CHAPTER 9

# ATM Networks

In Chapter 7 we saw that asynchronous transfer mode (ATM) was developed to combine the attributes of time-division circuit-switched networks and packet-switched networks. We also saw how ATM provides the capability of providing Quality-of-Service support in a connection-oriented packet network. In this chapter we present the details of ATM network architecture.

The chapter is organized as follows:

1. *Why ATM?* We first provide a historical context and explain the motivation for the development of ATM networks.
2. *BISDN reference model*. We examine the BISDN reference model that forms the basis for ATM, and we explain the role of the user and control planes.
3. *ATM layer*. We examine the ‘‘network layer’’ of the ATM architecture, and we explain the operation of the ATM protocol. Quality of service (QoS) and the ATM network service categories and the associated traffic management mechanisms are introduced.
4. *ATM adaptation layer*. We introduce the various types of ATM adaptation layer protocols that have been developed to support applications over ATM connections.
5. *ATM signaling*. We provide an introduction to ATM addressing and to ATM signaling standards.
5. *PNNI routing*. We briefly describe a dynamic routing protocol for ATM networks, called PNNI.

In the next chapter we show how IP and ATM networks can be made to work together. We also show how IP networks are evolving to provide QoS support.

## 9.1 WHY ATM?

The concept of ATM networks emerged from standardization activities directed at the development of *Integrated Services Digital Networks (ISDNs)*. In the 1970s the trend toward an all-digital (circuit switched) telephone network was clearly established and the need to (eventually) extend digital connectivity to the end user was recognized. It was also apparent that data applications (e.g., computer communications and facsimile) and other nonvoice applications (e.g., videoconferencing) would need to be accommodated by future networks. It was also clear that circuit switching would not be suitable for bursty data traffic and that packet switching would have to be provided. The ISDN standards were the first effort at addressing these needs.

The recommendations adopted by the CCITT (now the telecommunications branch of the International Telecommunications Union) in 1984 defined an *ISDN* as a network that provides *end-to-end digital connectivity* to support a *wide range of services* to users through a limited set of *standard user-network interfaces*. The basic rate interface consisted of two constant bit rate 64 kbps B channels and a 16 kbps D channel. The primary rate interface provided for either 23 B channels and a 64 kbps channel, primarily in North America, or for 30 B channels and a 64 kbps channel elsewhere. The ISDN recommendations provided for the establishment of voice and data connections. The recommendations focused exclusively on the interface between the user and the network and did not address the internal organization of the network. Indeed once inside the network, voice and data traffic would typically be directed to separate circuit-switched and packet-switched networks. Thus the network operator was still burdened with the complex task of operating multiple dissimilar networks.

It soon became clear that higher rate interfaces would be required to handle applications such as the interconnection of high-speed local area network (LANs) as well as to transfer high-quality digital television. The initial discussions on *broadband ISDN* (BISDN) focused on defining additional interfaces along the lines of established rates in the telephone digital multiplexing hierarchy. However, eventually the discussions led to a radically different approach, known as ATM, that attempts to handle steady stream traffic, bursty data traffic, and everything in between. ATM involves converting all traffic that flows in the network into 53-byte blocks called *cells*. As shown in Figure 9.1, each cell has 48 bytes of *payload* and a 5-byte *header* that allows the network to forward each cell to its destination.

The connection-oriented cell-switching and multiplexing principles underlying ATM were already discussed in Chapter 7. Here we reiterate the anticipated advantages of ATM networks.

1. The network infrastructure and its management is simplified by using a single transfer mode for the network; indeed, extensive bandwidth management capabilities have been built into the ATM architecture.

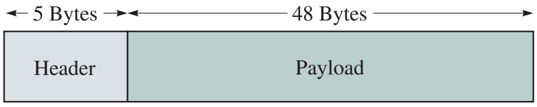


**FIGURE 9.1** The ATM cell

2. Unlike shared media networks, ATM is not limited by speed or distance; the switched nature of ATM allows it to operate over LANs as well as global backbone networks at speeds ranging from a few Mbps to several Gbps.
3. The QoS attributes of ATM allow it to carry voice, data, and video, thus making ATM suitable for an integrated services network.

The ATM standardization process has taken place under the auspices of the ITU-T in concert with national and regional bodies such as ANSI in the United States and ETSI in Europe. The development of industry implementation agreements has been mainly driven by the ATM Forum.

## 9.2 BISDN REFERENCE MODEL

The BISDN reference model is shown in Figure 9.2. The model contains three planes: the user plane, the control plane, and the management plane.The *user plane* is concerned with the transfer of user data including flow control and error recovery. The *control plane* deals with the signaling required to set up, manage, and release connections. The *management plane* is split into a layer management plane that is concerned with the management of network resources and a plane

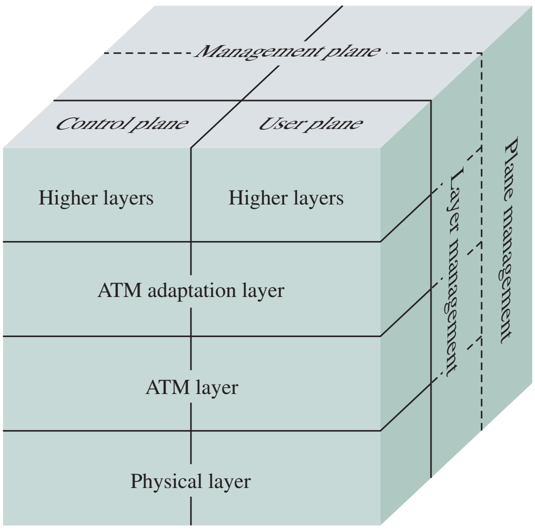


**FIGURE 9.2** The broadband ISDN reference model

management plane that deals with the coordination of the other planes. We focus on the user and control planes.

The user plane has three basic layers that together provide support for user applications: the ATM adaptation layer, the ATM layer, and the physical layer. The **ATM adaptation layer (AAL)** is responsible for providing different applications with the appropriate support, much as the transport layer does in the OSI reference model. Several AAL types have been defined for different classes of user traffic. The AAL is also responsible for the conversion of the higher-layer service data units (SDUs) into 48-byte blocks that can be carried inside ATM cells. Figure 9.3 shows how the information generated by voice, data, and video applications are taken by AALs and converted into sequences of cells that can be transported by the ATM network. The AAL entity on the receiver side is responsible for reassembling and delivering the information in a manner that is consistent with the requirements of the given application. Note that the AAL entities reside in the terminal equipment, and hence they communicate on an end-to-end basis across the ATM network as shown in Figure 9.4.

The **ATM layer** is concerned solely with the sequenced transfer of ATM cells in connections set up across the network. The ATM layer accepts 48-byte blocks of information from the AAL and adds a 5-byte header to form the ATM cell. The header contains a label that identifies the connection and that is used by a switch to determine the next hop in the path as well as the type of priority/scheduling that the cell is to receive.

ATM can provide different QoS to different connections. This requires that a *service contract* be negotiated between the user and the network when the connection is set up. The user is required to describe its traffic and the required QoS when it requests a connection. If the network accepts the request, a contract is

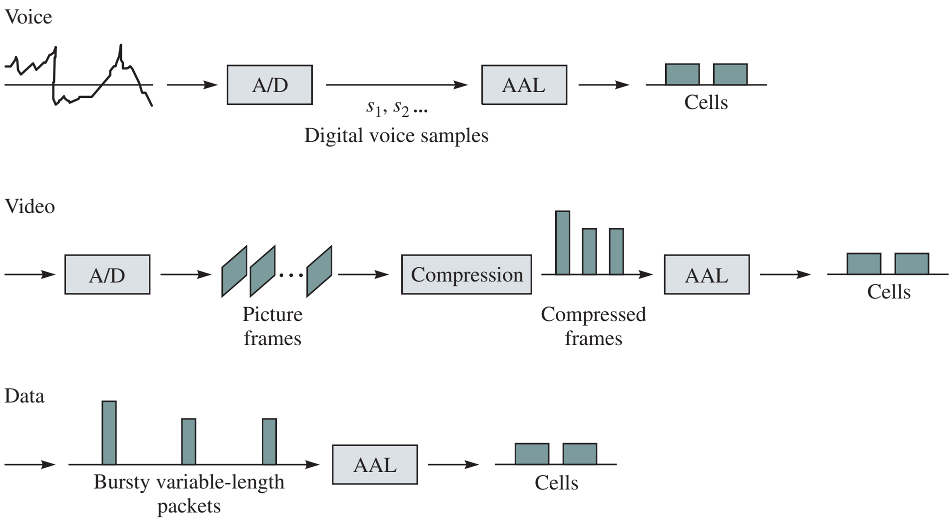


**FIGURE 9.3** The AAL converts user information into cells

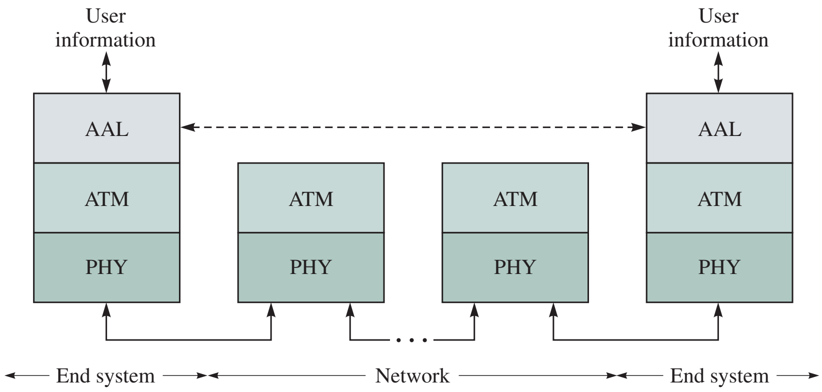


**FIGURE 9.4** User plane layers

established that guarantees the QoS as long as the user complies with its traffic description. Queue priority and scheduling mechanisms implemented in ATM switches provide the capability of delivering QoS. To deliver on its QoS commitments, the ATM network uses policing mechanisms to monitor user compliance with the connection contract and may discard cells not found in compliance.

In terms of number of users involved, ATM supports two types of connections: point to point and point to multipoint. Point-to-point connections can be unidirectional or bidirectional. In the latter case different QoS requirements can be negotiated for each direction. Point-to-multipoint connections are always unidirectional.

In terms of duration, ATM provides permanent virtual connections (PVCs) and switched virtual connections (SVCs). PVCs act as "permanent" leased lines between user sites. PVCs are typically provisioned "manually" by an operator. SVCs are set up and released on demand by the end user.

SVCs are set up through signaling procedures. Initially the source user must interact with the network through a **user-network interface (UNI)** (see Figure 9.5). The connection request must propagate across the network and eventually involve an interaction at the destination UNI. Within a network, switches must interact across the **network-network interface (NNI)** to exchange information. Switches that belong to different public networks communicate across a **broadband intercarrier interface (B-ICI)**. The source and destination end systems as well as all switches along the path across the network are eventually involved in the allocation of resources to meet the QoS requirements of a connection.

Signaling can be viewed as an application in which end systems and switches exchange higher-level messages that establish connections across the network. The role of the **control plane** is to support signaling and network control applications. The control plane has the same three basic layers as the user plane. A *signaling AAL* has been defined for the control plane to provide for the reliable exchange of messages between ATM systems. Higher-layer protocols have been defined for use over the UNI, for the NNI, and for the B-ICI.

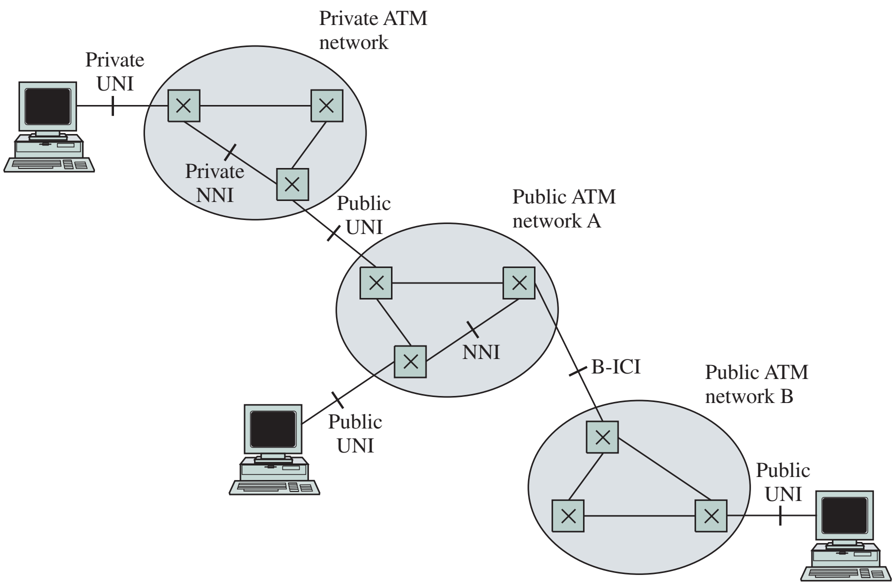


**FIGURE 9.5** ATM network interfaces

The *physical layer* is divided into two sublayers as shown in Figure 9.6. The *physical medium dependent sublayer* is the lower of the two layers and is concerned with details of the transmission of bits over the specific medium, such as line coding, timing recovery, pulse shape, as well as connectors. The *transmission convergence sublayer* establishes and maintains the boundaries of the ATM cells in the bit stream; generates and verifies header checksums; inserts and removes "idle" ATM cells when cells are not available for transmission; and, of course, converts ATM cells into a format appropriate for transmission in the given physical medium.

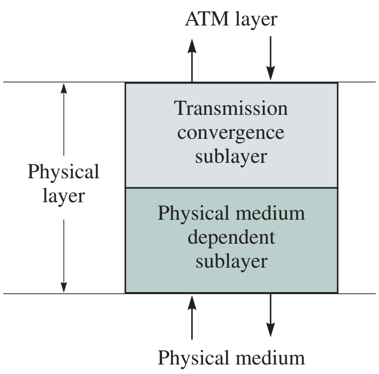


**FIGURE 9.6** ATM physical layer

A large number of physical layers have been defined to provide for support for ATM in a wide range of network scenarios, for example, LAN/WAN and private/public scenarios. The approach in the ATM Forum has been to use and adapt existing physical layer standards as much as possible. In addition to SONET/SDH, physical layers have been defined for 1.5 Mbps (DS-1), 2.0 Mbps (E-1), 45 Mbps (DS-3), 34.4 Mbps (E3), 139 Mbps (E-4), 100 Mbps (FDDI), and 155 Mbps (Fiber Channel). Other physical layers will be defined as needed.

## 9.3 ATM LAYER

The ATM layer is concerned with the sequenced transfer of cells of information across connections established through the network. In this section we examine the operation of the ATM layer in detail. We begin with a description of the ATM cell header. We then discuss how fields in the ATM header are used to identify network connections. This section is followed by a description of the types of ATM network service categories and the traffic management mechanisms required to provide these services. A later section deals with ATM addressing and ATM signaling.

### 9.3.1 ATM Cell Header

Different ATM cell headers have been defined for use in the UNI and in the NNI. The UNI is the interface point between ATM end users and a private or public ATM switch, or between a private ATM switch and a public carrier ATM network, as shown in Figure 9.5. The NNI is the interface between two nodes (switches) in the same ATM network.

Figure 9.7 shows the 5-byte cell header for the UNI. We first briefly describe the functions of the various fields. We then elaborate on their role in ATM networks.

**Generic flow control:** The GFC field is 4 bits long and was intended to provide flow control and shared medium access to several terminals at the UNI. It is currently undefined and is set to zero. The GFC field has significance only at the UNI and is not carried end to end across the network. The UNI and NNI cell headers differ in that the GFC field does not appear in the NNI cell header; instead the VPI field is augmented to 12 bits.

**Virtual path identifier:** The VPI field is 8 bits long, so it allows the definition of up to $2^8 = 256$ virtual paths in a given UNI link. Recall from Figure 7.31 that each virtual path consists of a bundle of virtual channels that are switched as a unit over the sequence of network nodes that correspond to the path.

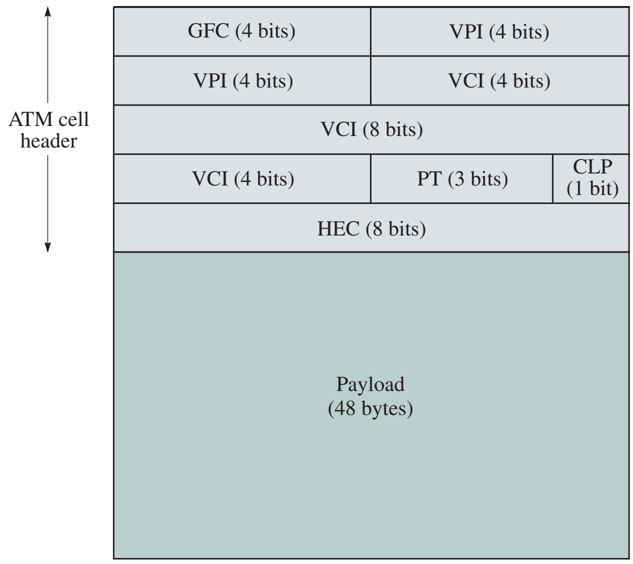


**FIGURE 9.7** ATM cell header format

**Virtual channel identifier:** The VCI field is 16 bits long, so it allows the definition of up to $2^{16} = 65{,}536$ virtual channels per virtual path. The VIP/VCI field is the *local* identifier for a given connection in a given link, and the value of the field changes at every switch.

**Payload type:** The 3-bit payload type field allows eight types of ATM payloads as shown in Table 9.1. The most significant bit is used to distinguish between data cells ($b_3 = 0$) and operations, administration, and maintenance (OAM) cells ($b_3 = 1$).

For data cells ($b_3 = 0$), the second bit serves as the explicit forward congestion indication (EFCI), which is set by switches to indicate congestion and is used by the congestion control mechanism for the available bit rate (ABR) service defined below.

| Payload type identifier | Meaning |
|---|---|
| 000 | User data cell, congestion not experienced, SDU type = 0 (that is, beginning or continuation of SAR-SDU in AAL5) |
| 001 | User data cell, congestion not experienced, SDU type = 1 (that is, end of SAR-SDU in AAL5) |
| 010 | User data cell, congestion experienced, SDU type = 0 |
| 011 | User data cell, congestion experienced, SDU type = 1 |
| 100 | OAM F5 segment associated cell |
| 101 | OAM F5 end-to-end associated cell |
| 110 | Resource management (RM) cell (used in traffic management) |
| 111 | Reserved for future use |

**TABLE 9.1.** ATM payload types

For data cells ($b_3 = 0$), the least significant bit ($b_1$) is carried transparently across the network. We show below that $b_1 = 1$ is used by AAL type 5 (AAL5) to signal that a cell carries the end of a SDU.

The payload field (110) defines resource management cells that are used in traffic management.

**Cell loss priority:** The CLP bit establishes two levels of priorities for ATM cells. A cell that has CLP = 0 is to be treated with higher priority than a cell with CLP = 1 during periods of congestion. In particular, CLP = 1 cells should be discarded before CLP = 0 cells. The CLP bit can be set by terminals to indicate less important traffic or may be set by the network to indicate lower-priority QoS flows or cells that have violated their traffic contract.

**Header error control:** An 8-bit CRC checksum, using the generator polynomial described in Table 3.8, is calculated over the first four bytes of the header. This code can correct all single errors and detect all double errors in the header. The checksum provides protection against misdelivery of cells from errors that may occur in transit. Two modes are defined. In *detection* mode cells with inconsistent checksums are discarded. In *correction* mode single bit errors are corrected. Correction mode is suitable only in media where single errors predominate over multibit errors. The HEC needs to be recomputed at every switch, since the VPI/VCI value changes at every hop.[1]

### 9.3.2 Virtual Connections

In Chapter 7 we describe how ATM uses virtual path and virtual channel identifiers in the cell headers to identify a connection across a network. These locally defined identifiers are used to forward cells that arrive at a switch to the appropriate output port. At each switch the VPI/VCI identifier are used to access tables that specify the output port and the VPI/VCI identifier that is to be used in the next hop. In this manner the chain of identifiers define a connection across the network.

The VPI/VCI format allows ATM to switch traffic at two levels. In VP switching, entire bundles of VCs arriving at a given input port and identified by a given VPI are transferred to the same output port. The switch does not look at the VCI value. Prior to transfer along the next hop, the VPI value is mapped into the value that is defined for the next hop. In VP switching, however, the VCI value is not changed. The ability to handle bundles of VCs at a time is very useful to the network operator in facilitating the management of network resources and in simplifying routing topologies.

[1]The HEC may also be used for cell delineation.

As indicated above, ATM networks provide two basic types of connections. *Permanent virtual connection* (PVCs) are long-term connections that are typically used by network operators to provision bandwidth between endpoints in an ATM network. *Switched virtual connections* (SVCs) are shorter-term connections that are established in response to customer requests. In SVCs the table entries are established during the call setup procedure that precedes the transfer of ATM cells in a connection.

### 9.3.3 QoS Parameters

A central objective of ATM is to provide QoS guarantees in the transfer of cell streams across the network. In ATM the QoS provided by the network is specified in terms of the values of several end-to-end, cell-level parameters. A total of six QoS performance parameters have been specified.

The following three QoS network performance parameters are defined in ATM standards. These parameters are not negotiated at the time of connection setup and are indicators of the intrinsic performance of a given network.

**Cell error ratio:** The *CER* of a connection is the ratio of the number of cells that are delivered with one or more bit errors during a transmission to the total number of transmitted cells. The CER is dependent on the underlying physical medium. The CER calculation excludes blocks of cells that are severely errored (defined below).

**Cell misinsertion rate:** The *CMR* is the average number of cells/second that are delivered mistakenly to a given connection destination (that is, that originated from the wrong source). The CMR depends primarily on the rate at which undetected header errors result in misdelivered cells. The CMR calculation excludes blocks of cells that are severely errored (defined below). Note that CMR is a rate in cells/second rather than a ratio, since the mechanism that produces misinsertion is independent of the number of cells produced in a connection.

**Severely-errored cell block ratio:** A severely-errored cell block event occurs when more than $M$ cells are lost, in error, or misdelivered in a given received block of $N$ cells, where $M$ and $N$ are defined by the network provider. The severely-errored cell block ratio (SECBR) is the ratio of severely-errored cell blocks to total number of transmitted cell blocks in a connection. The SECBR is determined by the properties of the error mechanisms of the transmission medium, by buffer overflows, and by operational effects of the underlying transmission system such as losses of information that occur when active transmission links are replaced by backup transmission links in response to faults.

The following three QoS parameters may be negotiated between the user and the network during connection setup.

**Cell loss ratio:** The *CLR* for a connection is the ratio of the number of lost cells to total number of transmitted cells. The CLR value is negotiated between the user and the network during call setup and specifies the CLR objective for the given connection. It is specified as an order of magnitude in the range of $10^{-1}$ to $10^{-15}$. It can also be left unspecified. The CLR objective can apply to either the CLP = 0 (conforming) cell flow or to the CLP = 0 + 1 (all cells) in the cell flow. The degree to which CLR can be negotiated depends on the sophistication of the buffer-allocation strategies that are available in a given network.

**Cell transfer delay:** The *CTD* is the time that elapses from the instant when a cell enters the network at the source UNI to the instant when it exists at the destination UNI. The CTD includes propagation delay, processing delays, and queueing delays at multiplexers and switches. In general, different cells in a connection experience different values of delays, so the CTD is specified by a probability density function as shown in Figure 9.8.[2] The standards provide for the negotiation of the "maximum" CTD. As shown in Figure 9.8, the *maximum CTD* is defined as the value $D_{max}$ for which the fraction $1 - \alpha$ of all cells have CTD less than $D_{max}$, where $\alpha$ is some appropriately small value. For example, the requested value of CLR places an upper bound on the value of $\alpha$. Queue-scheduling algorithms in the ATM switches can be used to control the CTD experienced by cells in a given connection.

**Cell delay variation:** The *CDV* measures the variablity of the total delay encountered by cells in a connection. The CDV excludes the fixed component $D_0$ of the CTD that is experienced by all cells in a connection, for example, the propagation delay and fixed processing delays (see Figure 9.8). Current standards provide for the negotiation of the peak-to-peak

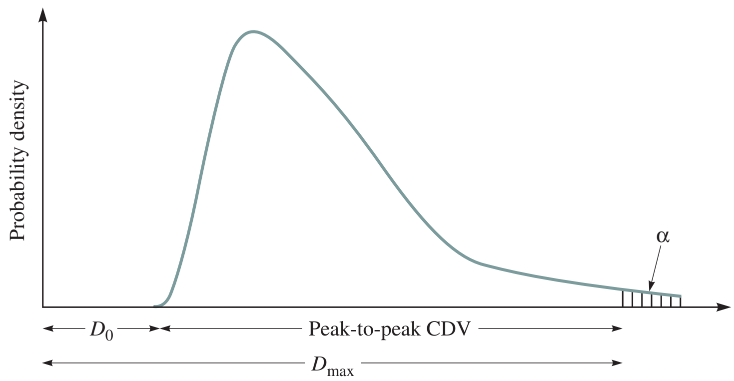


**FIGURE 9.8** Probability density function of cell transfer delay

[2]The probability that the delay value falls in an interval is the area under the probability density function for that interval. See [Leon-Garcia 1994].

CDV, which is simply the difference between the maximum CTD $D_{max}$ and the fixed delay component $D_0$. Note that network switches have only limited control over the spread (variance) of CTD values, and consequently the range of CDV values that can be negotiated for a connection is also limited.

### 9.3.4 Traffic Descriptors

The capability of a network to provide given levels of QoS to a connection depends on the manner in which the connection produces cells for transmission, that is, at a constant smooth rate or in a highly bursty fashion. This capability also depends on the amount of network resources, that is, bandwidth and buffers, that the network allocates to the connection. Therefore, the connection contract between the user and the network must specify the manner in which the source will produce cells. For this purpose standards have specified a number of *source traffic descriptor* parameters. To be enforceable, policing algorithms must be available to monitor the traffic produced by a source to determine whether it conforms to the connection contract.

The following source traffic parameters have been defined to specify this pattern of demand for transmission.

**Peak cell rate:** The *PCR* specifies the rate in cells/second that a source is never allowed to exceed. The minimum allowable interval between cells is given by T = 1/PCR.

**Sustainable cell rate:** The *SCR* is the average cell rate, in cells/second, produced by the source over a long time interval.

**Maximum burst size:** The *MBS*, in a number of cells, specifies the maximum number of consecutive cells that may be transmitted by a source at the peak cell rate (PCR).

**Minimum cell rate:** The *MCR* is the minimum average cell rate, in cells/second, that the source is always allowed to send.

Even if a source produces cells at exactly the PCR rate, subsequent ATM cell multiplexing and physical layer processing (e.g., insertion of a cell into a bit stream) can produce certain variability about the PCR rate. The policing mechanism must take into account this unavoidable variability. The following interface performance parameter has been defined for this purpose.

**Cell delay variation tolerance:** The *CDVT* specifies the level of cell delay variation that must be tolerated in a given connection.

### 9.3.5 ATM Service Categories

ATM connections with arbitrary traffic flow properties and arbitrary QoS are possible by selecting values for the traffic descriptor and the negotiable QoS parameters. In practice there are several clearly identifiable classes of traffic in terms of traffic properties and network QoS requirements. The ATM Forum has defined five *ATM service categories* as shown in Table 9.2. The first two categories apply to connections that are real time in the sense of having stringent delay and timing requirements.

**Constant bit rate:** The *CBR* ATM service category is intended for traffic with rigorous timing requirements, such as voice, circuit emulation, and certain types of video, that require a constant cell transmission rate for the entire duration of a connection. The traffic rate is specified by the PCR. The QoS is specified by the CTD and CDV, as well as by the CLR.

**Real-time variable bit rate:** The *rt-VBR* ATM service category is intended for variable-bit-traffic, such as certain types of video, with rigorous timing requirements. The traffic is described by the PCR, SCR, and MBS. The QoS is specified by the CLR, CTD, and CDV.

Three categories of non-real-time connections have been defined.

**Non-real-time variable-bit-rate:** The *nrt-VBR* ATM service category addresses bursty sources, such as data transfer, that do not have rigorous timing requirements. The traffic is described by the PCR, SCR, and MBS. The QoS is specified by the CLR, and no delay requirements are specified.

**Available bit rate:** The *ABR* ATM service category is intended for sources that can dynamically adapt the rate at which they transmit cells in response to feedback from the network. This service allows the sources

| **Attribute** | **ATM layer service category** | | | | |
|---|---|---|---|---|---|
| | CBR | rt-VBR | nrt-VBR | UBR | ABR |
| *Traffic parameters* | | | | | |
| PCR and CDVT[4,5] | Specified | Specified | Specified | Specified[2] | Specified[3] |
| SCR, MBS, CDVT[4,5] | n/a | Specified | Specified | n/a | n/a |
| MCR[4] | n/a | n/a | n/a | n/a | Specified |
| *QoS parameters* | | | | | |
| peak-to-peak CDVT | Specified | Specified | Unspecified | Unspecified | Unspecified |
| maxCTD | Specified | Specified | Unspecified | Unspecified | Unspecified |
| CLR[4] | Specified | Specified | Specified | Unspecified | See note 1 |
| *Other attributes* | | | | | |
| Feedback | Unspecified | Unspecified | Unspecified | Unspecified | Specified[6] |

*Notes*: [1]CLR is low for sources that adjust cell flow in response to control information. Whether a quantitative value for CLR is specified is network specific.
[2]May not be subject to connection admission control and usage parameter control procedures.
[3]Represents the maximum rate at which the ABR source may ever send data. The actual rate is subject to the control information.
[4]These parameters are either explicitly or implicitly specified for PVCs or SVCs.
[5]CDVT refers to the cell delay variation tolerance. CDTV is not signaled. In general, CDVT need not have a unique value for a connection. Different values may apply at each interface along the path of a connection.
[6]See discussion on ABR in Chapter 7.

**TABLE 9.2.** ATM service category attributes [ATM April 1996]

to exploit the bandwidth that is *available* in the network at a given point in time. The traffic is specified by a PCR and MCR, which can be zero. The sources adjust the rate at which they transmit into the network by implementing a congestion control algorithm that dictates their response to resource management cells that explicitly provide rate flow information. Connections that adapt their traffic in response to the network feedback can expect a low CLR as well as a "fair" share of the available bandwidth.

**Unspecified bit rate:** The *UBR* ATM service category does not provide *any* QoS guarantees. The PCR may or may not be specified. This service is appropriate for noncritical applications that can tolerate or readily adjust to the loss of cells.

In ATM the QoS guarantees are provided on a per connection basis; that is, every connection can expect that its QoS requirements will be met. Since ATM involves the handling of flows of cells from many connections that necessarily interact at multiplexing points, it is worth considering the nature of the QoS guarantees and the corresponding resource allocation strategies for the different ATM service categories.

For CBR connections the source is free to transmit at the negotiated PCR at any time for any duration. It follows then that the network must allocate sufficient bandwidth to allow the source to transmit continuously at the PCR. The scheduling/priority discipline in the multiplexer must ensure that such bandwidth is regularly available to the connection so that the CDV requirement is also met. The steady flow assumed for CBR sources implies that only limited interaction occurs between different CBR flows. In essence, the individual CBR flows act as if they were in separate, isolated transmission links.

For rt-VBR connections the source transmission rate is expected to vary dynamically around the SCR and below the PCR. Therefore, it is to the benefit of the network operator to statistically multiplex these flows to improve the actual utilization of the bandwidth. However, the mixing of rt-VBR flows must be done in a way that maintains some degree of isolation between flows. In particular it is essential to meet the delay and CLR requirements of the connections.

The situation for nrt-VBR connections is similar to that of rt-VBR connections. Again it is in the interest of the network operator to statistically multiplex nrt-VBR sources. In this case the degree of multiplexing is limited only by the commitment to provide conforming flows with the negotiated CLR.

UBR connections, with their lack of any QoS guarantees, provide an interesting contrast to the preceding service categories. When the traffic levels in the network are low, UBR connections may experience performance that is as good as that of the service categories with QoS guarantees. Only as network traffic levels increase, do the QoS guarantees become noticeable. From the point of view of the network operator, a low tariff for UBR service can be used to stimulate demand for bandwidth when the utilization of network sources is low. This approach is useful to a broad range of users when traffic levels are low; however, it is increasingly less useful as traffic levels increase and the network performance experienced becomes less consistent. The ABR service category fills a niche in this context. UBR connections receive no guarantees as network traffic levels vary. ABR connections, on the other hand, are assured of a low level of CLR as long as they conform by responding to the network feedback information. The option to negotiate a nonzero MCR also provides ABR connections with additional assurance of service.

### 9.3.6 Traffic Contracts, Connection Admission Control, and Traffic Management

Traffic management refers to the set of control functions that together ensure that connections receive the appropriate level of service. To provide QoS guarantees to each connection, the network must allocate an appropriate set of resources to each new connection. In particular the network must ensure that new VCs are assigned to links that have sufficient available bandwidth and to ports that have sufficient buffers to handle the new as well existing connections at the committed levels of QoS. The queue scheduling algorithms presented in section 7.6 can be used to ensure that the cells in specific connections receive appropriate levels of performance in terms of delay and loss.

*Connection admission control* (CAC) is a network function that determines whether a request for a new connection should be accepted or rejected. If accepted, the user and network are said to enter into a *traffic contract*. The contract for a connection includes the ATM service category, the traffic descriptors, and the QoS requirements that are introduced in section 9.3.5 and shown in Table 9.2. The CAC procedure takes the proposed traffic descriptors and QoS requirements and determines whether sufficient resources are available along a route from the source to the destination to support the new connection as well as already established connections. Specific CAC algorithms are not specified by the standards bodies and are selected by each network operator.

The QoS guarantees are valid only if the user traffic conforms to the connection contract. *Usage parameter control (UPC)* is the process of enforcing the traffic agreement at the UNI. Each connection contract includes a cell conformance definition that specifies how a cell can be policed, that is, determined to be either conforming or nonconforming to the connection contract. The generic cell rate algorithm (GCRA) is equivalent to the leaky-bucket algorithm described in section 7.7 and can be used to determine whether a cell conforms to the specified PCR and CDVT. Cells that are found to be not conforming are "tagged" by having their cell loss priority (CLP) bit in the header set to 1 at the access to the network. When congestion occurs in the network, cells with CLP = 1 are discarded first. Thus nonconforming cells are more likely to be discarded. The GCRA algorithm can be modified so that cell conformance to the PCR, CDVT, as well as the SCR and MBS can be checked.

A connection is said to be *compliant* with its contract if the proportion of nonconforming cells does not exceed a threshold specified by the network operator. As long as a connection remains compliant, the network will provide the

QoS specified in the contract. However, if a connection becomes noncompliant, the network may then cease providing the contracted QoS.

*Traffic shaping* is a mechanism that allows sources to ensure that their traffic conforms to the connection contract. In traffic shaping, a leaky bucket is used to identify nonconforming cells that are then buffered and delayed so that all cells entering the network are conforming. The token bucket mechanism discussed in Chapter 7 is a method for doing traffic shaping.

Congestion can still occur inside the network even if all cells that enter the network conform to their connection contract. Such congestion will occur when many cells from different connections temporarily coincide. The purpose of *congestion control* is to detect the onset of congestion and to activate mechanisms to minimize the impact and duration of congestion. ATM networks employ two types of congestion control. *Selective cell discarding* involves discarding CLP = 1 cells during periods of congestion. The onset of congestion may be defined by a threshold on the contents of buffers in switches and multiplexers. Discarding low-priority cells helps meet the QoS commitments that have been made to the high-priority cells.

A second class of congestion control involves sending *explicit congestion feedback* from the network to the sources. This type of control is appropriate for ABR connections, which by definition involve sources that can adapt their input rate to network feedback. The rate-based flow control algorithm described in Chapter 7 has been recommended for this type of congestion control.

## 9.4 ATM ADAPTATION LAYER

An application that operates across an ATM network has a choice of the five ATM connection service categories shown in Table 9.2. Every application involves the transfer of one or more blocks or of a stream of information across the network. The ATM service categories provide for the sequenced transfer of cells across the network with a certain delay or loss performance. At the very least a conversion is required from the application data blocks to ATM cells at the source and a conversion back to the application blocks at the destination. One purpose of the ATM adaptation layer is to provide for the mapping between application data blocks to cells.

Applications naturally specify their QoS requirements in terms of their data blocks, not in terms of ATM cells. It is also possible that the service provided by the ATM layer does not meet the requirements of the application. For example, the ATM layer does not provide reliable stream service by itself, since some cell losses can occur. Another purpose of the ATM adaptation layer, then, is to enhance the service provided by the ATM layer to the level required by the application. It should be noted that multiple higher layers may operate over the AAL, for example, HTTP over TCP over AAL. To be more precise, we emphasize that the function of the AAL is to provide support for the layer directly above it. Thus if the layer above the AAL is TCP, then the AAL need not be concerned with providing reliable stream service. On the other hand, the AAL may be called on to provide reliable stream service when such service is not available in the higher layers, as, for example, in signaling applications.

Different applications require a different combination of functions in an AAL. For example, circuit emulation applications require that information be transferred as if the underlying ATM connection were a dedicated digital transmission line. Real-time voice and certain video applications present similar requirements. On the other hand, frame relay applications require the non-real-time, connection-oriented transfer of a sequence of frames between two end systems. In yet another example, IP routers require the connectionless transfer of a packet to another router, using the ATM network as a "data link." In some cases the IP packets carry payloads that are controlled by TCP entities at the end systems. Each of these examples impose different requirements on the AAL.

There have been several attempts to categorize applications into a small set of classes and to design AALs to meet the requirements of each class. These efforts have not met with success, and no clear correspondence exists between application classes, AALs, and ATM connection service categories. Our approach here is to discuss the formats and services of the AALs that have been developed to date. We then discuss what combinations of applications, AALs, and ATM service categories make sense. It should be noted that users are also free to use proprietary (nonstandard) AALs.

The AAL is divided into two sublayers as shown in Figure 9.9. The purpose of the **segmentation and reassembly (SAR)** sublayer is to segment the PDUs of the higher layer into blocks that are suitable for insertion into the ATM cell payloads at the source and to reassemble the higher-layer PDUs from the sequence of received ATM cell payloads at the destination. The **convergence sublayer (CS)** is divided into a **common part (CPCS)** and a **service-specific part (SSCS)**. The CPCS deals with packet framing and error-detection functions that all AAL users require. The SSCS provides functions that depend on the requirements of specific classes of AAL users. Consequently, each AAL usually has a specific SAR and CPCS sublayer and several optional SSCS sublayers.

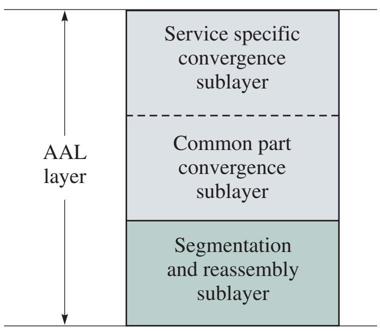


**FIGURE 9.9** AAL sublayers

### 9.4.1 AAL1

The ATM adaptation layer type 1 (AAL1) supports services that require the transfer of information at a constant rate. Examples of this type of service are a single 64 kbps PCM voice call, sub-T-1 connections that consist of $n \times 64$ kbps streams, T-1/E-1 and other digital circuits from the telephone hierarchy, and constant bit-rate digital video. The AAL PDU structure contains fields that enable clock recovery and sequence numbering. It also contains an option for the transfer of the internal (frame) structure within a continuous bit stream.

The generic AAL1 process is shown in Figure 9.10. The convergence sublayer function takes the user data stream, optionally inserts a 1-byte pointer to provide structure information, and produces 47-byte CS PDUs, which it then passes with three-bit sequence numbering to the segmentation and reassembly sublayer. Thus the CS PDU normally contains either 47 bytes or 46 bytes of user information, depending on whether a pointer is inserted. Note that the CS PDU need not be completely filled with 47 bytes of user information. In low-bit-rate applications with low-delay requirements, for example, a single 64 kbps voice call, the CS PDU may be required to pass 47-byte blocks that are only partially filled with user information.

Figure 9.11 shows the AAL1 PDUs. The SAR sublayer attaches a 1-byte header to each CS PDU as shown in Figure 9.11a. The first four bits constitute the *sequence number (SN)* field. The first bit in the SN field is the *convergence sublayer indicator (CSI)* and is followed by the 3-bit sequence number that can be used for the detection and monitoring of loss and misinsertion of SAR payloads, and hence cells. In even-numbered cells the CSI bit may be used to indicate the existence of a CS sublayer pointer to the destination, as shown in Figure 9.11b. In odd-numbered cells the CSI bit may also optionally be used to convey timing information from the source to the destination. The last four bits of the header contain *sequence number protection (SNP)* check bits that provide error-

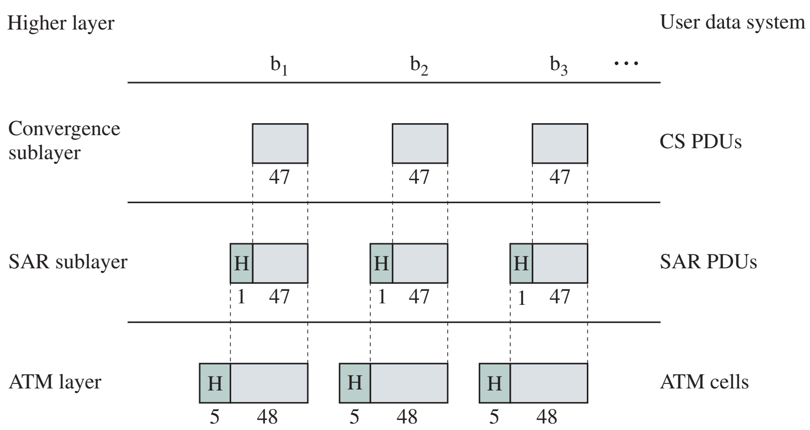


**FIGURE 9.10** AALI process

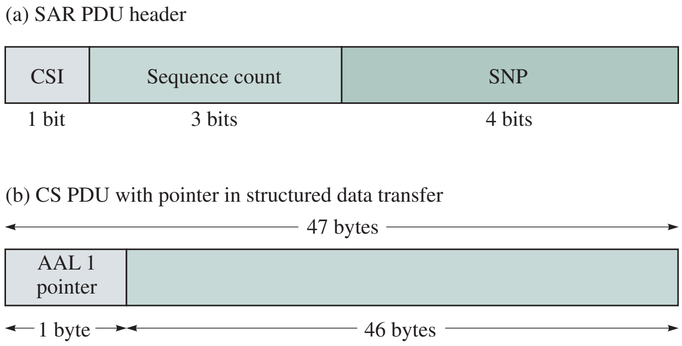


**FIGURE 9.11** AALI PDUs

detection and correction capability for the SAR header. The four check bits are computed by using a Hamming (7,4) code with an additional overall parity check bit. This makes it possible to correct all single-bit and to detect all double-bit error patterns.

As an example of the use of AAL1 for unstructured information transfer, consider the transfer of a T-1 connection. Recall that the T-1 stream consists of a frame consisting of one framing bit followed by 24 bytes at a repetition rate of 8 kHz. In unstructured AAL1 transfer the bits from the T-1 stream are grouped into blocks of 47 bytes and passed to the SAR. Note that the bytes in the CS PDU will not be aligned to the bytes in the T-1 stream, since each frame has an odd number of bits. Now suppose that the T-1 connection uses *structured data transfer*. The T-1 stream is now viewed as a sequence of 24-byte frames (the framing bit is ignored). The bytes in the T-1 stream are mapped directly into the bytes in the CS PDUs. By prior agreement between the source and destination AAL entities, every so many cells, eight, for example, the first byte of the CS PDU contains a *pointer* that can be used to determine the beginning of the next frame. Because pointers can be inserted only in even-numbered cells, the beginning of the frame can be anywhere in the payloads of the current or the next cell. The periodic insertion of pointers provides protection against loss of synchronization that could result from cell losses.

The convergence sublayer at the destination can provide a number of services that are useful to applications requiring a constant transfer rate. The odd-numbered CSI bits can be used to convey a residual timestamp that provides the relative timing between the local clock and a common reference clock. The destination uses these residual timestamps to reconstruct the source clock and replay the received data stream at the correct frequency. This synchronous residual timestamp method is described in section 5.5. It should be noted that other methods for clock recovery, such as adaptive buffer, do not require the use of CSI bits. These methods are also discussed in Chapter 5.

To deliver information to the user at a fixed rate, the convergence sublayer at the destination can carry out timing recovery and then use a playout technique to absorb the cell delay variation in the received sequence. The convergence sublayer can also use the sequence numbers to detect lost or misinserted cells. The capability to request retransmissions is not provided, so the CS can provide only an indication to the user that a loss or misinsertion has occurred.

The convergence sublayer can also implement either of two forward-error correction (FEC) techniques to correct for cell errors and losses. For applications that have a low-delay requirement, the first FEC technique operates on groups of 15 cells and adds sufficient check bits to form 16 cells. The technique can correct one lost cell per group of 16 and can also correct certain other error patterns. The additional FEC delay incurred is then 15 cells. The second FEC technique uses a variation of the interleaving techniques discussed in Chapter 3 to arrange the CS-PDUs of 124 cells as columns in an array. Four additional columns of check bits are added to provide the capability to correct up to four cell losses as well as certain other error patterns. The technique involves incurring a delay of 124 cells at the source and at the destination, so it is generally appropriate only for non-CTD-sensitive traffic, for example, video streaming.

### 9.4.2 AAL2

AAL type 2 (AAL2) was originally intended to provide support for applications that generate information at a bit rate that varies dynamically with time and that also has end-to-end timing requirements. The prime example of such an application is video that when compressed produces a bit stream that varies widely depending on the degree of detail and the degree of motion in a scene. The development of an AAL for this type of traffic was never completed and the ITU subsequently began work on the development of an AAL, also designated type 2, for a different class of applications. We will discuss this latter AAL in this section.

The new AAL2 is intended for the bandwidth-efficient transfer of low-bit-rate, short-packet traffic that has a low-delay requirement. In effect the AAL2 adds a third level of multiplexing to the VP/VC hierarchy of ATM so that two or more low-bit-rate users can share the same ATM connection. An example where this functionality is required arises in the transfer of compressed voice information from a base station in a digital cellular system to its mobile telephone switching office as shown in Figure 9.12. The low-bit-rate digital streams for individual voice calls need to be transferred in a timely fashion to the switching office where the actual telephone switching takes place. The low delay and low bit rate imply that cell payloads would be only partially filled if each call had its own VC. AAL2 multiplexes the streams from multiple calls to provide both low delay and high utilization of cell payloads.

Figure 9.13 shows the operation of the AAL2. The AAL2 layer is divided into the common part sublayer (CPCS) and the service-specific convergence sublayer (SSCS). In this discussion we focus on the CPCS that transfers CPCS SDUs from a CPCS user at the source to a CPCS user at the destination. The CPCS provides nonassured operation; that is, SDUs may be delivered incorrectly or not at all, as a result of cell losses. The CPCS can multiplex SDUs from

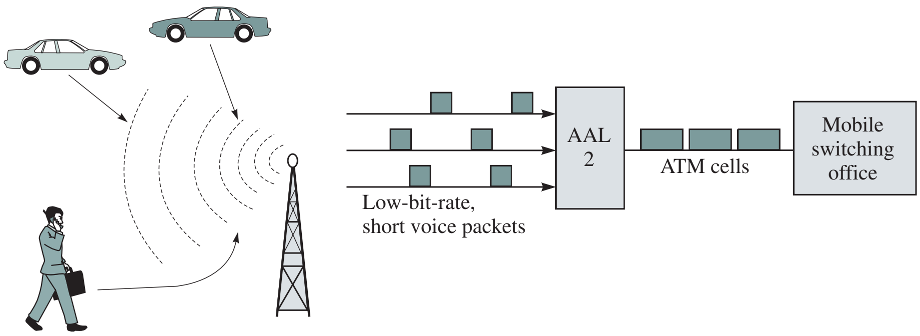


**FIGURE 9.12** Application scenario for AAL2

multiple SSCS users. These users can be of different type and involve different SSCS functions.

User layer packets are transferred to the AAL2 layer as shown in Figure 9.13. These packets can vary in size, since users can be of different type. The maximum allowable packet size is 64 bytes. Assuming that the SSCS is not present, a three-byte header is added to each packet to form a CPCS packet. As shown in Figure 9.14a, the first byte is the channel identifier (CID) that identifies each user. These AAL channels are bidirectional, and the same CID is used for both directions. The six higher-order bits of the next byte are a length indicator that specifies one less than the number of bytes in the CPCS-packet payload. The remaining two bits of the second byte in the header specify the

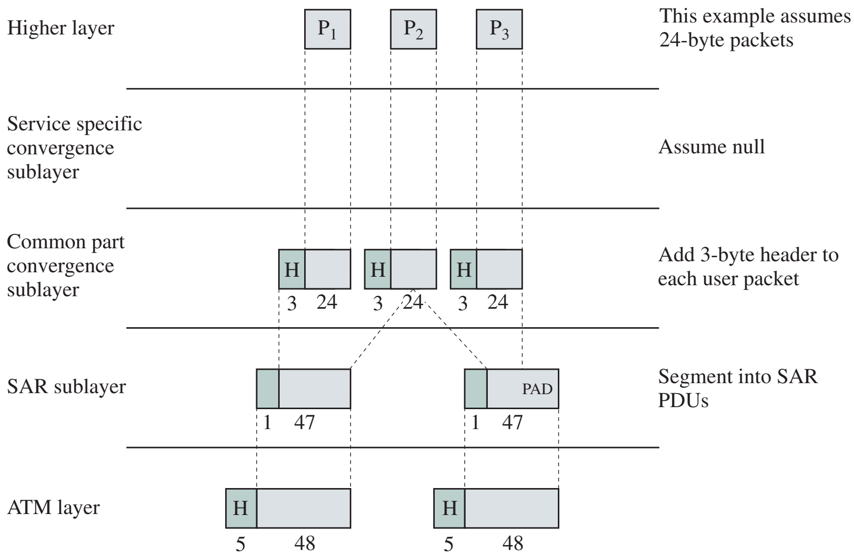


**FIGURE 9.13** AAL2 process

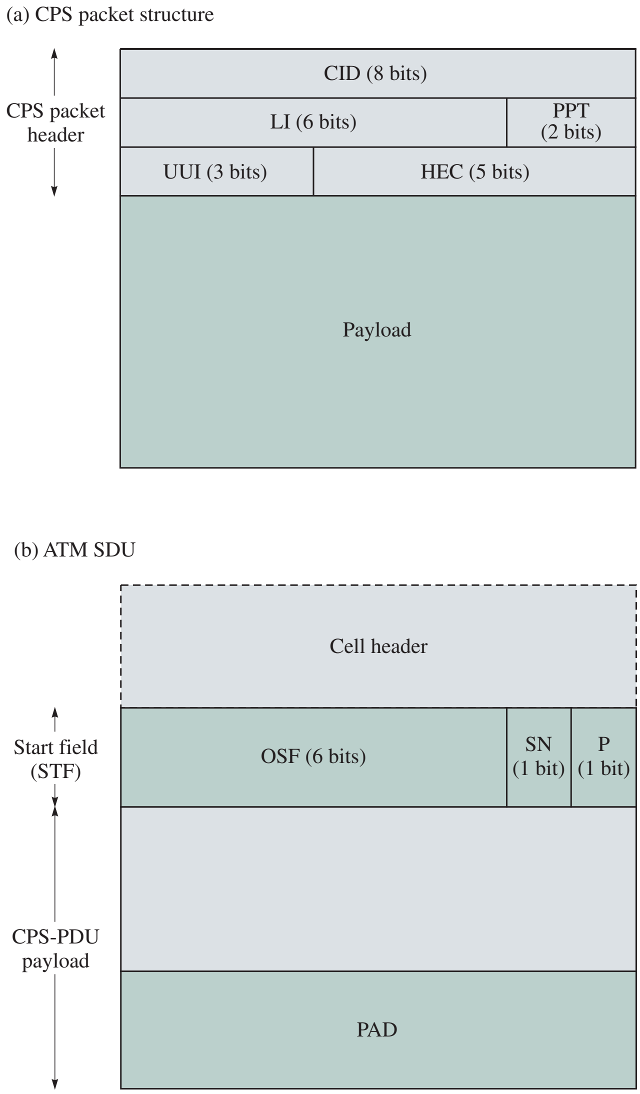


**FIGURE 9.14** CPS packets

packet payload type (PPT). A value of 3 indicates that the CPCS packet is serving an OAM function. When the value is not 3, the packet is serving the application-specific functions, for example, the transfer of voice. The higher-order three bits of the third header byte are the user-to-user indication (UUI). When the PPT is not 3 the UUI is carried transparently between the SSCS protocol entities. When the PPT is 3, the UUI is carried transparently between the AAL layer management entities. The final five bits of the header are check bits that are used to detect errors. The encoding uses the generator polynomial $g(x) = x^5 + x^2 + 1$.

As shown in Figure 9.13, the CPCS packets are concatenated one after another prior to segmentation into 48-byte ATM SDUs. Each ATM SDU consists of a 1-byte start field and 47 bytes of CPCS packet bytes or padding (see Figure 9.14b). The start field provides the offset from the end of the STF to the start of the first CPCS packet in the SDU or, in the absence of such, the start of the PAD field. The maximum CPCS packet size is 64 bytes, so a CPCS packet may overlap one or two ATM cell boundaries. To meet the low-delay requirements, it is possible for the payload in the last cell to be only partially filled as shown, for example, in the second cell in Figure 9.13.

### 9.4.3 AAL3/4

In the early ATM standardization efforts, AAL type 3 (AAL3) was intended for applications that generate bursts of data that need to be transferred in connection-oriented fashion with low loss, but with no delay requirement. AAL type 4 (AAL4) was similarly intended for connectionless transfer of such data. By convention *all* connectionless packets at the UNI use the *same* VPI/VCI number, so a multiplexing ID was introduced in AAL4 to distinguish different packets. The efforts to develop AAL3 and AAL4 were later combined to produce AAL3/4, which can be used for either connection-oriented or connectionless transfer. The distinguishing feature of AAL3/4 is that it allows long messages from multiple users to be simultaneously multiplexed and interleaved in the same ATM VC.[3]

AAL3/4 can operate in two modes: message mode and stream mode. In message mode the AAL accepts a single user message for segmentation into ATM payloads, and the destination delivers the message. In stream mode one or more user PDUs, each as small as a single byte, are accepted at a time by the AAL and are subsequently delivered to the destination without an indication of the boundaries between the original PDUs. Both modes allow for assured or nonassured operation. In assured operation an end-to-end protocol is implemented in the SSCS to allow for the error-free delivery of messages. In nonassured operation, messages may be delivered in error or not at all.

The AAL3/4 process in message mode is shown in Figure 9.15. The user information is first passed to the SSCS and then to the CPCS, which adds fill bytes to make the CPCS payload a multiple of four bytes (32 bits). The CPCS PDU is formed by adding four bytes of header and four bytes of trailer. The CPCS PDU is passed to the SAR sublayer, which produces SAR PDUs with 4 bytes of overhead and 44 bytes of payload. If necessary, the last SAR PDU is

[3]See Goralski for a discussion on the merging of AAL3 and AAL4.

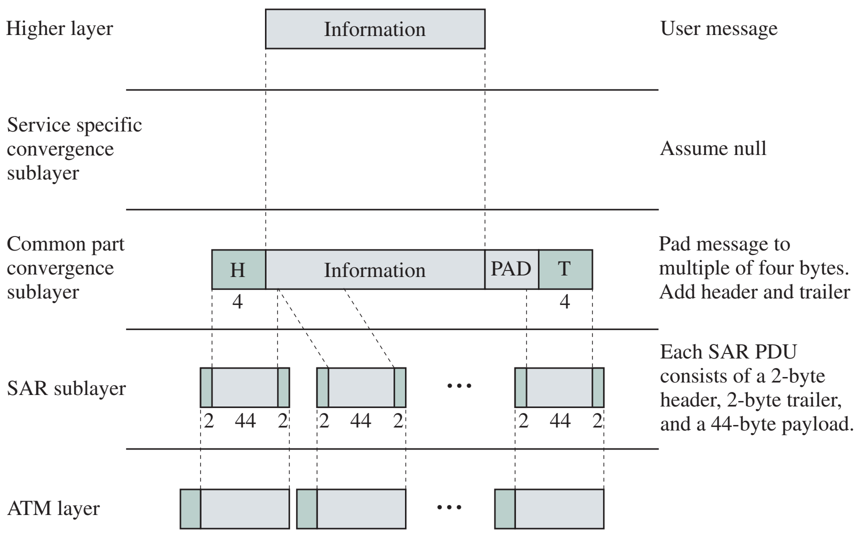


**FIGURE 9.15** AAL3/4 process

padded to fill the payload. Finally the 48-byte SAR PDUs are passed to the ATM layer.

The CPCS PDU has a header, followed by the CPCS-PDU payload, possibly with padding, and finally a trailer. The CPCS-PDU payload can have a length of 1 to 65,535 bytes. As shown in Figure 9.16a, the header begins with a one-byte common part indicator (CPI) field that specifies how subsequent fields are to be interpreted. Only CPI = 0 has been defined to indicate that the BAsize and Length fields are to be interpreted in units of bytes. The one-byte beginning tag (Btag) field and the end tag (Etag) field are set to the same value at the

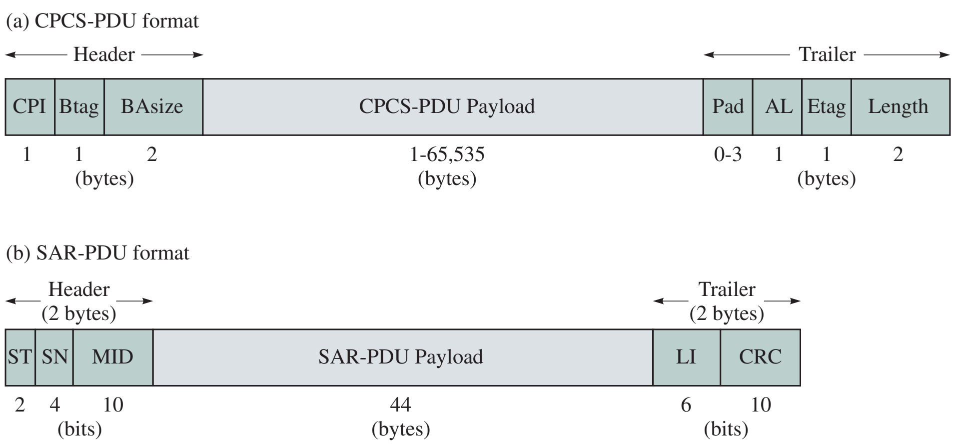


**FIGURE 9.16** AAL3/4 CPCS and SAR formats

source and changed for each successive CPCS PDU. This practice allows the destination to detect when the incorrect header and trailer have been associated. The two-byte buffer allocation size indication (BAsize) field informs the destination of the maximum buffer size required to receive the current CPCS PDU. The padding field contains zero to three pad bytes to make the CPCS PDU a multiple of four bytes. This approach ensures that the trailer part will be aligned to a 32-bit boundary, making it easier to process the trailer. The alignment field consists of a byte of zeros to make the trailer four bytes long. The two-byte length field indicates the length of the payload.

Figure 9.16b shows that the SAR PDU contains a 2-byte header, 44 bytes of payload, and a 2-byte trailer. The first two bits in the header are the segment type: the value 10 indicates that the PDU contains the beginning of a message (BOM), 00 denotes continuation of message (COM), 01 indicates end of message (EOM), and 11 indicates a single-segment message (SSM). The next four bits provide the sequence number (SN) of the SAR PDU within the same CPCS PDU. The sequence numbers are used to ensure correct sequencing of cells when the CPCS PDU is rebuilt at the destination. The remaining 10 bits in the header are for the **multiplexing identifier**, also called **message identifier**, **(MID)**. The MID allows the SAR sublayer to multiplex the messages of up to $2^{10}$ AAL users on a single ATM VC. All SAR PDUs of the same CPCS PDU have the same MID. The six-bit length indicator (LI) in the trailer specifies the size of the payload. Except for the last cell, all cells for a given CPCS PDU are full, so LI = 44. The last cell can have LI from 4 to 44. The 10-bit CRC provides for the detection of errors that may occur anywhere in the PDU.

Figure 9.17 elaborates on how multiplexing is done in AAL3/4. When multiple users share the same VC, the messages from each user are sent to a *different*

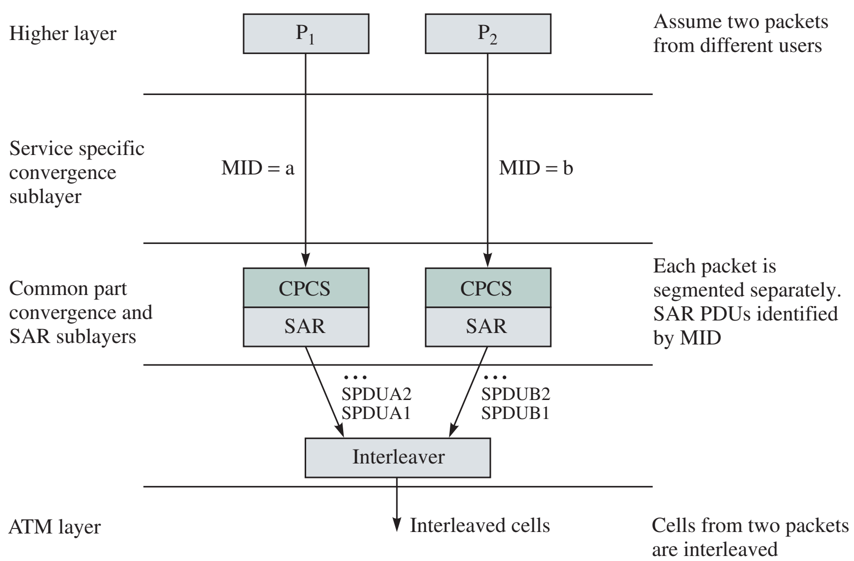


**FIGURE 9.17** Multiplexing in AAL3/4

instance of the AAL. Each such AAL will produce one or more SAR PDUs that are then intermixed and interleaved prior to being passed to the ATM layer. Thus the SAR PDUs from different users arrive intermixed at the destination. The MID allows the messages from each user to be reassembled. Note that the MID can be viewed as setting up a very short term connection within the VC. Each such connection is for the duration of a single packet transfer and is delimited by the BOM and EOM cells. The MID feature of AAL3/4 was developed to provide compatibility with the IEEE 802.6 Metropolitan Area Standard, which provides connectionless LAN interconnection service.

A problem with AAL3/4 is that it is heavy in terms of overhead. Each message has at least eight bytes added at the CSCP sublayer, and subsequently each ATM cell payload includes four additional bytes of overhead. The 10-bit CRC and the 4-bit sequence numbering also may not provide enough protection. These factors led to the development of AAL5.

### 9.4.4 AAL5

AAL type 5 (AAL5) provides an efficient alternative to AAL3/4. AAL5 forgoes the multiplexing capability of AAL3/4 but does support message and stream modes, as well as assured and nonassured delivery.

Figure 9.18 shows the operation of AAL5. A user PDU is accepted by the AAL layer and is processed by the SSCS if necessary. The SSCS then passes a block of data to the CPCS, which attaches 0 to 47 bytes of padding and an 8-byte

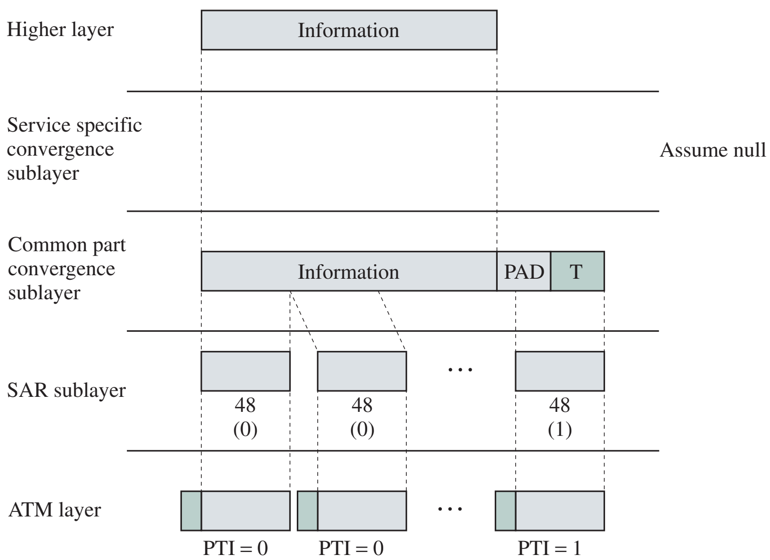


**FIGURE 9.18** AAL5 process

| Information | Pad | UU | CPI | Length | CRC |
|---|---|---|---|---|---|
| 0–65,535 (bytes) | 0–47 | 1 | 1 | 2 (bytes) | 4 |

**FIGURE 9.19** AAL5 PDU

trailer to produce a CPCS PDU that is a multiple of 48 bytes. The maximum CPCS-PDU payload is 65,535 bytes.

As shown in Figure 9.19, the trailer contains one byte of UU that is passed transparently between the end-system entities, one byte of CPI that aligns the trailer to eight bytes, a two-byte LI that specifies the number of bytes of user data in the CPCS-PDU payload, and a four-byte CRC check to detect errors in the PDU. The SAR sublayer segments the CPCS PDU into 48-byte payloads that are passed to the ATM layer. The SAR also directs the ATM layer to set the PTI field in the header of the last cell of a CPCS PDU. This step allows the boundary between groups of cells corresponding to different messages to be distinguished at the destination. Note that unlike AAL3/4, AAL5 can have only one packet at a time in a VC because the cells from different packets cannot be intermixed.

AAL5 is much more efficient than AAL3/4 is, as AAL5 does not add any overhead in the SAR sublayer. AAL5 does not include sequence numbers for the SAR PDUs and instead relies on its more powerful CRC checksum to detect lost, misinserted, or out-of-sequence cells. AAL5 is by far the most widely implemented AAL.

### 9.4.5 Signaling AAL

The signaling AAL (SAAL) has been standardized as the AAL in the control plane. The SAAL provides reliable transport for the signaling messages that are exchanged among end systems and switches to set up ATM VCs.

The SAAL is divided into a common part and a service-specific part as shown in Figure 9.20. The service-specific part in turn is divided into a service-specific connection-oriented protocol (SSCOP) and a service-specific coordination function (SSCF). The SSCF supports the signaling applications above it by mapping the services they require into the services provided by the SSCOP. SSCF sublayers have been developed for UNI and NNI.

The SSCOP is a peer-to-peer protocol that provides for the reliable transfer of messages. It provides for the ordered delivery of messages in either assured or unassured mode. In assured mode SSCOP uses a form of Selective Repeat ARQ for error recovery. The ARQ protocol is suitable for use in situations that have large delay-bandwidth products, for example, satellite channels and ATM links. Thus the protocol uses large window sizes to provide sequence numbers for the transmitted packets. To achieve bandwidth efficiency, selective retransmission is used. SSCOP makes use of the service provided by the convergence sublayer and

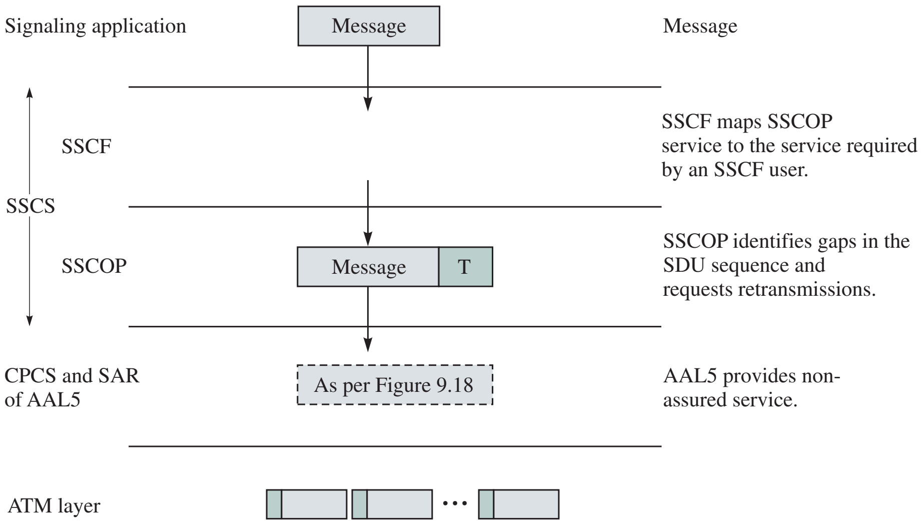


**FIGURE 9.20** SAAL process

SAR sublayers of AAL5 as shown in Figure 9.20. The destination AAL5 layer passes SDUs up to the SSCOP layer only if their checksum is correct. The SSCOP buffers all such SDUs and looks for gaps in the SDU sequence. Because ATM is connection-oriented, the SSCOP knows that the SDUs corresponding to these gaps have errors or have been lost. The sender periodically polls the receiver to find the status of the receive window. Based on the response from the receiver, the transmitter then selectively retransmits the appropriate SDUs. This approach allows SSCOP to ensure that SDUs are retransmitted and that messages are delivered in the correct order and without errors.

Figure 9.21 shows the structure of the SSCOP PDU. A trailer is added, consisting of zero to three bytes of padding, a two-bit pad length indicator, a two-bit reserved (unassigned) field, and a four-bit PDU type field to specify the type of message in the payload. The payload types include sequenced data messages as well as the poll and other messages used by the SSCOP protocol. A 24-bit sequence number is provided for each CPCS PDU. Note that the cell error and loss detection capabilities are provided by the 32-bit CRC that is used in the AAL5 layers, as per Figure 9.19.

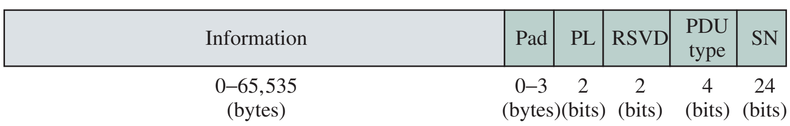


**FIGURE 9.21** SSCOP PDU

### 9.4.6 Applications, AALs, and ATM Service Categories

We saw in the previous section that AALs can be called upon to support a wide range of applications, from emulating a digital transmission line to transferring packet streams of various types. Table 9.3 lists features that characterize the requirements of various types of applications. Table 9.4 summarizes the capabilities of the various AALs.

The application/higher layer that operates over the AAL may require that information be transferred in the form of a stream or as discrete messages. The transfer may involve a constant rate of transfer or may vary with time. The application may be satisfied with a nonassured delivery of information, where there may be occasional losses or misdeliveries, or it may require high levels of assurance on the correct delivery of all information. In the case of nonassured delivery, there may be various degrees of tolerance to errors in the delivered information. Some applications may be tolerant to relatively high delays and significant levels of delay variation, whereas other applications may require tight tolerance in the delay and the jitter. The ability to multiplex streams/packets from different users is an important requirement in certain settings. Finally, the demand for efficiency in the use of the cell payload will depend on the cost of bandwidth in a given setting.

Various combinations of the features in Table 9.3 appear in different applications. A simple example such as voice can require different combinations of features in different contexts. Voice-based applications are almost always stream oriented. In many situations these applications are error tolerant. However, the degree of error tolerance is different when the stream is carrying voice than when it is carrying modem signals that carry data or fax. Furthermore, as the bit rate of voice signals decreases with compression, the degree of error tolerance also decreases. If silence suppression is used, the stream becomes variable bit rate rather than constant bit rate. Telephone conversations between humans requires real-time transfer with low delay and jitter. But voice mail and voice response applications are much more tolerant of delay. Finally, multiplexing may be important in voice applications where bandwidth costs are significant, but not relevant where bandwidth is cheap as in LANs.

| **Feature** | **Application Requirements** | |
|---|---|---|
| *Transfer granularity* | Stream | Message |
| *Bit rate* | Constant | Variable |
| *Reliability* | Nonassured | Assured |
| *Accuracy* | Error tolerant | Error intolerant |
| *Delay sensitivity* | Delay/jitter sensitive | Delay/jitter insensitive |
| *Multiplexing* | Single user | Multiple users |
| *Payload efficiency* | Bandwidth inexpensive | Bandwidth expensive |

**TABLE 9.3.** Features that characterize application requirements

| Sublayer | Feature | AAL1 | AAL2 | AAL3/4 | AAL5 | SAAL |
|---|---|---|---|---|---|---|
| SSCS | Forward error control | Optional | Optional | Optional | Optional | No |
| | Error detection and retransmission | No | No | Optional | Optional | SSCOP |
| | Timing recovery | Optional | Optional | No | Optional | No |
| CPCS | Multiplexing | No | 8-bit CID | 10-bit MID | No | No |
| | Framing structure | Yes | No | No | No | No |
| | Message delimiting | No | Yes | Yes | PTI | PTI |
| | Advance buffer allocation | No | No | Yes | No | No |
| | User-to-user indication | No | 3 bits | No | 1 byte | No |
| | Overhead | 0 | 3 bytes | 8 bytes | 8 bytes | 4 bytes |
| | padding | 0 | 0 | 4 bytes | 0–44 bytes | 0–47 bytes |
| | Checksum | No | No | No | 32 bit | 32 bit |
| | Sequence numbers | No | No | No | No | 24 bit |
| SAR | Payload/overhead | 46–47 bytes | 47 bytes | 44 bytes | 48 bytes | 48 bytes |
| | Overhead | 1–2 bytes | 1 byte | 4 bytes | 0 | 0 |
| | Checksum | No | No | 10 bits | No | No |
| | Timing information | Optional | No | No | No | No |
| | Sequence numbers | 3 bit | 1 bit | 4 bit | No | No |

**TABLE 9.4.** Capabilities of the AAL types

Given the diversity of requirements in different voice applications, it is not surprising that many of the AALs have been adapted for use with different voice applications. A simple AAL1 has been adapted for the transfer of individual 64 kbps voice calls. Other versions of AAL1 with structured data transfer (SDT) are intended for handling multiples of 64 kbps calls. In addition, AAL2 provides the capability to multiplex several low-bit-rate voice calls. AAL5 has also been adapted to carry voice traffic. The rationale here was that AAL5 was required for signaling, so the cost of adding AAL1 could be avoided by operating voice over AAL5.

The choice of which ATM service category to use below the AAL depends on the performance requirements the AAL is committed to deliver to the service above it. It also depends on the manner in which the user level passes SDUs to the AAL because this feature influences the manner in which the cell traffic is generated. Thus the CBR and the rt-VBR service categories are suitable for applications with a real-time requirement, for example, voice over AAL5 over CBR. However, under certain network loading conditions, the other service categories may provide adequate performance. For example, under low network loads voice over AAL5 over UBR may give adequate performance. Policing may also be required to ensure that the cell stream produced by the AAL conforms to the connection contract.

Data transfer applications require different combinations of features in the AAL. These applications tend to be message oriented and variable in bit rate. Best-effort (nonassured) service is usually sufficient, but certain applications can require assured service. Most data transfer applications are relatively insensitive to delays, but it is possible to conceive of monitoring/control applications that require very low delay. As indicated above, AAL5 with nonassured service is the most widely deployed AAL. AAL5 with SSCOP has also been used to provide assured service in the user plane.

Video applications represent a third broad class of applications that include relatively low-bit-rate videoconferencing, for example, $n \times 64$ kbps, to high-quality video distribution and video on demand. Delay plays an important part in applications that involve real-time interactivity. Cell-delay variation plays an important role in almost all video applications because the receiver must compensate for the CDV and present the data to the video decoder at very nearly constant rate. This stringent CDV requirement arises from the manner in which traditional analog television signals deal with the three color components. Very small errors in synchronization among the color components have a dramatic impact on the picture quality.

As a concrete example, consider the transport of constant-bit-rate MPEG2 video in a video-on-demand application. An issue in the design of such a system is whether the timing recovery should be done by the AAL, by the MPEG2 systems layer,[4] or possibly by a layer in between. Another issue involves the degree of error detection and correction required and the use of error concealment techniques. Efficiency in the use of cell payload is also a concern. The recommendation developed by the ATM Forum addressed the issues as follows. AAL1 was not selected for a number of reasons. The SRTS capability of AAL1 could not be used because many end systems do not have access to the common network clock required by the method. The FEC interleaving technique was deemed to introduce too much delay and to be too costly. On the other hand, AAL5 is less costly because of its much wider deployment. It was also found that AAL5 over CBR ATM service could carry MPEG2 transport packets (of 188 bytes each) with sufficiently low jitter that timing recovery could be done in the MPEG2 systems layer. The recommendation does not require that the CBR ATM stream coming out of the video server be shaped to conform to the negotiated CBR parameters.

## 9.5 ATM SIGNALING

The utility of a network is directly related to the capability to connect dynamically to any number of destinations. Signaling provides the means for dynamically setting up and releasing switched virtual connections in an ATM network. The establishment of an end-to-end connection involves the exchange of signaling messages across a number of interfaces, for example, user-network interface (UNI), network-network interface (NNI), and broadband intercarrier interface (B-ICI). Signaling standards are required for each of these interfaces. In this section we focus on the signaling standards for UNI and NNI.

[4]MPEG video coding is discussed in Chapter 12.

The establishment of dynamic connections requires the ability to identify endpoints that are attached in the network. This function is provided by network addresses, the topic of the next section.

### 9.5.1 ATM Addressing

ATM uses two basic types of addresses: telephony-oriented E-164 addresses intended for use in public networks and ATM end system addresses (AESAs) intended for use in private networks. E-164 telephone numbers can be variable in length and have a maximum length of 15 digits. For example in the United States and Canada, 11-digit numbers are used: 1-NPA-NXX-ABCD, for example, 1-416-978-4764. The first 1 is the ITU assigned country code; the next three digits, NPA, are the area code; the following three digits, NXX, are the office code; and the final four digits, ABCD, are the subscriber number. Telephone numbers for other countries begin with a different country code and then follow a different format.

AESAs are based on the ISO Network Service Access Point (NSAP) format that consists of 20-byte addresses. The NSAP format has a hierarchical structure as shown in Figure 9.22. Each address has two parts.

- The *initial domain part (IDP)* identifies the administrative authority that is responsible for allocating the addresses that follow.

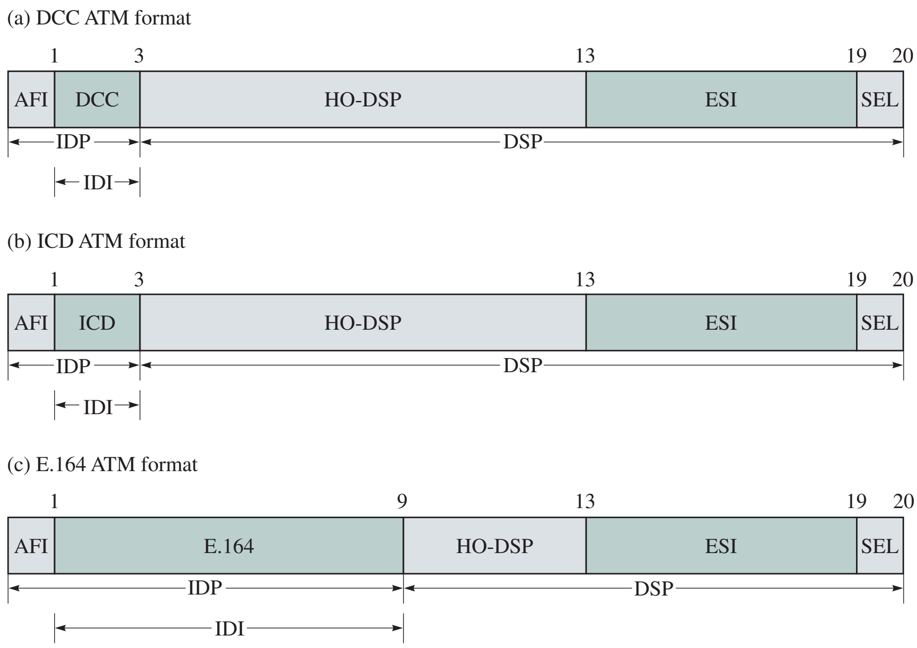


**FIGURE 9.22** ATM formats

- The *domain-specific part (DSP)* contains the address allocated by the given authority.

The IDP itself consists of two parts: a one-byte authority and format identifier (AFI) identifies which structure is to follow and the initial domain identifier (IDI) specifies the authority that allocates the DSP that follows.

Figure 9.22 shows the format of the three initial types of AESAs: data country code (DCC) with AFI = $39_{HEX}$; international code designator (ICD) with AFI = $47_{HEX}$; and E.164 (contained within the AESA format) with AFI = $45_{HEX}$. From Figure 9.22a, it can be seen that the IDI in the DCC format is 2 bytes long, followed by a 10 byte "higher order domain specific part (HO-DSP)." For example, IDI = $840_{HEX}$ identifies the United States, for which ANSI administers the DCC addresses. The format used by ANSI is as follows: the first three bytes identify the organization that uses these addresses, the next byte is used for other purposes, and the final six bytes are for the organization to assign. The ICD addresses are administered by the British Standards Institute. The ICD format, shown in Figure 9.22b, places a four-digit code in the IDI bytes to identify an organization, which is then allowed to administer the next 10 bytes. The six-byte end system identifier (ESI) identifies an end system and usually consists of a MAC address. The one-byte selector (SEL) can be used by the end system to identify higher-layer protocol entities that are to receive the traffic. The AESA format for E-164 addresses is shown in Figure 9.22c. The IDI consists of eight bytes that can hold the 15 digits of the E-164 address.

E-164 "native" addresses are supported in the public UNI and in the B-ICI. AESAs are supported in the private UNI and NNI, as well as in the public UNI. The ATM Forum and other bodies are working on the interworking of these public and private addresses to provide end-to-end connectivity.

### 9.5.2 UNI Signaling

The ATM signaling standards are based on the standards developed for telephone networks. We saw in Chapter 4 that telephone networks use two signaling standards: ISDN signaling (Q.931) is used in the exchange of call setup messages at the UNI; the ISUP protocol of Signaling System #7 is used to establish a connection from a source switch to a destination switch within the network. ATM signaling has developed along similar lines with signaling procedures developed for the UNI, the NNI, and the B-ICI. In this section we consider UNI signaling.

ITU-T recommendation Q.2931, derived from Q.931, specifies B-ISDN signaling at the *ATM UNI*. ATM Forum UNI signaling 4.0 is based on Q.2931. A number of messages have been defined for use in the setup and release of connections. ATM connections involve many more parameters than narrowband ISDN involves, so the signaling messages carry special fields, called *information elements (IEs)*, that describe the user requests. These signaling messages are transferred across the UNI using the services of the SAAL layer in the control plane. Recall that the SAAL provide reliable message transfer using the SSCOP protocol that operates over AAL5. ITU-T recommendation Q.2130 specifies the SSCF that is used between the Q.2931 signaling application and SSCOP. The signaling cells that are produced by AAL5 use the default virtual channel identifier by VPI = 0 and VCI = 5.

Table 9.5 shows the capabilities provided by UNI 4.0. These capabilities are categorized as being applicable to end-system or switch equipment and as being mandatory (M) or optional (O). Point-to-point as well as point-to-multipoint calls are supported. Point-to-point ABR connections are supported. Signaling of individual QoS parameters and negotiation of traffic parameters are also supported. The leaf-initiated join capability allows an end system to join a point-to-multipoint connection with or without the intervention of the root. Group addressing allows a group of end systems to be identified. Anycast capability allows the setting up of a connection to an end system that is part of an ATM group. The reader is referred to ATM Forum UNI 4.0 signaling specification for details on these capabilities.

Table 9.6 shows a few of the messages used by UNI 4.0 and Q.2931 and their significance when sent by the host or the network. Each signaling message con-

| Number | Capability | Terminal equipment | Switching system |
|---|---|---|---|
| 1 | Point-to-point calls | M | M |
| 2 | Point-to-multipoint calls | O | M |
| 3 | Signaling of individual QoS parameters | O | M |
| 4 | Leaf-initiated join | M | M |
| 5 | ATM anycast | O | O |
| 6 | ABR signaling for point-to-point calls | O | (1) |
| 7 | Generic identifier transport | O | O |
| 8 | Virtual UNIs | O | O |
| 9 | Switched virtual path (VP) service | O | O |
| 10 | Proxy signaling | O | O |
| 11 | Frame discard | O | O (2) |
| 12 | Traffic parameter negotiation | O | O |
| 13 | Supplementary services | – | – |
| 13.1 | Direct dialing in (DDI) | O | O |
| 13.2 | Multiple subscriber number (MSN) | O | O |
| 13.3 | Calling line identification presentation (CLIP) | O | O |
| 13.4 | Calling line identification restriction (CLIR) | O | O |
| 13.5 | Connected identification presentation (COLP) | O | O |
| 13.6 | Connected line identification restriction (COLR) | O | O |
| 13.7 | Subaddressing (SUB) | O | (3) |
| 13.8 | User-user signaling (UUS) | O | O |

*Notes*: [1]This capability is optional for public networks/switching systems and is mandatory for private networks/switching systems.
[2]Transport of the frame discard indication is mandatory.
[3]This capability is mandatory for network/switching systems (public and private) that support only native E.164 address formats.

**TABLE 9.5.** Capabilities of UNI 4.0

| Message | Meaning (when sent by host) | Meaning (when sent by network) |
|---|---|---|
| SETUP | Requests that a call be established | Indicates an incoming call |
| CALL PROCEEDING | Acknowledges the incoming call | Indicates the call request will be attempted |
| CONNECT | Indicates acceptance of the call | Indicates the call was accepted |
| CONNECT ACK | Acknowledges acceptance of the call | Acknowledges making the call |
| RELEASE | Requests that the call be terminated | Terminates the call |
| RELEASE COMPLETE | Acknowledges releasing the call | Acknowledges releasing the call |

**TABLE 9.6.** Signaling messages involved in connection setup

tains a *call reference* that serves as a local identifier for the connection at the UNI. Each message also contains a number of information elements. These include obvious parameters such as calling and called party numbers, AAL parameters, ATM traffic descriptor and QoS parameters, and connection identifier. Many other mandatory and optional parameters are defined.

The signaling procedures specify the sequence of message exchanges to establish and release connections. They also address the handling of many error conditions that can arise. In Figure 9.23, we consider the simple case of the establishment of a point-to-point virtual connection using Q.2931.

1. Host A sends a SETUP message on VPI/VCI = 0/5 identifying the destination (host B) and other parameters specifying details of the requested connection.

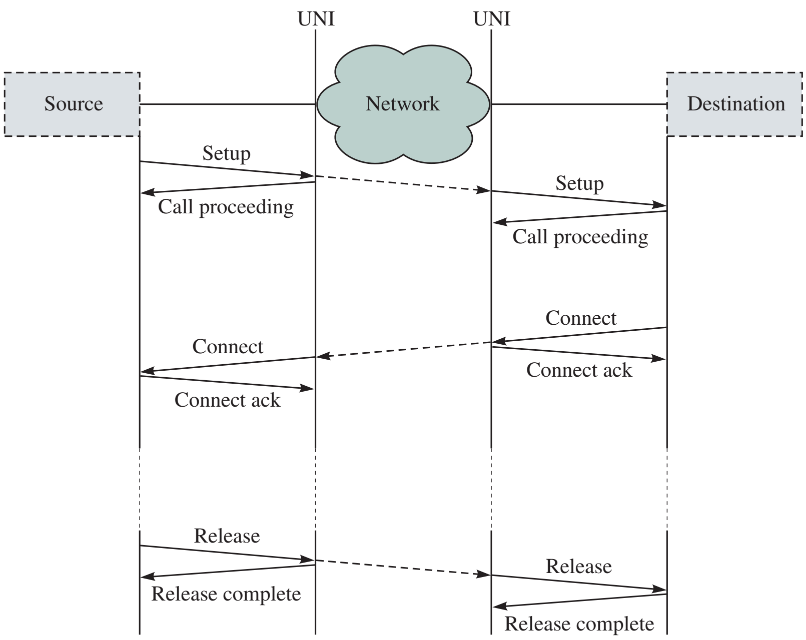


**FIGURE 9.23** UNI signaling example

2. The first switch analyzes the contents of the SETUP message to see whether it can handle the requested connection. If the switch can handle the request, the network returns a CALL PROCEEDING message to the host containing the VPI/VCI for the first link. It also forwards the SETUP message across the network to the destination.
3. Upon arrival of the SETUP message, the destination sends a CALL PROCEEDING message.
4. If the destination accepts the call, it sends a CONNECT message that is forwarded across the network back to host A. The CONNECT messages trigger CONNECT ACKNOWLEDGE messages from the network and eventually from the source.
5. The connection is now established, and the source and destination can exchange cells in the bidirectional VC that has been established.
6. Either party can subsequently initiate the termination of the call by issuing a RELEASE message. This step will trigger RELEASE COMPLETE messages from the network and from the other party.

A point-to-multipoint connection is established as follows. The root of the connection begins by establishing a connection to the first destination (leaf) by using the above procedure. It then issues ADD PARTY messages that attach additional destinations (leaves) to the connection. Note that point-to-multipoint connections are unidirectional, so cells can flow only from the root to the leaves.

### 9.5.3 PNNI Signaling

The ATM Forum has developed the PNNI specification for use between private ATM switches (private network node interface) and between groups of private ATM switches (private network-to-network interface) as shown in Figure 9.24. The PNNI specification includes two types of protocols.

1. A routing protocol that provides for the selection of routes that can meet QoS requirements (this routing protocol is discussed in the next section).

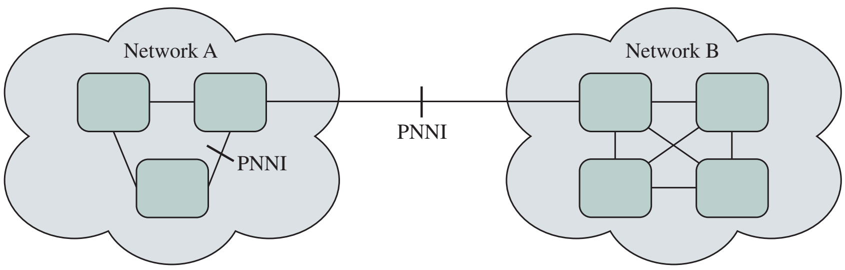


**FIGURE 9.24** PNNI contexts

2. A complementary signaling protocol for the exchange of messages between switches and between private networks. In this section we consider the signaling protocol.

The PNNI signaling protocol provides for the establishment and release of point-to-point as well as point-to-multipoint connections. The protocol is based on UNI 4.0 with extensions to provide support for source routing, for crankback (a feature of the routing protocol), and for alternate routing of connection requests in the case of connection setup failure. UNI signaling is asymmetric in that it involves a user and a network. PNNI modifies UNI signaling to make it symmetric. It also includes modifications in the information elements to carry routing information.

PNNI uses source routing where the first switch selects the route to the destination. In Figure 9.25 the source host requests a connection to host B by sending a SETUP message, using UNI signaling. The first switch carries out the connection admission control (CAC) function and returns a CALL PROCEEDING message if it can handle the connection request. The first switch maintains and uses a topology database to calculate a route to the destination that can meet the requirements of the connection contract.[5] The route consists of a vector of switches that are to be traversed. The SETUP message propagates

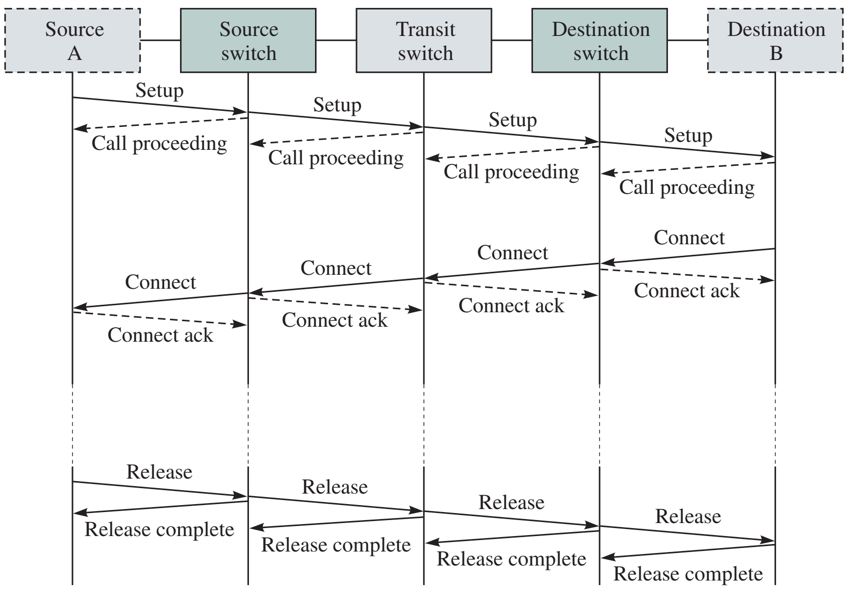


**FIGURE 9.25** PNNI signaling example

[5]The PNNI routing protocol includes procedures for distributing the link-state information required to determine the routes that can meet specific QoS requirements.

across the network, using the source route. Each switch along the path performs CAC and forwards the SETUP message along the next hop if it can handle the connection request. It also issues a CALL PROCEEDING message to the preceding switch along the route. If the destination accepts the call, a connect message is returned across the network to the source. Connection release proceeds in similar fashion as shown in Figure 9.25.

The PNNI routing protocol introduces hierarchy in the ATM network that provides a switch with detailed routing information in its immediate vicinity and only summary information about distant destinations. The signaling protocol is somewhat more complicated than the preceding example because of this hierarchical feature.

## 9.6 PNNI ROUTING

Routing in ATM networks is a subject that has not been investigated as much as other ATM areas such as congestion control and switching. The most visible result on ATM routing comes from the ATM Forum standards work on the **private network-to-network interface (PNNI[6])**. In this section, we briefly look at the main concepts behind PNNI routing techniques. Unlike Internet routing, which is divided into intradomain and interdomain routing protocols, PNNI works for both cases.

PNNI adopts the link-state philosophy in that each node would know the topology of the network. However, PNNI adds a new twist to make the routing protocol scalable so that it can work well for small networks as well for large networks with thousands of nodes. This goal is achieved by constructing a **routing hierarchy**, as illustrated in Figure 9.26.

As shown in the figure, a *peer group* (PG) is a collection of nodes (physical or logical) where each maintain an identical view of the group. For example, peer group A.1 consists of three nodes A.1.1, A.1.2, and A.1.3. A PG is abstractly represented in a higher-level routing hierarchy as a *logical group node* (LGN). For example, LGN B represents PG B in the next hierarchy. Note that the definition of PG and LGN is recursive. At the lowest level a PG is a collection of physical switches connected by physical links. At higher levels a PG is a collection of LGNs connected by logical links. Each PG contains a *peer group leader* (PGL) that actually executes the functions of the logical group nodes for that PG. A PGL summarizes the topological information within the PG and injects this information into the higher-order group. A PGL also passes down summarized topological information to its PG. The advantage of using this hierarchical structure is that each switch maintains only a partial view of the entire network topology, thereby reducing the amount of routing information kept at each switch. For example, from the point of view of switch A.1.1, the

[6]PNNI also stands for private network node interface.

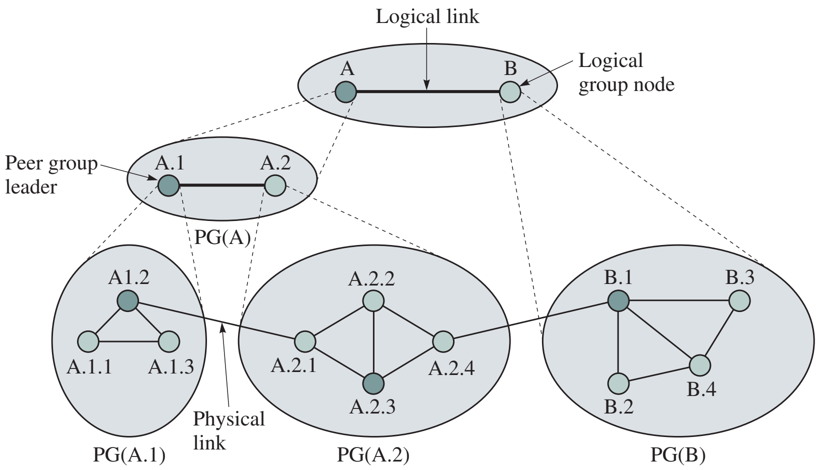


**FIGURE 9.26** Example of PNNI hierarchy

topology of the network is shown in Figure 9.27, which is much simpler than that of the entire topology.

PNNI uses source routing to set up connections. First, the source node specifies the entire path across its peer group, which is described by a **designated transit list (DTL)**. Suppose that a station attached to switch A.1.1 requests a connection setup to another station attached to switch B.3. After the source station requests a connection setup, switch A.1.1 chooses the path to be (A.1.1, A.1.2, A.2, B). In this case three DTLs organized in a stack will be built by A.1.1 in the call setup:

DTL: [A.1.1, A.1.2] pointer-2
DTL: [A.1, A.2] pointer-1
DTL: [A, B] pointer-1

The current transit pointer specifies which node in the list is currently being visited at that level, except at the top TDL where the transit pointer specifies the node to be visited next. Thus the top pointer points to A.1.2 and the other pointers point to A.1 and A respectively.

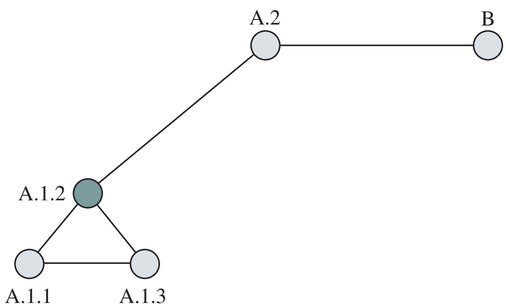


**FIGURE 9.27** Topology seen by switch A.1.1

When A.1.2 receives the call setup message, the switch realizes that the entry in the top of the stack is exhausted. After removing the top DTL, A.1.2 finds that the next entry is A.2, which is also its immediate neighbor. The DTLs become

DTL: [A.1, A.2] pointer-2
DTL: [A, B] pointer-1

When A.2.1 receives the call setup message, the switch finds that the target has been reached, since A.2.1 is in A.2. So A.2.1 builds a route to B, say, through A.2.3 and A.2.4, and pushes a new DTL onto the stack:

DTL: [A.2.1, A.2.3, A.2.4] pointer-2
DTL: [A.1, A.2] pointer-2
DTL: [A, B] pointer-1

When A.2.3 receives the call setup message, the switch advances the current transit pointer and forwards the message to A.2.4. When A.2.4 receives the call setup message, it finds that the targets at the top two DTLs have been reached. So A.2.4 removes the top two DTLs and forwards the message with the following DTL to its neighbor:

DTL: [A, B] pointer-2

When B.1 receives the call setup message, B.1 finds that the curernt DTL has been reached. B.1 builds a new DTL, giving

DTL: [B.1, B.3] pointer-2
DTL: [A, B] pointer-2

When the message reaches its destination, B.3 determines that it is a DTL terminator, since all DTLs are at the end and B.3 is the lowest-level node.

It is possible that a call setup will be blocked when the requested resources at a switch are not available. PNNI provides **crankback** and **alternate routing**. When a call is blocked at a particular DTL, it is cranked back to the creator of the DTL.

A switch determines the path based on the connection's traffic descriptor, QoS requirements, and the resources stored in its database. Because CAC is not standardized, PNNI uses a **generic connection admission control (GCAC)** to select a path that is likely to satisfy the connection's end-to-end traffic and QoS requirements. GCAC, however, only predicts the most likely path that can satisfy the connection traffic, since the information known by the switch may be outdated when the actual call request is made at other switches along the path.

GCAC requires the following parameters:

- Available cell rate (ACR): A measure of available bandwidth on the link.
- Cell rate margin (CRM): A measure of the difference between the aggregate allocated bandwidth and sustained rate of the existing connections.
- Variance factor (VF): A relative measure of the CRM normalized by the variance of the aggregate rate.

When a new connection with peak cell rate PCR and sustained cell rate SCR requests a connection setup, the GCAC algorithm examines each link along the path and performs the following decision:

if PCR $\leq$ ACR
    include the link
else if SCR $>$ ACR
    exclude the link
else if [ACR $-$ SCR][ACR $-$ SCR $+$ 2 * CRM] $\geq$ VF * SCR (PCR $-$ SCR)
    include the link
else
    exclude the link

The VF is $CRM^2/VAR$, and the variance is given by

$$VAR = \sum_i SCR(i)[PCR(i) - SCR(i)]$$

After the path has been selected, each switch along the path eventually has to perform its own CAC to decide the acceptance/rejection of the connection request. Note that GCAC uses VF only in the case SCR $<$ ACR $<$ PCR.

**ATM IS DEAD, LONG LIVE ATM!**

The development of ATM standards was a massive effort that attempted to develop an entire future network architecture from the ground up. BISDN was viewed as a future multiservice network that could encompass LANs and WANs in a common framework. Alas, this was not to be.

The ATM connection-oriented networking paradigm emerged at the same time as the explosion of the World Wide Web. The Web is built on HTTP and the TCP/IP protocol suite which is decidedly connectionless in its network orientation. ATM to the user is not what was needed and so the vision of an end-to-end universal ATM network perished.

ATM technology does have advantages such as facilitating high-speed switching and enabling the management of traffic flows in the network. For this reason ATM has found wide application in carrier backbone networks and, to a lesser extent, in campus networks that require prioritization of traffic flows.

The long-term future of ATM in an all-IP world is uncertain. In Chapter 10 we examine the issues in operating IP over ATM. We discuss a number of initiatives that incorporate features of ATM in IP. In effect, it is possible that ATM may disappear over the long run, but that many of its innovations in high-speed switching, traffic management, and QoS will survive in an IP-networking framework.

## SUMMARY

In this chapter we examined the architecture of ATM networks. We discussed the BISDN reference model and the role of the control plane in setting up ATM connections. We then examined the structure of the ATM header and its relationship to virtual paths and virtual connections. The connection setup involves the establishment of a traffic contract between the user and the network that commits the user to a certain pattern of cell transmission requests and the network to a certain level of QoS support. We examined the traffic management mechanisms that allow ATM to provide QoS guarantees across a network. The various types of ATM service categories to meet various information transfer requirements were introduced.

We discussed the role of the ATM adaptation layer in providing support for a wide range of user applications. This support included segmentation and reassembly, reliable transmission, timing recovery and synchronization, and multiplexing.

Last, we introduced PNNI signaling and its associated routing protocol which are used to set up connections with QoS guarantees across ATM networks.

## CHECKLIST OF IMPORTANT TERMS

alternate routing
ATM adaptation layer (AAL)
ATM layer
broadband intercarrier interface (B-ICI)
common part (CSCP)
control plane
convergence sublayer (CS)
crankback
designated transit list (DTL)
generic connection admission control (GCAC)
message identifier (MID)
multiplexing identifier
network-network interface (NNI)
private network-to-network interface (PNNI)
routing hierarchy
segmentation and reassembly (SAR)
service specific part (SSCS)
user-network interface (UNI)

## FURTHER READING

ATM Forum Technical Committee, "ATM User-Network Interface (UNI) Signaling Specification, Version 4.0," af-sig-0061.000, July 1996.

ATM Forum Technical Committee, "Private Network-Network Interface Specification, Version 1.0," af-pnni-0055.000, March 1996.

ATM Forum Technical Committee, "Traffic Management Specification, Version 4.0," af-tm-0056.000, April 1996.

Black, U., *ATM: Foundation for Broadband Networks*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.

Dutton, H. J. R. and P. Lenhard, *Asynchronous Transfer Mode (ATM)*, Prentice Hall PTR, Upper Saddle River, New Jersey, 1995.

Helgert, H. J., *Integrated Services Digital Networks: Architectures, Protocols, Standards*, Addison-Wesley, Reading, Massachusetts, 1991.

Ibe, O. C., *Essentials of ATM Networks and Services*, Addison-Wesley, Reading, Massachusetts, 1997. Excellent concise introduction to ATM.

ITU draft, I.363.2, "B-ISDN ATM Adaptation Layer Type 2 Specification," November 1996.

Leon-Garcia, A., *Probability and Random Processes for Electrical Engineering*, Addison-Wesley, Reading, Massachusetts, 1994.

McDysan, D. E. and D. L. Spohn, *ATM: Theory and Application*, McGraw-Hill, New York, 1995.

Prycker, M. de, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.

## PROBLEMS

**1.** Suppose that instead of ATM, BISDN had adopted a transfer mode that would provide constant-bit-rate connections with bit rates given by integer multiples of 64 kbps connections. Comment on the multiplexing and switching procedures that would be required to provide this transfer mode. Can you give some reasons why BISDN did not adopt this transfer mode?

**2.** a. Compare the bandwidth management capabilities provided by ATM virtual paths to the capabilities provided by SONET networks.
   b. Can ATM virtual paths be adapted to provide the fault tolerance capabilities of SONET rings? Explain your answer.

**3.** In Chapter 6 we saw that the performance of MACs for LANs can depend strongly on the delay-bandwidth product of the LAN. Consider the use of ATM in a LAN. Does the performance depend on the delay-bandwidth product? If yes, explain how; if no explain why not? Does the same conclusion apply to any LAN that involves the switching packets, rather than the use of a broadcast medium?

**4.** Does the performance of ATM depend strongly on the delay-bandwidth product of a wide-area network? Specifically, consider the performance of ATM congestion control in a network with large delay-bandwidth product. Does the size of the buffers in the ATM switches influence the severity of the problem? Give an order of magnitude for the amount of buffering required in the switches.

**5.** Compare a conventional TDM leased line with an ATM PVC from the user's point of view and from the network operator's point of view. Which features of PVCs make them attractive from both points of view?

**6.** An inverse multiplexer is a device that takes as input a high-speed digital stream and divides it into several lower-speed streams that are then transmitted over parallel transmission lines to the same destination. Suppose that a very high speed ATM stream is to be sent over an inverse multiplexer. Explain which requirements must be met so that the inverse demultiplexer produces the original ATM stream.

**7.** Suppose an ATM switch has 32 input ports and 32 output ports.
a. Theoretically, how many connections can the switch support?
b. What are the table lookup requirements for supporting a large number of connections? Do they limit the practical size on the number of connections that can actually be supported?
c. (Optional) Do a Web search on content addressable memories (CAMs) and explain how these can help in addressing the table-lookup problem.

**8.** Explain how the header error checksum can be used to synchronize to the boundary of a sequence of contiguous ATM cells. What is the probability that an arbitrary five-octet block will satisfy the header error checksum? Assume bits are equally likely to be 0 or 1. What is the probability that two random five-octet blocks that correspond to two consecutive headers pass the error check?

**9.** We saw that ATM provides a GFC field in the UNI ATM header. Suppose that several terminals share a medium to access an ATM network.
a. Explain why flow control may be required to regulate the access of traffic from the terminals into the network. Explain how the GFC field can be used to do this task?
b. Explain how the GFC field can be used as a subaddress to provide point-to-multipoint access to an ATM network. Does this usage conflict with the flow control requirement?

**10.** The purpose of the header error control (HEC) field is to protect against errors in the header that may result in the misaddressing and misdelivery of cells. The CRC used in the ATM header can correct all single errors and can detect (but not correct) all double errors that occur in the header. Some, but not all, multiple errors in excess of two can also be detected.
a. Suppose that bit errors occur at random and that the bit error rate is $p$. Find the probability that the header contains no errors, a single error, a double error, more than two errors. Evaluate these probabilities for $p = 10^{-3}$, $10^{-6}$, $10^{-9}$.
b. Relate the calculations in part (a) to the CMR experienced in such an ATM network.
c. In practice the bit errors may not always be random, and so to protect against bursts of errors the following adaptive procedure may be used.

Normally the receiver is in the "correction" mode. If a header is error free, the receiver stays in this mode; if the receiver detects a single error, the receiver corrects the error and changes to "detection" mode; if the receiver detects a multiple error, the receiver changes to HEC error detection mode. If the receiver is in "detection" mode and one or more errors are detected in the header, then the cell is discarded and the receiver stays in the detection mode; if the header is error free, then the receiver changes to the "detection" mode.

Explain why this procedure protects against bursts of errors. If the bit errors are independent, what is the probability that two consecutive headers contain single errors and hence that a cell is discarded unnecessarily?

**11.** What is the difference between CER and CLR? Why is one negotiated during connection setup and the other is not?

**12.** Why does the calculation of CER and CMR exclude blocks that are counted in the SECBR?

**13.** Explain how weighted fair queueing scheduling can be used to affect the CLR and CTD experienced by cells in an ATM connection.

**14.** Explain the effect of a single cell loss from a long packet in a situation that uses an end-to-end ARQ retransmission protocol. Can you think of strategy to deal with such losses? Can you think of a way to ameliorate the impact of packet retransmission?

**15.** Consider a sequence of ATM cells carrying PCM voice from a single speaker.
a. What are the appropriate traffic descriptors for this sequence of cells, and what is an appropriate leaky bucket for policing this stream?
b. Suppose that an ATM connection is to carry the cell streams for *M* speakers. What are appropriate traffic descriptors for the resulting aggregate stream, and how can it be policed?

**16.** Consider a sequence of ATM cells carrying PCM voice from a single speaker, but suppose that silence suppression is used.
a. What are the appropriate traffic descriptors for this sequence of cells, and what is an appropriate leaky bucket(s) arrangement for policing this stream? Which situation leads to nonconforming cells?
b. Suppose that an ATM connection is to carry the cell streams for *M* speakers. What are appropriate traffic descriptors for the resulting aggregate stream, and how can it be policed?

**17.** Suppose that constant-length packets (of size equal to *M* cells) arrive at a source to be carried by an ATM connection and that such packets are separated by exponential random times *T*. What are the appropriate traffic descriptors for this sequence of cells, and what is an appropriate leaky bucket(s) arrangement for policing this stream? Which situation leads to nonconforming cells?

**18.** Explain why each specific set of traffic descriptors and QoS parameters were selected for each of the ATM service categories.

**19.** Suppose that IP packets use AAL5 prior to transmission over an ATM connection. Explain the transfer-delay properties of the IP packets if the ATM connection is of the following type: CBR, rt-VBR, nrt-VBR, ABR, or UBR.

**20.** Proponents of ATM argue that VBR connections provide a means of attaining multiplexing gains while providing QoS. Proponents of IP argue that connectionless IP routing can provide much higher multiplexing gains. Can you think of arguments to support each claim. Are these claims conflicting, or can they both be correct?

**21.** a. Consider a link that carries connections of individual voice calls using PCM. What information is required to perform call admission control on the link?

b. Now suppose that the link carries connections of individual voice calls using PCM but with silence suppression. What information is required to do call admission control?

**22.** Suppose that an ATM traffic stream contains cells of two priorities, that is, high-priority cells with CLP = 0 in the headers and low-priority cells with CLP = 1.

a. Suppose we wish to police the peak cell rate of the CLP = 0 traffic to $p_0$ as well as the peak rate of the combined CLP = 0 and CLP = 1 traffic to $P_{0+1}$. Give an arrangement of two leaky buckets to do this policing. Nonconforming cells are dropped.

b. Compare the following policing schemes: (1) police CLP = 0 traffic to peak rate $p_0$ and police CLP = 1 traffic to peak rate $p_1$, (2) police the combined CLP = 0 and CLP = 1 traffic to peak rate $p_0 + p_1$. Which approach is more flexible?

c. Repeat part (a) if nonconforming CLP = 0 cells that do not conform to the $p_0$ are tagged by changing the CLP bit to 1. Cells that do not conform to $p_{0+1}$ are dropped.

**23.** Suppose that an ATM traffic stream contains cells of two priorities, that is, high-priority cells with CLP = 0 in the headers and low-priority cells with CLP = 1.

a. Suppose we wish to police the sustainable cell rate of the CLP = 0 traffic to $SCR_0$ and BT as well as the peak rate of the combined CLP = 0 and CLP = 1 traffic to $p_{0+1}$. Give an arrangement of two leaky buckets to do this policing. Nonconforming cells are dropped.

b. Repeat part (a) if cells that do not conform to $SCR_0$ and BT are tagged by changing the CLP bit to 1. Cells that do not conform to $p_{0+1}$ are dropped.

**24.** Suppose that an ATM traffic stream contains cells of two priorities, that is, high-priority cells with CLP = 0 in the headers and low-priority cells with CLP = 1. Suppose we wish to police the peak cell rate and the sustainable cell rate of the combined CLP = 0 and CLP = 1 traffic. Give an arrangement of two leaky buckets to do this policing. Nonconforming cells are dropped.

**25.** Explain how weighted fair queueing might be used to combine the five ATM service categories onto a single ATM transmission link. How are the different service categories affected as congestion on the link increases?

**26.** a. Discuss what is involved in calculating the end-to-end CLR, CTD, and CDV.

b. Compare the following two approaches to allocating the end-to-end QoS to per link QoS: equal allocation to each link; unequal allocation to various links. Which is more flexible? Which is more complex?

**27.** Suppose that an application uses the reliable stream service of TCP that in turns uses IP over ATM over AAL5.

a. Compare the performance seen by the application if the AAL5 uses CBR connection; nrt-VBR connection; ABR connection; UBR connection.

b. Discuss the effect on the peformance seen by the application if the ATM connection encounters congestion somewhere in the network.

**28.** Suppose that an ATM connection carries voice over AAL1. Suppose that the packetization delay is to be kept below 10 ms.

a. Calculate the percentage of overhead if the voice is encoded using PCM.

b. Calculate the percentage of overhead if the voice is encoded using a 12 kbps speech-encoding scheme from cellular telephony.

**29.** Explain how the three-bit sequence number in the AAL1 header can be used to deal with lost cells and with misinserted cells.

**30.** How much delay is introduced by the two interleaving techniques that can be used in AAL1?

**31.** a. How many low-bit-rate calls can be supported by AAL2 in a single ATM connection?
b. Estimate the bit rate of the connection if an AAL2 carries the maximum number of calls carrying voice at 12 kbps.
c. What is the percentage of overhead in part (b)?

**32.** Compare the overhead of AAL3/4 with that of AAL5 for a 64K byte packet.

**33.** Discuss the purpose of all the error checking that is carried out at the end systems and in the network for an ATM connection that carries cells produced by AAL3/4. Repeat for AAL5.

**34.** Suppose that in Figure 9.17 packets from A and B arrive simultaneously and each produces 10 cells. Show the sequence of SPDUs produced including segment type, sequence number, and multiplexing ID.

**35.** Consider SSCOP, the AAL protocol for signaling. Discuss the operation of the Selective Repeat ARQ procedure to recover from cell losses. In particular, discuss how the protocol differs from the Selective Repeat ARQ protocol introduced in Chapter 5.

**36.** Can the SSCOP AAL protocol be modified to provide the same reliable stream service that is provided by TCP? If yes, explain how; if no, explain why not.

**37.** The "cells in frames" proposal attempts to implement ATM on workstations attached to a switched Ethernet LAN. The workstation implements the ATM protocol stack to produce cells, but the cells are transmitted using Ethernet frames as follows. The payload of the Ethernet frame consists of a four-byte CIF header, followed by a single ATM header, and up to 31 ATM cell payloads.
a. Find the percentage of overhead of this approach and compare it to standard ATM.
b. The workstation implements ATM signaling, and the NIC driver is modified to handle several queues to provide QoS. Discuss the changes required in the Ethernet switch so that it can connect directly to an ATM switch.

**38.** Compare the size of the address spaces provided by E-164 addressing, AESA addressing, IPv4 addressing, IPV6 addressing, and IEEE 802 MAC addressing.

**39.** Can IP addresses be used in ATM? Explain why or why not?

**40.** Identify the components that contribute to the end-to-end delay experienced in setting up an ATM connection using PNNI.

**41.** Describe the sequence of DTLs that are used in setting up a connection from A.1.3 to A.2.2 in Figure 9.26. Repeat for a connection from B.4 to A.1.2.

**42.** a. Discuss the differences and similarities between PNNI and OSPF.
b. Can PNNI be modified to provide QoS routing in the Internet? Explain why or why not?

**43.** Compare the hierarchical features of the combination of BGP4 and OSPF with the features of PNNI.

**44.** Which aspects of the ATM network architecture depend on the fixed-length nature of ATM cells? What happens if ATM cells are allowed to be variable in length?

**45.** Which aspects of ATM traffic management change if ATM connections must belong to one of a limited number of classes and if QoS is guaranteed not to individual connections but the class as a whole? Can VPs play a role in providing QoS to these classes?

**46.** Explain how the ATM architecture facilitates the creation of multiple virtual networks that coexist over the same physical ATM network infrastructure but that can be operated as if they were separate independent networks. Explain how such virtual networks can be created and terminated on demand.