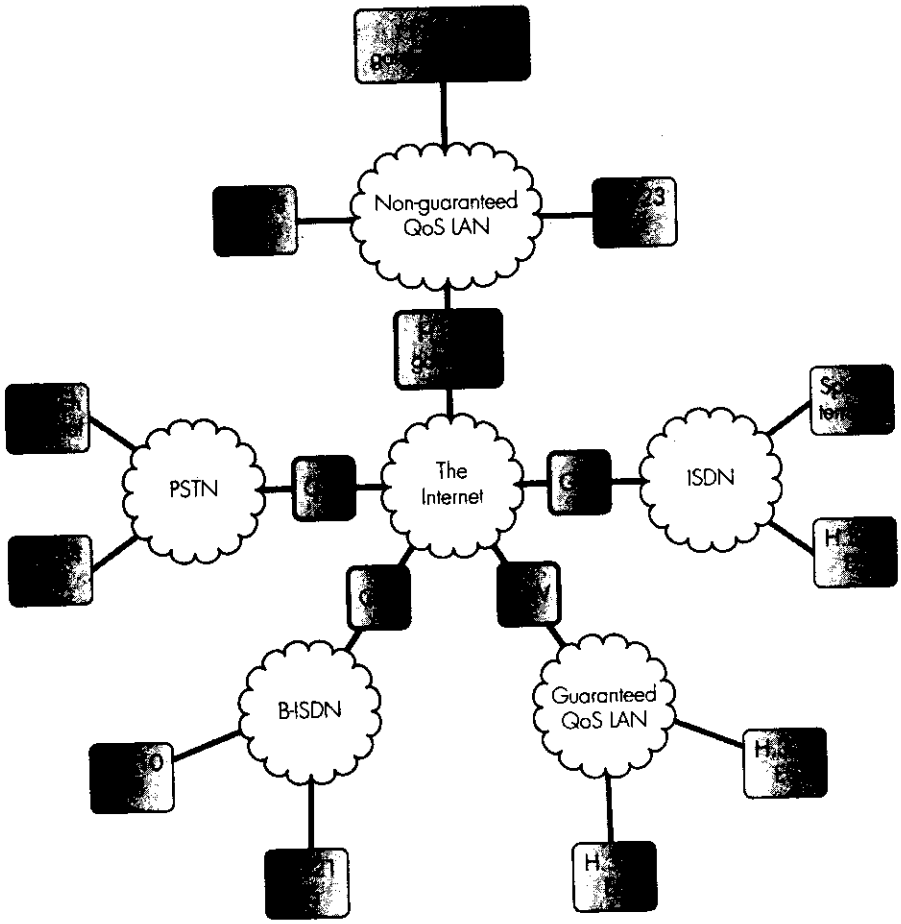


Figure 5.9 Interworking using an H.323 gateway.

The role of a gateway is to provide translations between the different procedures (and related control messages) associated with each network type. Hence translations are necessary for the procedures and messages associated with call setup and clearing (signaling), system control, and the two different multiplexing techniques. Also, if the two (or more) communicating end systems are using different audio and video codec standards, then transcoding between the two different coding techniques must be carried out. In order to minimize the amount of transcoding required in the gateway, the same audio and video codec standards are used whenever possible. Hence, as we saw earlier in Table 5.1, to make interworking easier, selected audio and video codec standards are made mandatory, which means that the end system must always be able to operate with these codecs. Thus in the case of interworking with an H.320-end system connected to an ISDN, both end systems should operate using G.711 audio and H.261/QCIF video. Similarly, in the case of interworking with an H.324-terminal connected to a PSTN, both terminals should operate using G.723.1 and either H.261 or H.263 video. If a common standard is not supported – determined during the capability exchange procedure – then the gateway must perform the appropriate transcoding procedures. However, as we show in Figure 5.9, a gateway can have multiple interfaces and, in practice, can support multiple calls simultaneously. Hence if transcoding is necessary, this restricts the number of simultaneous calls. Also, as with a gatekeeper, in some instances the gateway incorporates the MCU functions since this minimizes the amount of traffic on the LAN itself.

A second function associated with a gateway relates to address translation. This is necessary when interworking between end systems that are attached to different networks because each uses a different addressing scheme. For example, in the case of a LAN that uses the TCP/IP protocol set, the address of an end system is an IP address while with a PSTN or an ISDN the address is a conventional telephone number, all of which have a different format. Also, in the case of a LAN, end systems (computers in practice) are often referred to by a symbolic name rather than by their IP address. So to simplify interworking, all the end systems on a LAN are given an alias PSTN and/or ISDN number which can be used by a caller from outside that is connected to either a PSTN or an ISDN. Similarly, all the end systems that are external to the LAN can be allocated an alias IP address or symbolic name. The gatekeeper then performs the necessary translations between the different address types during the call setup procedure.

IETF

The early IETF standards relating to interpersonal communications over the Internet were concerned with providing a basic two-party telephony service between two IP hosts. Later, this was expanded to provide a more versatile facility supporting both multiparty conferencing and broadcast services. The membership of a conference or broadcast can vary as it progresses and the audio, video, and data involved can be integrated together in a dynamic way. To support these functions the IETF has developed a range of protocols which, in general, complement those associated with H.323.

In terms of the protocols used within a host/end system attached to the Internet, the main difference is the use of a different signaling protocol set from that used with H.323. These are the **session initiation protocol (SIP)** and the related **session description protocol (SDP)**. Essentially, these replace the RAS, call setup, and system control protocols shown in Figure 5.6.

SIP provides services for user location, call establishment, and call participation management. It is a simple request-response – also known as transaction – type of protocol and is defined in RFC 2543. The user of a host – which wants to set up a telephony call for example – sends a request message to the user of the called host which then responds by returning a suitable response message. Both the request and the response are made through an application program known as a user agent (UA) which maps the request and its response into the standard message format used by SIP.

On receipt of a request, the UA in the calling host formats a SIP message and this is transferred to the UA in the called host, normally using UDP. On receipt of the request, appropriate actions are initiated by the UA which then proceeds to map the response into the standard SIP format. The response is then returned to the UA in the client using the service provided by UDP.

Examples of SIP request messages – also known as *commands* or *methods* – are:

- *options*: this is sent to solicit the capabilities supported by a host;
- *invite*: this is sent to invite the user of a host to join in a call/session;
- *bye*: this is sent when the user of a host intends to leave a call/session.

Each message (PDU) consists of a header and a body. For example, the header of an *invite* request message contains fields such as *to* for the address of the called user and *from* for the address of the caller. Normally, the message body contains the individual media streams relating to the call.

An important feature of SIP is the location service it supports. Normally, users are identified by a symbolic name similar to an email address which, as we shall see in Section 14.2, is converted into an actual IP address and port number by a server called the **domain name server (DNS)**. When the required user host is attached to a different domain, then multiple servers are involved in performing the name-to-address resolution service. In addition, as with the H.323 standard, gateways are used to enable a user attached to the Internet to set up a call/session involving a user attached to another network such as a PSTN. An associated gateway location protocol (GLP) is then used to enable a SIP server to locate the gateway associated with a different network. We shall discuss the user location service of SIP and the operation of a SIP gateway in Section 14.6.1.

The SDP protocol is closely related to SIP. When a user is invited to join in a call/session, for example, the *invite* message includes SDP fields which define the individual media streams the caller can support, their format and the address and port numbers associated with each stream. Also, for broad-

cast calls/sessions – a lecture given over the Internet for example – the start and stop times of the lecture and contact details of the lecturer. The response contains a list of the media streams that the called party can support. All the fields relating to SDP are represented in a textual form. The contact details of the lecturer, for example, are symbolic names and IP addresses are in dotted-decimal. SDP, therefore, is a form of description language. Again, we shall discuss this further in Chapter 14, Section 14.6.2.

5.3.3 Electronic mail

From a user perspective, electronic mail (email) is probably the most popular interpersonal communications facility. As we explained earlier in Section 1.4.1, since an email message is delivered to the recipient's mailbox in a matter of seconds, email is much faster than postal mail and, since it does not require the recipient to be available to receive the message, it is often more convenient than a telephone call. Further advantages are that it is very straightforward to send the same message to groups of people and, with multimedia extensions, a mail message can contain various types of media including speech and video.

Internet mail

The two basic components associated with a text-based email system that uses the Internet – email clients and email servers – are shown in Figure 5.10.

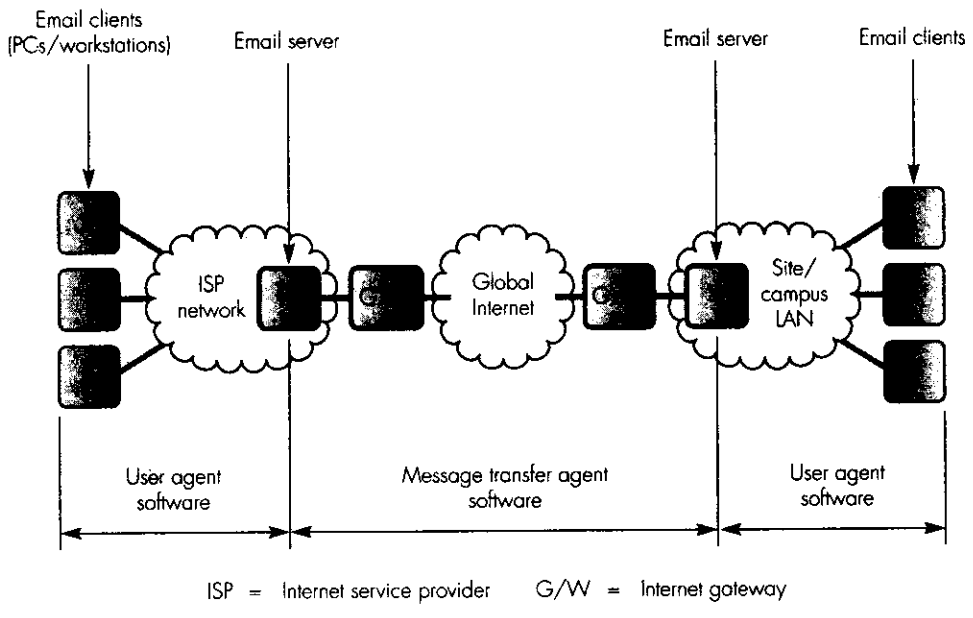


Figure 5.10 Email over the Internet.

Normally, an email client is a desktop PC which runs a program known as the **user agent (UA)**. As its name implies, this provides the user interface to the email system and acts as an agent to create new (mail) messages, initiate the sending of a message, read a received message from the user's mailbox, reply to a received message, forward a received message to another user, and to delete unwanted messages from the user's mailbox.

The email server is a server computer that maintains an IN and OUT mailbox for all the users/clients that are registered with it. The IN mailbox is used to store mail messages that have been received for the user and the OUT mailbox messages that have either been sent or are awaiting delivery. In addition, the server has software both to interact with the user agent software in each client and also to manage the transfer of mail messages over the Internet. The software associated with the latter function is known as the **message transfer agent (MTA)** and is concerned with the sending and receiving of messages to/from email servers that are also connected directly to the Internet over previously established TCP logical connections.

The protocol stack used to support email over the Internet is shown in Figure 5.11. Normally, the protocol stack associated with the access network – the internet service provider (ISP) network, and the site/campus LAN shown in the figure – is based on PC network protocols such as Novell Netware. A copy of the user agent software is run in each client (the UA client) and this communicates with a similar piece of software in the email server (the UA server) in order to log in to the server and to deposit and retrieve mail into/from the mailbox of the client. The set of mailboxes in the email server are contained in a database known as the **message store (MS)**. In addition, a copy of the user's (IN and OUT) mailbox is often held in the client machine. The UA client periodically retrieves any received mail from the message store and transfers this to its own local mailbox ready for reading by the user. An example protocol associated with the user agent function is the **post office protocol, version 3, (POP3)** which is defined in **RFC 1939**.

The standard structure of (ASCII) text-based mail associated with the Internet is defined in **RFC 822** and we showed an example of this earlier in Figure 1.9(b). Essentially, an email message comprises a header and a message body. The structure of the header is defined in **RFC 821**. It comprises several fields which include the email address of the sender and the intended recipient(s). The body part contains the actual message contents.

As we described in the previous section, the (reliable) transport protocol and the network protocol associated with the Internet are the TCP and IP respectively. Because the equivalent protocols associated with the various types of access network are different from these, the email server normally has two interfaces: one for communicating with the set of registered clients over the access network and the other for communicating with other email servers that are also connected directly to the Internet.

The application protocol associated with the transfer of messages between the MTA in two servers is the **simple mail transfer protocol (SMTP)**.

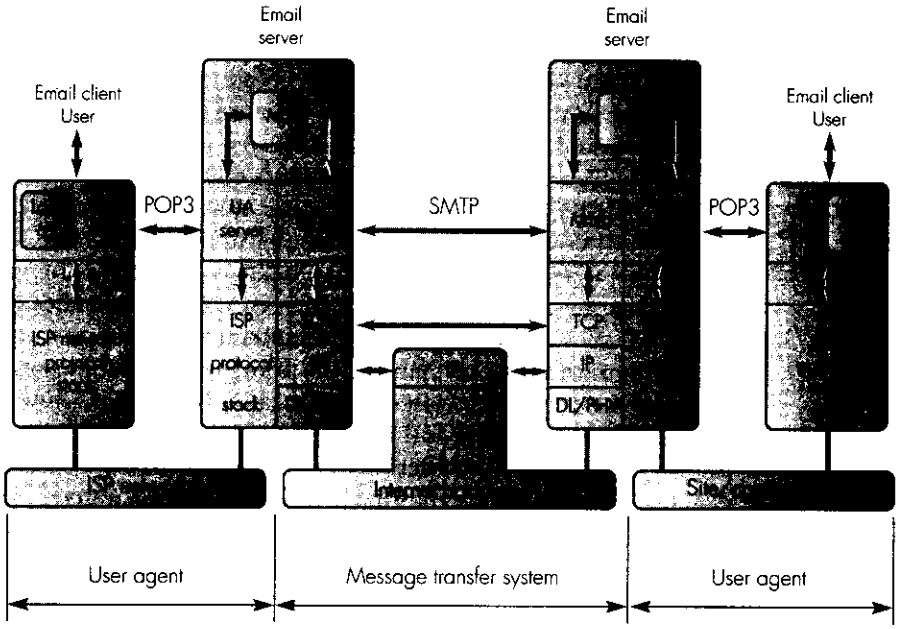


Figure 5.11 Protocol stack to support email over the Internet.

This is also defined in RFC 821 and we shall explain its operation in Section 14.3. Essentially, an email message is transferred by the MTA in the sending server by it first establishing a TCP connection to the MTA in the recipient server. The email address of both the sender and the intended recipient in the header of the message are in the form of symbolic names. Hence the sending MTA requests another application protocol known as the **domain name server (DNS)** for the related Internet address of the recipient server. It then uses this, together with its own Internet address, to create an Internet packet – also known as a **datagram**. As we shall expand upon in Section 9.6, the routing of packets over the Internet is carried out by the IP and hence the Internet address is known also as the **IP address**. Each Internet packet, therefore, contains the IP address of both the sending and recipient servers at its head and the email message as its contents. The IP address of the recipient server is used to route the packet over the Internet and, on receipt, the MTA in the server deposits the message contained in the packet into the recipient's mailbox.

MIME

The RFC 822 standard was defined for use with email messages that are written in English and are made up of just ASCII characters. As the Internet expanded, however, so the need to send messages in different languages evolved. And with the advent of multimedia PCs and workstations, so the need to send messages that contain other media types – audio, images, and video – arose. As a result, extensions to the basic format defined in RFC 822 were added. These are defined in **RFC 2045** and are known as **multipurpose Internet mail extensions** or **MIME**. By retaining the same basic format, existing mail programs and protocols can still be used.

In order to retain compatibility with RFC 822, with MIME additional encoding rules are given in the message header which define the type and structure of the message contents. Examples include image/JPEG and video/MPEG. It is also possible to define multipart messages with each part being of a different type and, if required, output in parallel. An example use of this is for a message which contains a video clip and an associated sound track which need to be output simultaneously. The first field in a MIME message header, however, is the *MIME-Version* and, if this is not present, then the message is assumed to consist of just ASCII text. More details about MIME are given in Section 14.3.2.

Email gateways

The structure we showed earlier in Figure 5.11 assumed that both email servers were connected to the Internet and hence could communicate directly using the SMTP/TCP/IP protocol stack. With many companies and large enterprises, however, often this is not the case and instead a different email system is used. Nevertheless, in addition to sending and receiving mail to/from other employees within the company/enterprise, many of these employees may also require to send and receive mail to/from people whose computers/PCs are connected either to a different company/enterprise network or to the Internet. In practice, there are two problems associated with doing this: first the format of the mail messages is often different and second the application protocols are also different. To overcome these problems, a device known as an **email gateway** is used, the general arrangement being shown in Figure 5.12.

As we show in the figure, the gateway has a number of interfaces, one for connecting to the local email server at the site and the others for connecting to those networks (including the Internet) with which the employees at the site wish to communicate. To transfer a message that is addressed to an outside network, the email server first transfers the message to a message buffer in the email gateway using the protocol stack associated with the company/enterprise network. The email address in the header of the message is then read by an application-level program to determine the network over which the mail should be forwarded. Assuming the external network is the Internet, the program proceeds to reformat the message into the RFC 822 format and then forwards this using the TCP/IP protocol stack as

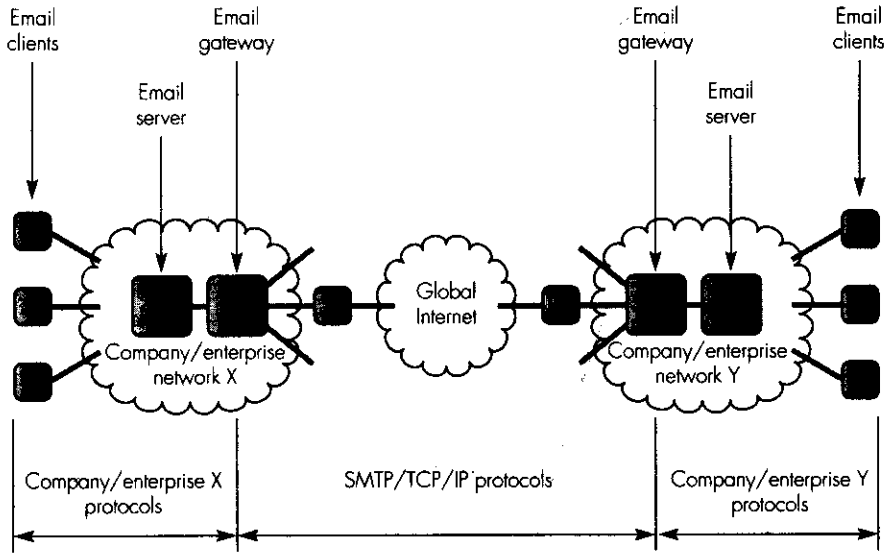


Figure 5.12 Email across dissimilar networks using an email gateway.

described previously. A similar procedure is followed in the reverse direction except the message format has to be changed from RFC 822 to the format used by the company network.

As we can see from the above, the translation procedure can be a time consuming process, especially when a number of different networks are involved and the message contents comprise more than just text. It is for this reason that, increasingly, companies are converting their networks to work using the SMTP/TCP/IP protocols. In the case of a large multisite company, the complete enterprise network is then known as an intranet. This has the advantage of not only removing the need for email gateways, but also enabling employees of the company to access and browse the World Wide Web directly and for selected servers – containing product literature for example – to be accessed from people outside the company.

5.4 Standards relating to interactive applications over the Internet

As we explained in Section 1.4.2, in the context of multimedia communications, most interactive applications over the Internet are concerned with interactions with a World Wide Web server. Hence in this section we shall identify a selection of the standards that have been defined for use with this type of interactive application. We shall identify and explain the role of these standards by considering various application scenarios.

5.4.1 Information browsing

The most basic type of interaction using the Web is for information browsing since with this, the Web user wishes only to browse through information that has been made available on a particular Web server at a site. Typically, as we explained in section 1.4.2, the information comprises an integrated set of one or more Web pages. Each page in the set contains linkages to other pages which can be located either on the same server or on any other server that is connected to the Internet. Typically, the Web pages are written in the HyperText Markup Language (HTML) and contain all the information necessary both to display the contents of a page – text, images, and so on – on the screen of the user/client machine and also to locate the other pages that have linkages with the page. The general arrangement used for information browsing is as shown in Figure 5.13.

A page is accessed and its contents displayed by means of a program known as a **browser** which runs in the user/client machine. The browser locates and fetches each requested page and, by interpreting the formatting commands that the page contains, the page contents are displayed. In addition, by the user clicking the mouse on a linkage point within the displayed page, the page that is linked to that point is accessed by the browser and displayed in the same way. There are a number of browser programs available, some popular examples being Netscape Navigator, NSCA Mosaic, and Microsoft Internet Explorer.

A page can contain two types of text – plaintext and underlined text (hypertext) – tables, images, and sometimes other media such as a sound track or a video clip. In the case of underlined text, in addition to the text,

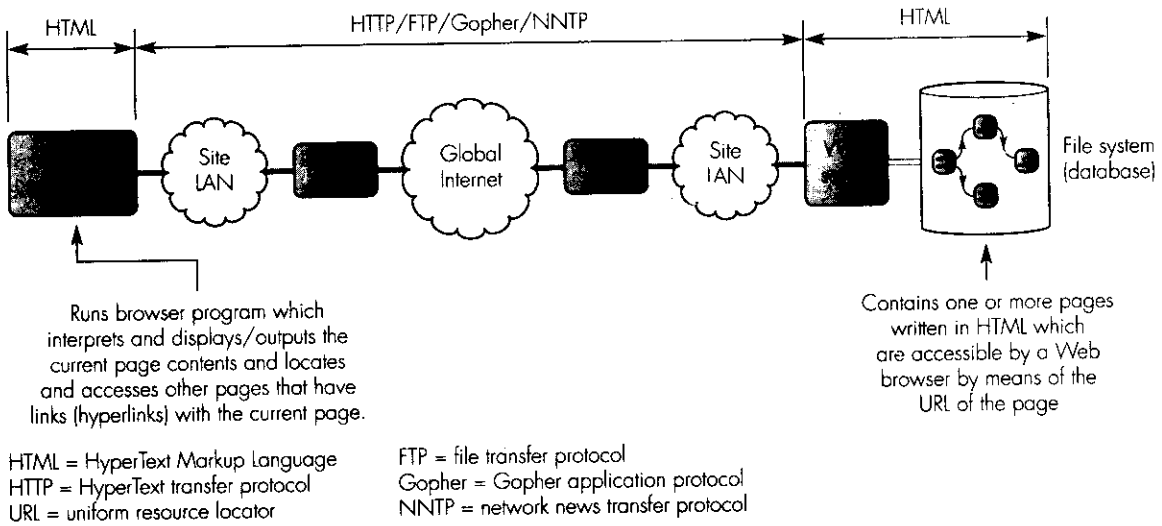


Figure 5.13 Information browsing.

this contains all the information that is necessary for the browser to access the contents of the linked page. This is known as a **hyperlink** and the linkage comprises the name of the application protocol (also known as the **scheme**) that is to be used – normally the **hypertext transfer protocol (HTTP)** – and the symbolic Internet name of the server machine (also known as the **domain**) in which the page is stored. Normally, this contains the top-level page for the site, which in turn contains hyperlinks to all the next-level pages. Alternatively, if a specific page is required, it is also possible to specify the (local) directory and name of the file that contains the required page. Collectively, these fields form what is called the **uniform resource locator (URL)** for the page. Two examples are:

http://www.microsoft.com
http://www.mpeg.org/index.html

Notice that all the characters can be either upper or lower case.

In the case of images (and other media types), the media type is in the form of a **tag (IMG)** with a parameter that indicates the name of the field where the image is stored. These are written in the page text at the point where the image is to be displayed and, when the browser interprets the tag, it reads the (compressed) image from the file. The file name also includes the image format and, by using a corresponding decompression algorithm, the browser displays the (decompressed) image at the appropriate point on the screen. Example formats include GIF and JPEG which we described in Sections 3.4.1 and 3.4.5 respectively. In the case of audio and video, these are output either in a similar way (if the browser contains the appropriate decompression code) or the contents of the file containing the audio/video are passed by the browser to a separate program for output known as a **helper application** or **external viewer**.

Each page is accessed and transferred over a TCP connection using the HTTP application-level protocol. Each HTTP interaction comprises a request from the browser written in the form of an ASCII string and a response from the server written in the RFC 822 format with MIME extensions, both of which we described earlier in Section 5.3.3. In order to allow for the possibility that the linked set of pages may be distributed over a number of different servers, a separate TCP connection is established between the client and server for each interaction and, once the response has been received, the TCP connection is cleared. HTTP is known, therefore, as a **stateless protocol** and we shall describe it in more detail in Section 15.2.2.

In addition to using HTTP to access pages written in HTML, a browser can also access other information using a number of the older application protocols. These allow access to the contents of:

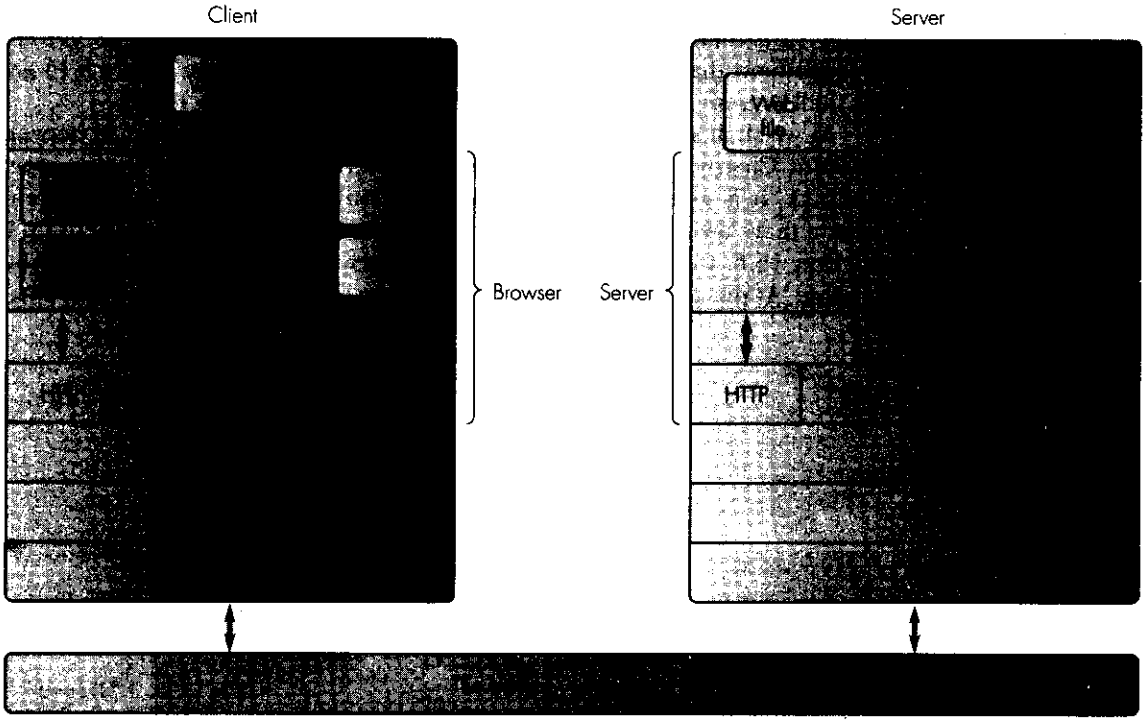
- a file on the client machine on which the browser is running,
- a remote file using the **file transfer protocol (FTP)**,

- a Gopher (text-only) file using the **gopher protocol**,
- a news article from a UseNet server using the **network news transfer protocol (NNTP)**.

A summary of the protocols that we have identified is presented in Figure 5.14. In the figure it is assumed that all pages are written in HTML and that the browser, in addition to HTTP, supports all the other application protocols we have just listed. Also, that it has various support facilities to decompress the contents of image, audio, and video files that may be included in a page.

5.4.2 Electronic commerce

When browsing the Web for information, the flow of information is unidirectional from the server to the client machine. As we explained in Section 1.4.2, however, some applications also involve the transfer of information in the



HTML = HyperText Markup Language
HTTP = HyperText transfer protocol
FTP = file transfer protocol
Gopher = Gopher application protocol
NNTP = network news transfer protocol

Figure 5.14 Protocol stack to support information browsing.

reverse direction from the client to a server; for example, after browsing the information at a site, to send details of your credit card in order to purchase, say, a book or theater ticket. This is just one example of what is known more generally as **electronic commerce** or **e-commerce** and there is a range of standards associated with this type of application.

In order to meet this requirement, it is possible to include what is known as a **form** into an HTML page. In the same way that a printed order form contains blank spaces for you to enter your name and other information and to make selections, so a typical HTML form is written to have a similar appearance. The user then uses the mouse and keyboard to enter the requested information and, when all the information has been entered and the appropriate selections made, typically, the user clicks on a symbolic **submit** button to initiate the sending of the entered information back to the server machine.

In addition to having a standardized way for a user of a client machine to enter and initiate the sending of information (forms/submit), there is also a standard for use at the server for processing the received information. This is known as the **common gateway interface (CGI)** and, in addition to accepting and processing the input from forms, the CGI may also initiate the output of other (unsolicited) pages that contain related information. The general arrangement that is used to support e-commerce is shown in Figure 5.15 and we shall present further details of forms and CGI in Section 15.3.6.

As we shall see, a second function associated with CGI is that of **network security**. Clearly, when information such as credit card details is sent over a network, it is essential that it is received only by the intended recipient. Hence there are standards for achieving this and, as we shall expand upon in Section 15.6, they are based on either a **private** or **public key encryption** scheme. At the application level, it is also necessary to authenticate that a par-

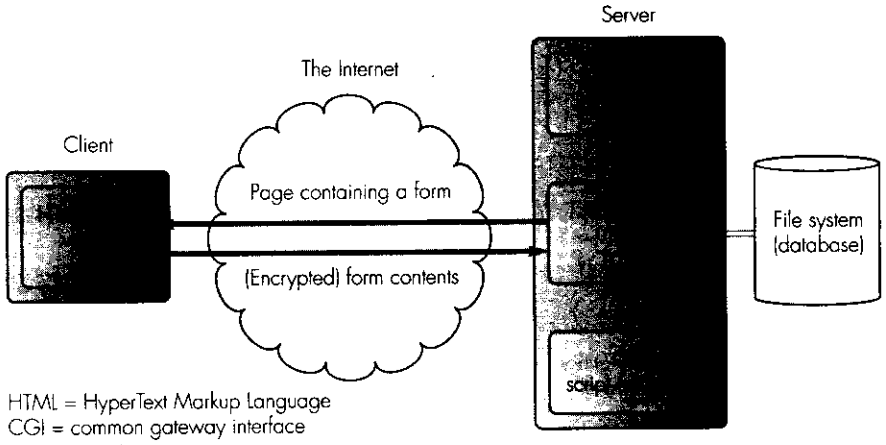


Figure 5.15 Electronic commerce.

ticular transaction was initiated by the owner of the credit card and not an impostor. Again there are associated standards which we shall also discuss in Section 15.6.

5.4.3 Intermediate systems

The discussion in the previous two sections assumed that both the client and server machines were connected directly to the Internet. In some instances, however, this is not the case and communication between the client and server is achieved through a networking device known as an **intermediate system**.

For example, as we explained in the last section, many enterprise networks now use the same set of protocols as are used with the Internet, the enterprise network then being known as an intranet. This is done both to simplify access to the Web from the various sites that make up the enterprise network and also to enable (external) Web users to access information that is stored on a server connected to the enterprise network. Normally, however, for security reasons access to a server that is connected to an intranet is not provided directly but rather through an intermediate system known as a **firewall** or **security gateway**.

As we show in Figure 5.16(a), the gateway controls the flow of information both to and from the intranet. To do this, the gateway intercepts all incoming requests from the Internet for access to the enterprise server and also all responses from the server that need to be forwarded over the Internet. Each Internet packet contains the IP address of both the source of the packet and the intended recipient/destination. Hence in the most basic type of gateway, the gateway simply maintains a separate list of all source and destination IP addresses that are allowed to pass both into and out from the intranet and any packets that have addresses different from these are discarded. This approach is known as **packet filtering** and is often used when the intranet itself comprises a large number of interconnected sites. In practice, however, it is not too difficult for a hacker to break this system and hence an alternative approach that performs the filtering operation at the application layer rather than the IP layer is also used. With this approach, the gateway behaves like the enterprise server to all incoming requests and only if the gateway is satisfied that a request is from a legitimate user is it relayed to the real server. Similarly, all responses from the real server are sent via the gateway. The same controls are applied to internal requests from a client connected to the intranet for an external server.

A second type of intermediate system is required when a browser supports only the HTTP application protocol. To access information that requires a different application protocol from HTTP it is necessary to use what is known as a **proxy server**. As we show in Figure 5.16(b), all requests for information are passed to the proxy server using the HTTP, but a proxy server can also communicate with other servers using application protocols

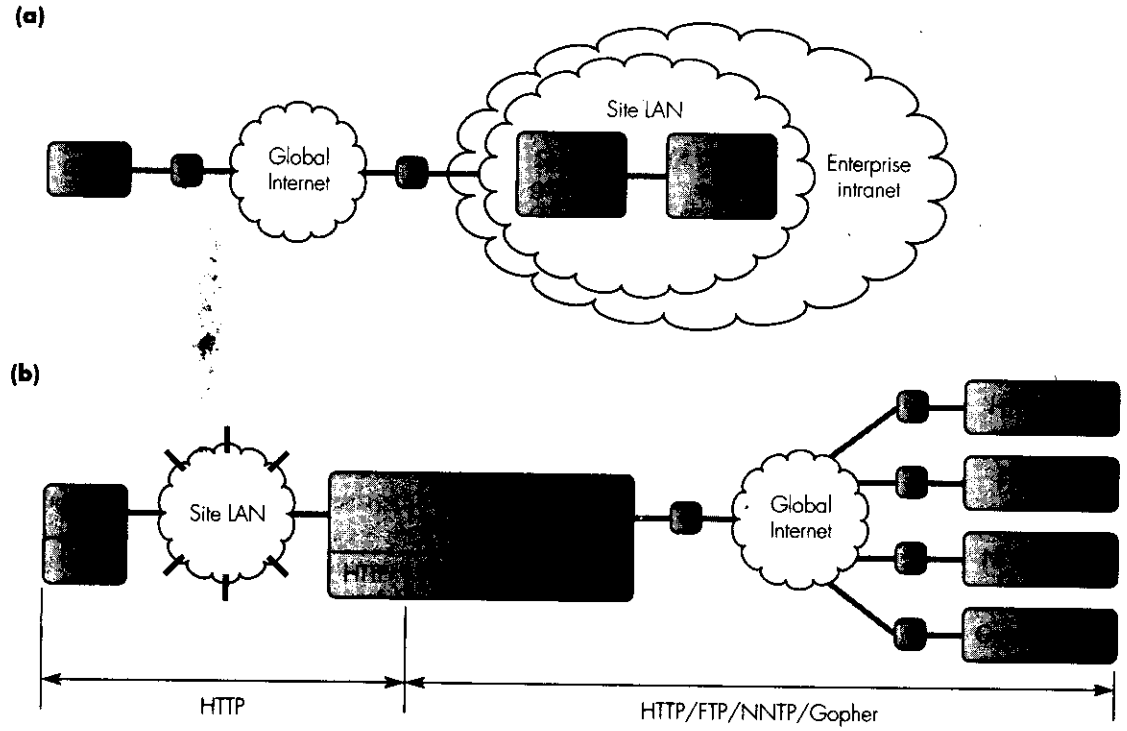


Figure 5.16 Intermediate systems: (a) security gateway; (b) proxy server.

such as FTP, NNTP, and Gopher. Hence if the information requested requires a different application protocol from HTTP, the proxy server makes the request on behalf of the client using the appropriate protocol. Similarly, on receipt of the requested information, this is passed to the client using HTTP. As we show in the figure, a proxy server can support a number of clients and, in some instances, performs other functions such as those associated with a security gateway.

5.4.4 Java and JavaScript

A disadvantage of using only HTML to write Web pages is that it is then relatively difficult to incorporate new features into pages – such as a new decoder – since this would necessitate modifications to the browser code. To overcome this constraint, it is possible to implement portions of the code for a page as self-contained subprograms – known as **applets** – which are independent of the HTML interpreter. Then, in the same way that an image is accessed and displayed when its tag is interpreted (by the HTML interpreter), so an applet is identified by a tag and, when this is interpreted, the applet code is loaded and run.

The advantage of using applets is that since each applet is a self-contained program, by implementing those parts of a page that contain code that is likely to change in the form of applets (for example media decoders), then any changes that do occur can be incorporated into the server rather than the browser. For example, if the browser contained only a particular type of image decoder and pages became available that contained images that were encoded using a different coder, then without applets the browser code would need to be modified. By using applets, however, the new decoder could be written as an applet – located either on the same server as the current page or on a different server – and, by simply specifying the applet with an applet tag within the page, so the applet for the new decoder would be loaded and run without any modifications to the browser itself. It is also possible to include applets for sound and video. The sound/video can then be output either when the applet is loaded or under control of the user at the click of the mouse.

An example of a programming language that is used to produce applets that are downloaded from a server is **Java**. This is based on C++ but, in order to obtain portability, there are no input/output statements associated with Java. A program written in Java is compiled to run on any machine. The compiled program is known as an applet and, in order for the applet to be run on a variety of different types of (client) machine, the applet code produced by the compiler is for what is called a **virtual machine**. The compiled/applet code is known as **bytecode** and, to run the applet, the browser, in addition to an HTML interpreter, must also contain an interpreter of the Java bytecode. The general scheme is shown in Figure 5.17.

In addition to downloading applets into pages written in HTML, it is also possible to implement a browser simply as a collection of applets. In this case, at startup, the browser comprises just a Java loader/interpreter and everything else is then implemented in the form of applets which are loaded on demand. Hence to access a page written in HTML, the HTML loader/interpreter would be loaded first and, if the accessed page contains an image, then the corresponding image decoder would be loaded and so on. Again, the advantage of this approach is that new versions of the various software components can be introduced more readily.

It is also possible to embed Java code into an HTML page directly. The language used to do this is called **JavaScript**. We shall return to the subject of Java and JavaScript in Section 15.5.

5.5 Standards for entertainment applications

We identified the two types of entertainment applications in Section 1.4.3 under the headings of movie/video-on-demand and interactive television. The standards relating to both types of application are concerned with the way the audio and video are integrated together prior to transmission – the transmission format – and the operational characteristics of the different

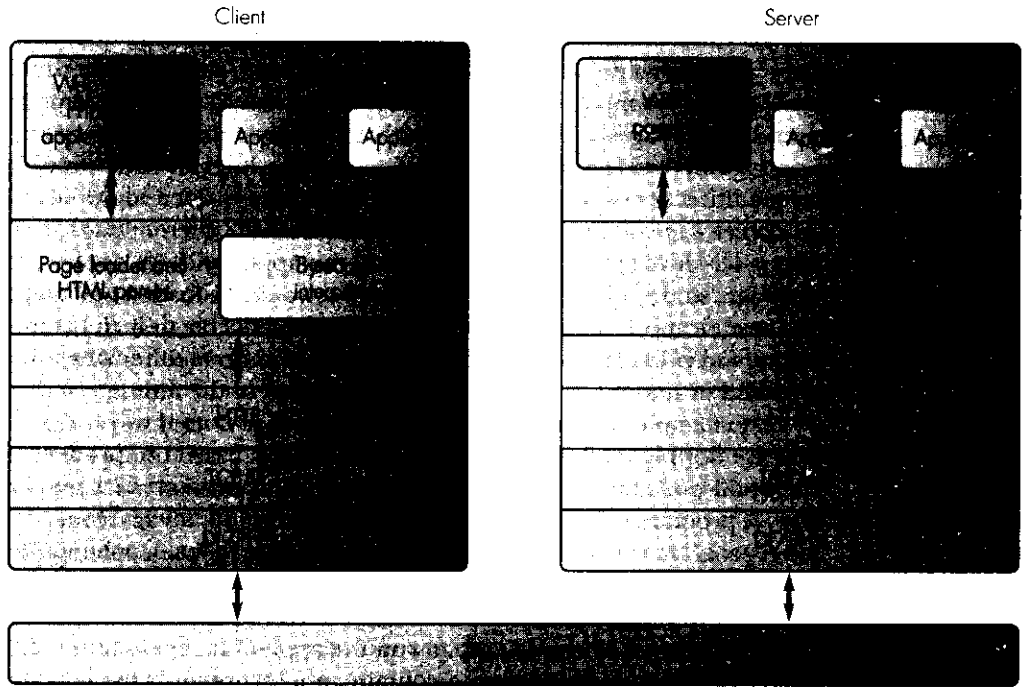


Figure 5.17 Protocol stack to support the browsing of pages containing Java applets.

types of distribution network that are used. We shall discuss the standards relating to both types of application separately.

5.5.1 Movie/video-on-demand

As we showed in Figure 1.15, movie/video-on-demand involves a movie/video being accessed from a server and transmitted over either the access network of a PSTN or a cable TV network. In both cases, the server is managed by the particular network operator and the subscriber interacts with the server to request a specific movie/video.

Transmission format

As we explained in Section 4.3.4, the standard relating to the storage of a VHS-quality video on a server is MPEG-1. The bit rate associated with MPEG-1 is 1.5 Mbps and this must support the video and audio streams associated with the movie/video.

The audio compression standards adopted for all the MPEG standards are defined in **ISO Recommendation 11172-3**. They are based on perceptual/subband coding which we described in Section 4.2.6. As you may recall, there are three alternative standards – known as layers – and typical

performance figures for each were given in Table 4.2. The figures shown in the table are for a single audio channel. Also, as we concluded at the end of Section 4.2.7, because the complexity and cost of the encoder – for a particular perceived quality – increases with the level of compression obtained, the choice of compression standard is a compromise between the bandwidth available and the desired perceived quality. As we indicated at the end of Example 4.2, the amount of bandwidth available for the audio associated with MPEG-1 is limited to 300 kbps. Hence a typical choice of audio is either layer-1 compression with single-channel joint-stereo at 256 kbps or layer-2 compression with dual-channel stereo at 128 kbps per channel. The format of the layer-1 audio stream was shown earlier in Figure 4.8(b).

We described the MPEG-1 video compression standard in Section 4.3.5 and the frame structure used for the video was shown in Figure 4.21. The digitization format used is the source intermediate format (SIF) with a resolution of either 352 × 240 pixels at 30 fps (NTSC) or 352 × 288 at 25 fps (PAL). The audio and video streams produced by the related encoders are then multiplexed together for storage on the server – a CD-ROM, DVD or hard disk – and subsequent transmission over the distribution network. The essential components that make up an MPEG-1 encoder and decoder are illustrated in Figure 5.18(a) and the format of the output bitstream from the encoder is given in part (b) of the figure

As we can see, in addition to the encoded video and audio streams, it is also possible to store related private (user) data. The output streams produced by both the video and audio encoders are known as **elementary streams (ESs)**. Prior to multiplexing the two streams together, however, since they are both encoded independently, timing information is added to both streams to enable the decoder to output both in synchronism. This is carried out by first dividing each (audio and video) ES into a sequence of discrete packets. Each is known as a **packetized ES** or **PES** and, at its head, is a *type* field that indicates whether the packet contents are audio, video, or private data. There is also a *time-stamp* which indicates the time the audio/video contained within the PES should be output. The time-stamps are generated using a 90 kHz **system time clock (STC)** which has a resolution of 33 bits. The header also indicates the amount of data – audio, video, or private – in the PES. Typically, the content of a PES is 2048 bytes.

The stream of PESs from both sources – and private data if present – are multiplexed together into a data structure known as a **pack**. At its head are two headers: a *pack header* and a *system header*. The first contains timing and output bit rate information. The timing information is a **system clock reference (SCR)** and is used by the decoder to synchronize its own system time clock at regular intervals to that of the encoder. The system header contains information such as the buffer size requirements and the number and type of elementary streams that are present.

On receipt of the pack bitstream, the demultiplexer passes the system clock reference to the local system time clock for synchronization and then routes each PES to the appropriate (audio/video) depacketizer. The latter

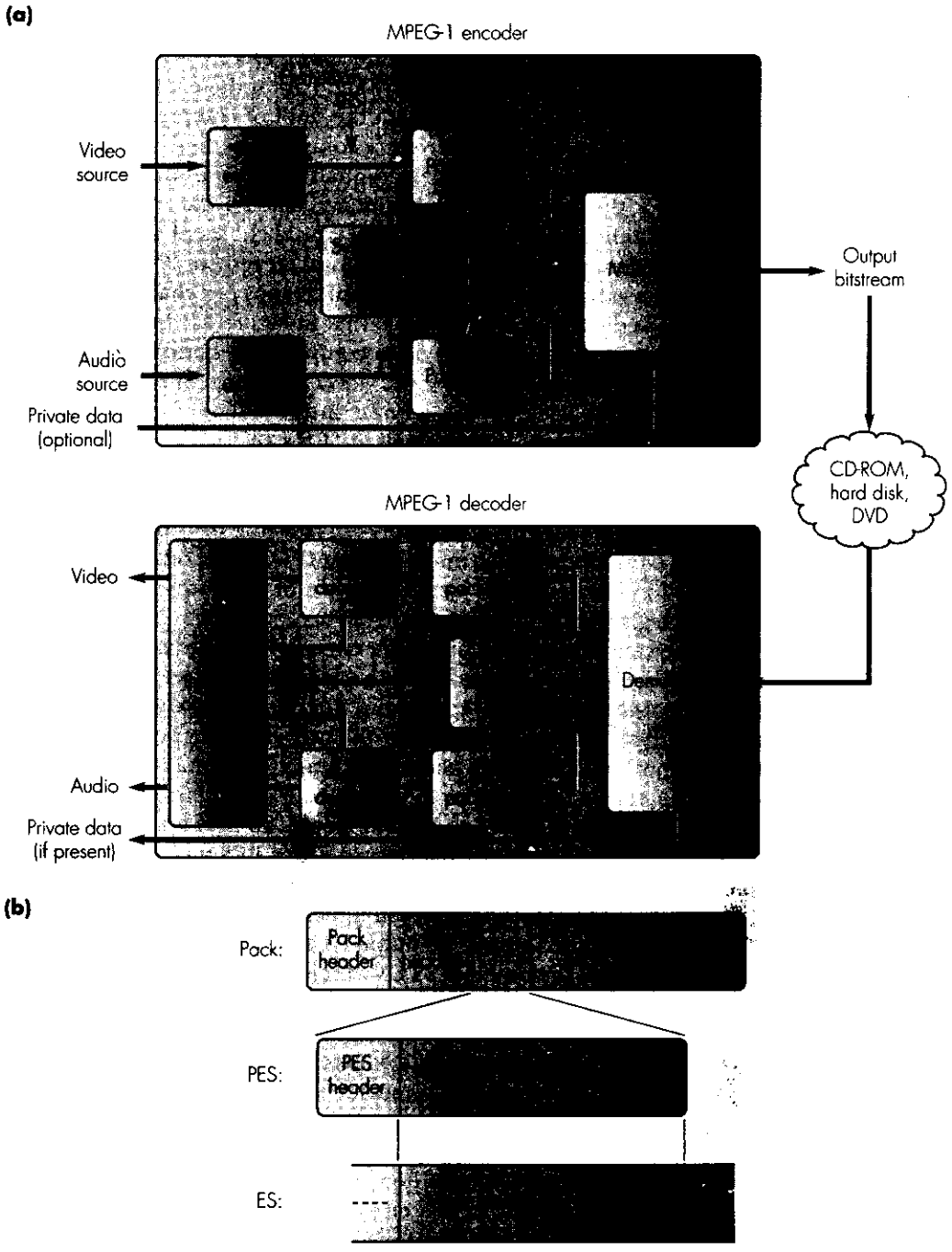


Figure 5.18 MPEG-1: (a) encoder/decoder; (b) output bitstream format.

then rebuilds the original ESs from the related PES contents and these are passed to the corresponding decoder. The timing information relating to each video/audio packet/frame is extracted from the ES and passed, together with the decompressed contents of the ES, to the **output interface controller**. The latter is responsible for outputting the synchronized audio and video in the desired (analog) format and hence includes digital-to-analog converters.

Distribution network

As we indicated at the beginning of this section, movie/video-on-demand to the home can be provided in one of two ways: over an existing twisted-pair line owned by a telephone company or a cable distribution network owned by a cable TV company. In addition, near movie/video-on-demand can be provided by a satellite company, normally by means of pay-per-view. In this case, however, a normal (higher bit rate) broadcast channel is used and hence we shall discuss this under the heading of interactive television.

As we shall see in Section 11.5, high bit rate modems for use with twisted-pair telephone lines provide a forward channel from the local exchange/switching office to subscriber premises of in excess of 1.5 Mbps together with a lower bit rate return channel for interaction purposes. Both these channels are in addition to the existing (analog) channel used for telephony and such modems are used not only to support movie/video-on-demand but also other interactive applications such as fast-access to the Internet.

The technology used in high bit rate modems is known as **asymmetric digital subscriber line** (**ADSL**) and we shall discuss it further in Section 11.5.1. Essentially, however, as we see in Figure 5.19(a), at the customer premises the conventional telephone socket outlet is replaced by an **ADSL subscriber unit** which, in addition to a telephone socket, provides a connection to a television set-top box. The subscriber is able to interact with the server and receive the requested movie/video over the forward channel. Typically, the technology supports a maximum length of twisted-pair cable of 2.5 miles/4 km.

With the telephone network a separate twisted-pair wire goes to each home and hence the facility is available to all subscribers. As we show in the figure, on the network side, the twisted-pair wires from multiple subscribers in an area are terminated in a piece of equipment called an **ADSL network termination unit**. This is located either in the local exchange/switching office or, more usually, in a kurb-side unit. The outputs from a number of such units are then multiplexed/demultiplexed together for transmission purposes back to the local exchange using optical fiber cable, the overall distribution network being known as **fiber-to-the-kurb** (**FTTK**). In addition, although much more expensive because of the need to lay new access cables, it is also possible to take the fiber cable directly to the home. The network is then known as **fiber-to-the-home** (**FTTH**) and clearly, this solution removes the necessity for having (ADSL) modems.

In the case of cable TV networks, a typical distribution network architecture is as shown in Figure 5.19(b). These support, in addition to the multiple TV channels broadcast from the cable TV company local office to all

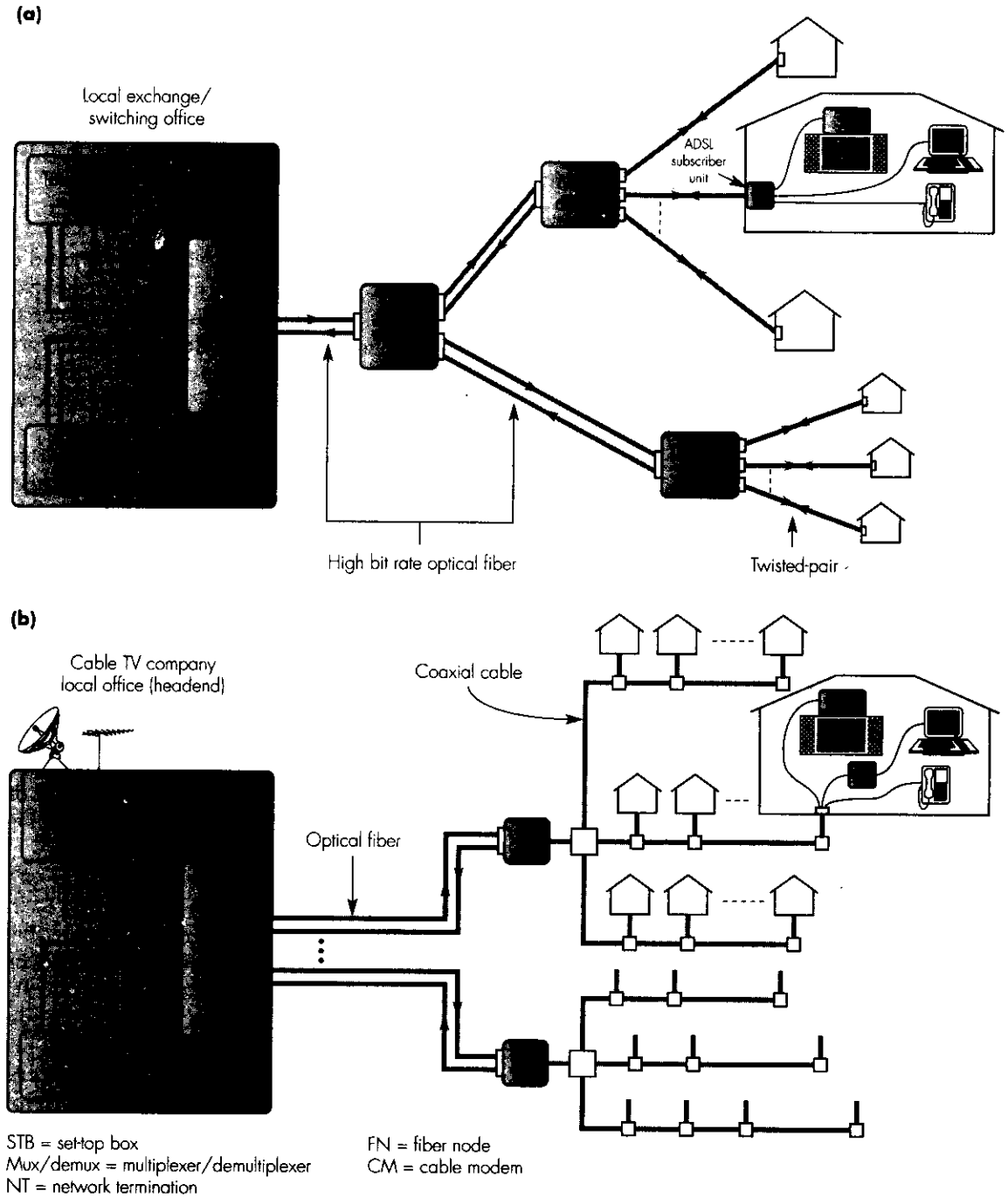


Figure 5.19 Movie/video-on-demand: (a) telephony company architecture; (b) cable TV company architecture.

subscribers, an additional duplex channel that is accessible by each subscriber. The unit used to provide this facility is known as a **cable modem** and this can be used to support a range of applications including movie/video-on-demand, fast Internet access, and so on.

As we show in the figure, the distribution network within a localized area consists of coaxial cable. This passes all the homes within that area and, in the case of broadcast TV programs, these are all multiplexed together onto the one cable and are accessible to all subscribers. Normally, however, each channel is encrypted so that only those registered subscribers with the appropriate decoder can decrypt and view the channels. In the case of the services supported by the cable modem, each subscriber can select the appropriate application separately. Also, because of the high bit rates involved, optical fiber is normally used to link each cable segment to the cable TV company local office. This type of distribution network is known as a **hybrid-fiber-coax** (**HFC**) network.

5.5.2 Interactive television

As we showed earlier in Figure 1.16, the digitized TV programs associated with interactive television can be provided by a cable, satellite, or terrestrial broadcast network. In the case of interactive television, however, the quality of the (broadcast) TV signal is better than that used for video-on-demand.

Transmission format

The video compression standard is based on MPEG-2 which we described earlier in Section 4.3.6. For screens with a 4/3 aspect ratio, the 4:2:0 digitization format is used with a resolution of either 720 × 480 pixels at 30 fps (NTSC), 720 × 576 at 25 fps (PAL) or 1440 × 1152 at 25 fps (European HDTV). For screens with a 16/9 aspect ratio, the same 4:2:0 digitization format is used but with a resolution of 1280 × 720 pixels (ATV/GA).

The audio compression standard is either MPEG layer 2 (PAL) or Dolby AC-3 (NTSC and ATV/GA). In both cases, up to five full-bandwidth channels – left, right, center, and two surround-sound channels – can be present and also one or more lower-bandwidth channels for commentaries and voice-overs such as translations.

The format of both the audio and video elementary and packetized streams and the encoding of a single television program are similar to those used for MPEG-1. However, since in broadcast applications multiple TV programs are multiplexed together for transmission over the distribution network, an additional format is defined for this purpose. The output stream produced by a single program encoder is known as a **program stream** (**PS**) and the output stream containing multiple programs the **transport stream** (**TS**). The general arrangement is shown in Figure 5.20.

As we can see in Figure 5.20(a), if the digitized TV program is to be stored – rather than broadcast – then the program stream is used directly. If it is to be multiplexed with other programs, then the output streams from the set of program encoders are fed to a second multiplexer known as the **transport stream multiplexer** and it is the output of this that is fed to either a

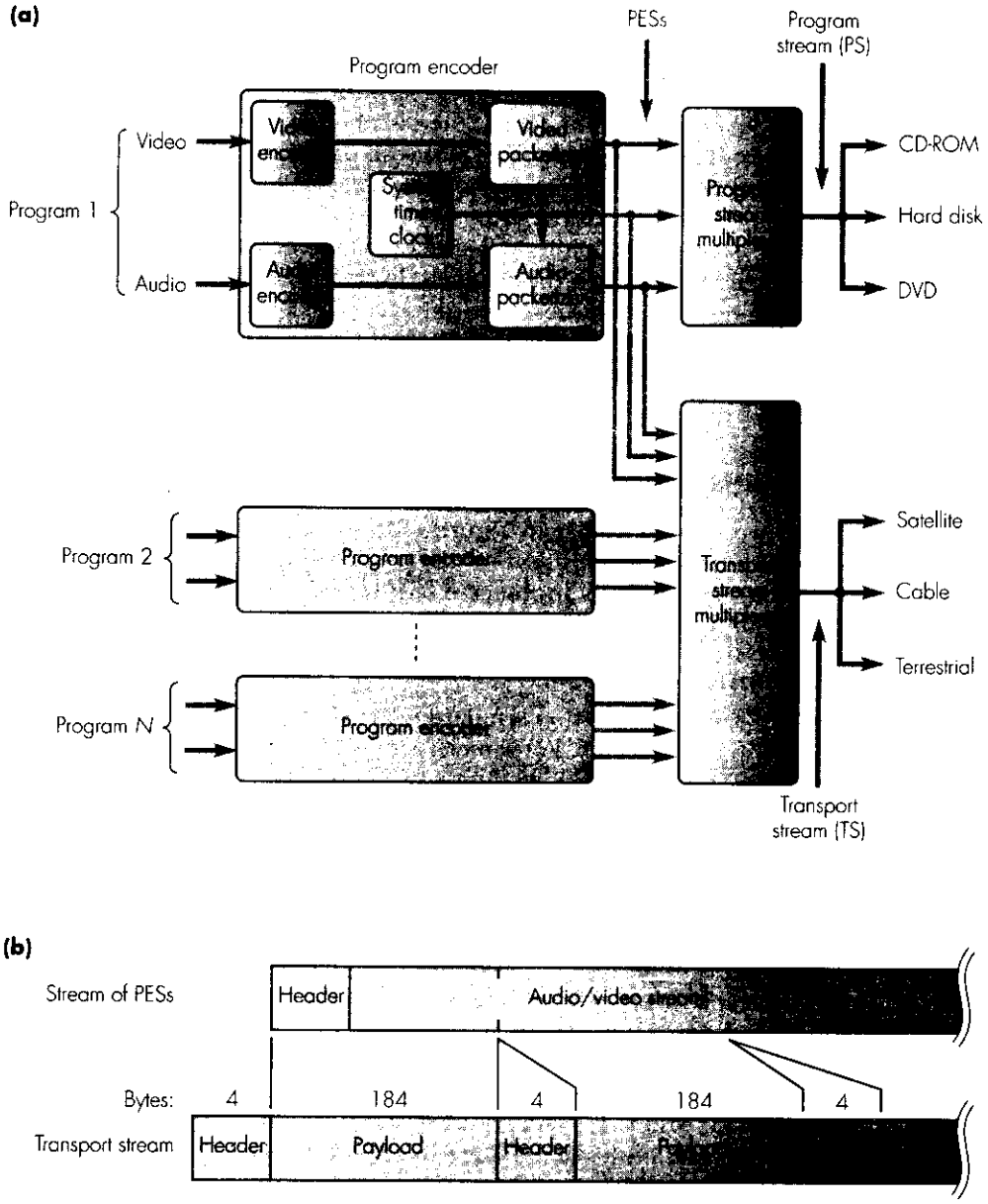


Figure 5.20 TV program multiplexing: (a) PS and TS generation; (b) TS format.

satellite, cable, or terrestrial transmitter. The format of this stream is shown in part (b) of the figure and, as we can see, is divided into a string of 188-byte packets, each comprising a 44-byte header and a 184-byte contents field known as the **payload.**

The header contains a number of fields which include a *synchronization byte*, to enable the receiver(s) to interpret the packet header and its contents on the correct byte boundaries, a *packet identifier* (*PID*), to enable the receiver to relate the TS packet payload to the correct PES during its reassembly, an *adaptation flag* (bit) and a *payload flag* (bit). Typically, the length of the PES output by the video and audio packetizers is 2048 bytes and hence these must be fragmented/segmented into multiple 184-byte segments for transmission. Since this does not divide equally into 2048, then the payload of the last TS packet may not be full. If this is the case, then the adaptation flag in the header of this packet is set and the first byte in the payload field indicates the number of bytes in what is known as the *adaptation field*. This comprises, in addition to *stuffing bytes* to fill the payload, a time-stamp indicating when the TS packet was created. This is known as the *program clock reference* and has the same role as the SCR used in an MPEG-1 output stream. Because of the presence of the adaptation field, the payload may not contain any PES data and, if this is the case, the payload flag in the header is set to inform the receiver of this.

In addition to fragments of PES packets, the payload of TS packets is also used to carry system-level tables. These are the **program allocation table** (**PAT**), the **program map table** (**PMT**), the **conditional access table** (**CAT**), and also one or more **private tables**. The presence of a system-level table – instead of a PES fragment – is indicated by a PID of zero in the TS packet header. Since a possibly large number of different program streams – each comprising multiple PES packets – can be present in the transport stream, collectively the system tables are used to inform the receiver about the program streams that are currently present. For example, the PAT contains a field that defines the link between the PID carried in each TS packet header and the corresponding television program. Similarly, the PMT indicates the PID of the elementary streams that make up each program. And, if the program has conditional access – for example, pay-per-view – it indicates the code required to unscramble the related program TS packet contents. A CAT is sent to the receiver whenever a program is present in the TS which has conditional access. The CAT contains the necessary access control data used by the set-top box to allow or inhibit access to the program. Finally, the private tables are intended for service providers and include tables such as a program guide (to allow a subscriber to see the programs currently available), the set of channel frequencies that are being used for the current set of programs, and time-and-date information for the set-top box.

Summary

In this chapter we have identified and presented an overview of the standards that have been defined for use with multimedia communications. We showed that these are many and varied and relate both to the operation of the various networks that are used and also to the hardware and software in the computers (and other types of terminal equipment) that are attached to these

networks to provide their users with access to multimedia communication services. A summary of the specific multimedia applications that we described is given in Figure 5.21.

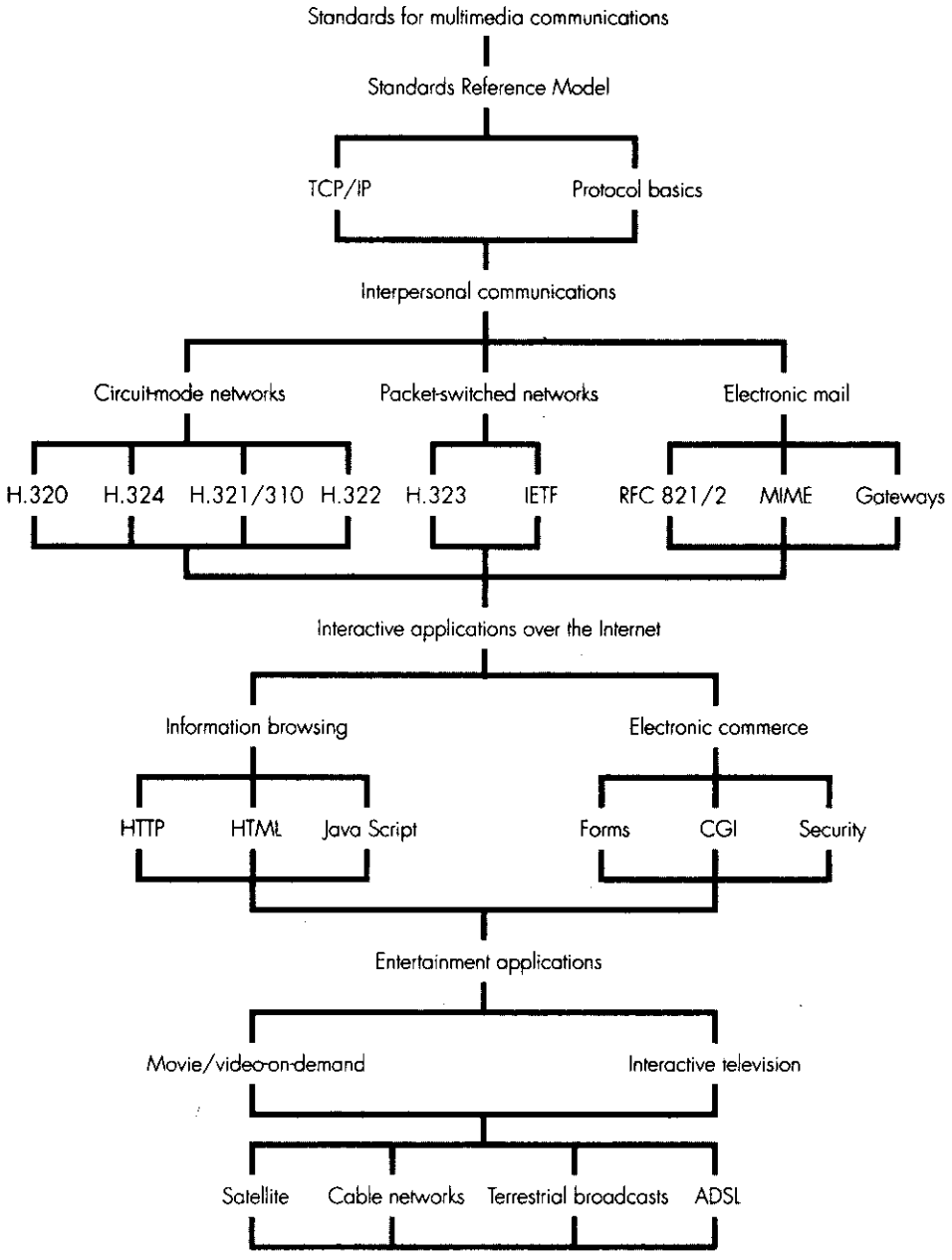


Figure 5.21 Summary of standards identified/discussed in Chapter 5.

Exercises

Section 5.2

5.1 List the reasons why standards are necessary for networked applications involving dissimilar computer/end systems.

5.2 With the aid of the diagrams shown in Figure 5.1, explain the meaning of the following terms relating to standards for multimedia applications:
(i) application standards,
(ii) network interface standards,
(iii) internal network standards,
(iv) networking environment,
(v) local significance,
(vi) end-to-end significance,
(vii) application environment.

5.3 With the aid of the diagrams shown in Figure 5.2, describe the role of each of the five layers that make up the TCP/IP reference model.

5.4 With the aid of a diagram, explain the meaning and role of the following:
(i) peer layer communication,
(ii) protocol data unit (PDU),
(iii) protocol control information,
(iv) outgoing stream construction,
(v) incoming stream reduction.

Section 5.3

5.5 In relation to the set of standards for circuit-mode networks shown in Figure 5.4, explain the role of the following component parts:
(i) call setup,
(ii) multiplexer/demultiplexer,
(iii) system control,
(iv) MCS/MCU,
(v) receive path delay.

5.6 In relation to the set of standards that are used with the different types of circuit-mode network identified in Table 5.1, discuss the following issues:
(i) the choice of audio codec,
(ii) the choice of video codec,
(iii) the role of the user data application protocols,
(iv) the role of the system control and call setup protocols.

5.7 With the aid of the diagrams shown in Figure 5.5, explain the operation of the H.223 multiplexer as used with a PSTN. Include in your explanation:
(i) the structure and content of the information field,
(ii) the structure and role of the multiplex table.

5.8 In relation to the set of standards relating to H.323 shown in Figure 5.6, explain the following:
(i) the role of the gatekeeper during the setting up of a call,
(ii) the role of the gateway during interworking with end systems attached to a circuit-mode network.

5.9 In relation to the set of standards defined by the IETF for interpersonal communications over the Internet, explain the roles of the following:
(i) the session initiation protocol and the related session description protocol,
(ii) the gateway location protocol.

5.10 List the advantages of email over postal mail.

5.11 With the aid of the schematic diagrams shown in Figures 5.10 and 5.11, explain the role of the following when sending an email message:
(i) the user agent,
(ii) an email server including the message store,
(iii) the POP3 protocol,
(iv) the message transfer agent and the SMTP,
(v) the domain name server,
(vi) MIME.

5.12 With the aid of the schematic diagram shown in Figure 5.12, explain how email is sent across the Internet via an email gateway.

Section 5.4

5.13 With the aid of the schematic diagrams shown in Figures 5.13 and 5.14, explain the role of the following when browsing the Web:
(i) the browser,
(ii) HTML,
(iii) URL,
(iv) HTTP,
(v) helper application,
(vi) FTP, Gopher, and NNTP.

5.14 With the aid of a diagram, identify and explain the role of the following relating to e-commerce over the Internet:
(i) forms,
(ii) submit button,
(iii) CGI scripts,
(iv) encryption.

5.15 With the aid of the schematic diagrams shown in Figure 5.16 (a) and (b), explain the role of the following types of intermediate system:
(i) a security gateway,
(ii) a proxy server,
(iii) a cache server.

5.16 With the aid of Figure 5.17, explain how a Java applet can be downloaded and run in a browser. Include in your explanation the meaning and use of:
(i) bytecode,
(ii) bytecode interpreter.

Section 5.5

5.17 With the aid of the schematic diagrams shown in Figure 5.18, explain the use/meaning of the following relating to movie/video-on-demand:
(i) elementary stream,
(ii) packetized ES,
(iii) system time clock,
(iv) pack,
(v) pack header,
(vi) system clock reference,
(vii) output interface controller.

5.18 With the aid of a diagram, describe the essential parts of a twisted-pair access network that has been upgraded to support movie/video-on-demand and high-speed access to the Internet. Include in your description the meaning and use of:
(i) ADSL,
(ii) ADSL subscriber unit,
(iii) ADSL network termination unit.

5.19 With the aid of a diagram, describe the essential parts of a cable distribution network that has been upgraded to support digital TV and other high-speed services. Include in your description the meaning and use of:
(i) hybrid-fiber-coax network,
(ii) fiber node,
(iii) cable modem.

5.20 With the aid of Figure 5.20(a), explain the use/meaning of the following relating to interactive television:
(i) program stream,
(ii) transport stream multiplexer
(iii) transport stream format,
(iv) synchronization byte,
(v) program clock reference,
(vi) system-level tables.