18.2 FRAME RELAY

Frame Relay is a virtual circuit wide area network that was designed to respond to demands for a new type of WAN in the late 1980s and early 1990s.

1. Prior to Frame Relay, some organizations were using a virtual circuit switching network called **X.25** that performed switching at the network layer. For example, the Internet, which needs wide area networks to carry its packets from one place to another, used X.25. X.25 is still being used by the Internet, but it is being replaced by other WANs. However, X.25 has several drawbacks:
   a. X.25 has a low 64-Kbps data rate. By the 1990s, there was a need for higher-data-rate WANs.
   b. X.25 has extensive flow and error control at both the data link layer and the network layer. This was so because X.25 was designed in 1970s, when the available transmission media were more prone to errors. Flow and error control at both layers create a large overhead and slow down transmissions. X.25 requires acknowledgments for both data link layer frames and network layer packets that are sent between nodes and between source and destination.
   c. Originally X.25 was designed for private use, not for the Internet. X.25 has its own network layer. This means that the user's data are encapsulated in the network-layer packets of X.25. The Internet, however, has its own network layer, which means if the Internet wants to use X.25, the Internet must deliver its network-layer packet, called a datagram, to X.25 for encapsulation in the X.25 packet. This doubles the overhead.
2. Disappointed with X.25, some organizations started their own private WAN by leasing T-1 or T-3 lines from public service providers. This approach also has some drawbacks.
   a. If an organization has n branches spread over an area, it needs $n(n-1)/2$ T-1 or T-3 lines. The organization pays for all these lines although it may use the lines only 10 percent of the time. This can be very costly.
   b. The services provided by T-1 and T-3 lines assume that the user has fixed-rate data all the time. For example, a T-1 line is designed for a user who wants to use the line at a consistent 1.544 Mbps. This type of service is not suitable for the many users today that need to send **bursty data.** For example, a user may want to send data at 6 Mbps for 2 s, 0 Mbps (nothing) for 7 s, and 3.44 Mbps for 1 s for a total of 15.44 Mbits during a period of 10 s. Although the average data rate is still 1.544 Mbps, the T-1 line cannot accept this type of demand because it is designed for fixed-rate data, not bursty data. Bursty data requires what is called **bandwidth on demand.** The user needs different bandwidth allocations at different times.

In response to the above drawbacks, Frame Relay was designed. Frame Relay is a wide area network with the following features:

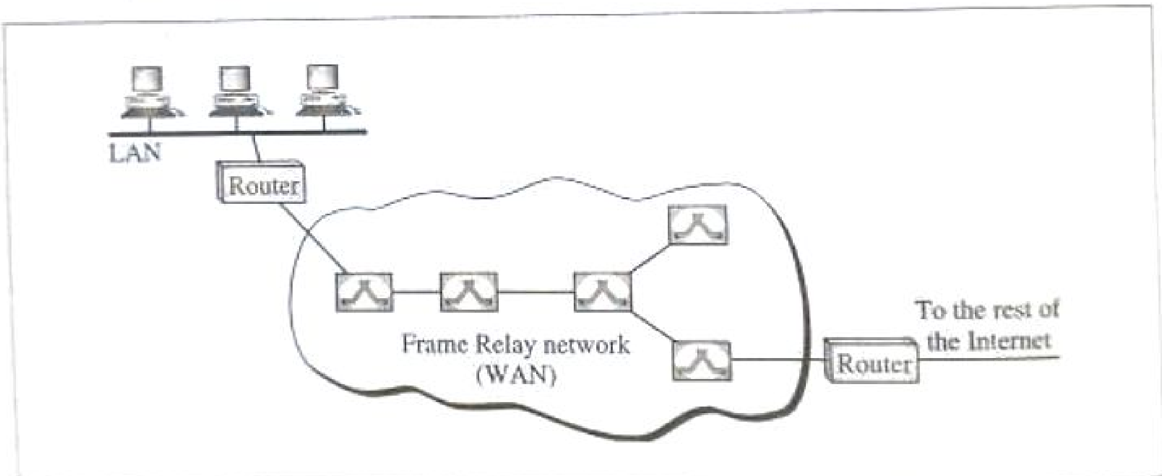
1. Frame Relay operates at a higher speed (1.544 Mbps and recently 44.376 Mbps). This means that it can easily be used instead of a mesh of T-1 or T-3 lines.
2. Frame Relay operates in just the physical and data link layers. This means it can easily be used as a backbone network to provide services to protocols that already have a network layer protocol, such as the Internet.

3. Frame Relay allows bursty data.
4. Frame Relay allows a frame size of 9000 bytes, which can accommodate all local area network frame sizes.
5. Frame Relay is less expensive than other traditional WANs.
6. Frame Relay has error detection at the data link layer only. There is no flow control or error control. There is not even a retransmission policy if a frame is damaged; it is silently dropped. Frame Relay was designed in this way to provide fast transmission capability for more reliable media and for those protocols that have flow and error control at the higher layers.

Architecture

Frame Relay provides permanent virtual circuits and switched virtual circuits. Figure 18.8 shows an example of a Frame Relay network connected to the Internet. The routers are used, as we will see in Chapter 19, to connect LANs and WANs in the Internet. In the figure, Frame Relay WAN is used as one link in the global Internet.

Figure 18.8 *Frame Relay network*



Virtual Circuits

Frame Relay is a virtual circuit network. A virtual circuit in Frame Relay is identified by a number called a **data link connection identifier (DLCI).** Frame Relay uses both PVCs and SVCs.

> VCIs in Frame Relay are called DLCIs.

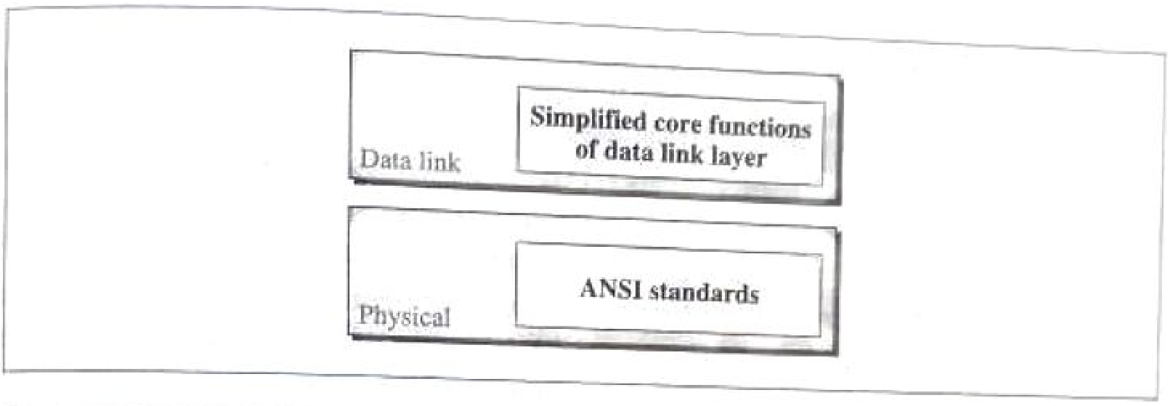
Switches

Each switch in a Frame Relay network has a table to route frames. The table matches an incoming port–DLCI combination with an outgoing port–DLCI combination as we described for general virtual circuit networks. The only difference is that VCIs are replaced by DLCIs.

Frame Relay Layers

Figure 18.9 shows the Frame Relay layers. Frame Relay has only physical and data link layers.

Figure 18.9 *Frame Relay layers*



Frame Relay operates only at the physical and data link layers.

Physical Layer

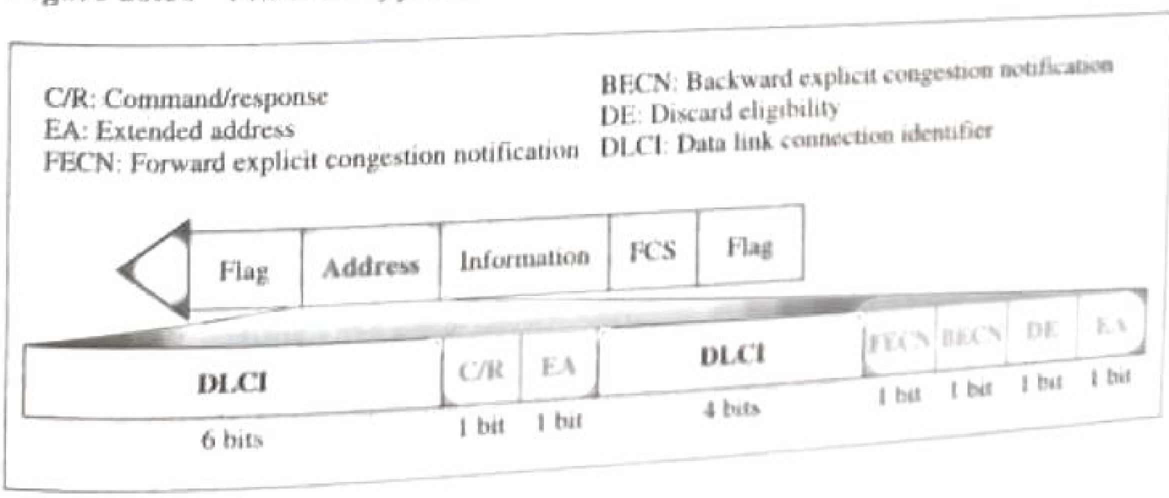
No specific protocol is defined for the physical layer in Frame Relay. Instead, it is left to the implementer to use whatever is available. Frame Relay supports any of the protocols recognized by ANSI.

Data Link Layer

At the data link layer, Frame Relay employs a simplified version of HDLC. The simpler version is used because HDLC provides extensive error and flow control fields that are not needed in Frame Relay.

Figure 18.10 shows the format of a Frame Relay frame. The frame is similar to that of HDLC. In fact, the flag, FCS, and information fields are the same. However, the control field is missing because this field was needed for flow and error control which is not provided by Frame Relay. The address field defines the DLCI as well as some bits used to control congestion and traffic.

Figure 18.10 *Frame Relay frame*



The descriptions of the fields are as follows:

- **Address (DLCI) field.** The first 6 bits of the first byte make up part 1 of the DLCI. The second part of the DLCI uses the first 4 bits of the second byte. These bits are part of the 10-bit data link connection identifier defined by the standard. The function of the DLCI was discussed previously. We will discuss extended addressing at the end of this section.
- **Command/response (C/R).** The command/response (C/R) bit is provided to allow upper layers to identify a frame as either a command or a response. It is not used by the Frame Relay protocol.
- **Extended address (EA).** The extended address (EA) bit indicates whether the current byte is the final byte of the address. An EA of 0 means that another address byte is to follow. An EA of 1 means that the current byte is the final one.
- **Forward explicit congestion notification (FECN).** The **forward explicit congestion notification (FECN)** bit can be set by any switch to indicate that traffic is congested in the direction in which the frame is traveling. This bit informs the destination that congestion has occurred. We will discuss the use of this bit when we discuss congestion control in Chapter 23.
- **Backward explicit congestion notification (BECN).** The **backward explicit congestion notification (BECN)** bit is set to indicate a congestion problem in the direction opposite to the one in which the frame is traveling. This bit informs the sender that congestion has occurred. We will discuss the use of this bit when we discuss congestion control in Chapter 23.
- **Discard eligibility (DE).** The **discard eligibility (DE)** bit indicates the priority level of the frame. In emergency situations, switches may have to discard frames to relieve bottlenecks and keep the network from collapsing due to overload. When set (DE 1), this bit tells the network to discard this frame if there is congestion. This bit can be set either by the sender of the frames (user) or by any switch in the network. We will discuss the use of this bit when we discuss congestion control in Chapter 23.

Frame Relay does not provide flow or error control; they must be provided by the upper-layer protocols.

Extended Address

To increase the range of DLCIs, the Frame Relay address has been extended from the original 2-byte address to 3- or 4-byte addresses. Figure 18.11 shows the different addresses. Note that the EA field defines the number of bytes; it is 1 in the last byte of the address, and it is 0 in the other bytes. Note that in the 3- and 4-byte formats, the bit before the last bit is set to 0.

FRADs

To handle frames arriving from other protocols, Frame Relay uses a device called a **Frame Relay assembler/disassembler (FRAD).** A FRAD assembles and disassembles frames coming from other protocols to allow them to be carried by Frame Relay frames

Figure 18.11 *Three address formats*

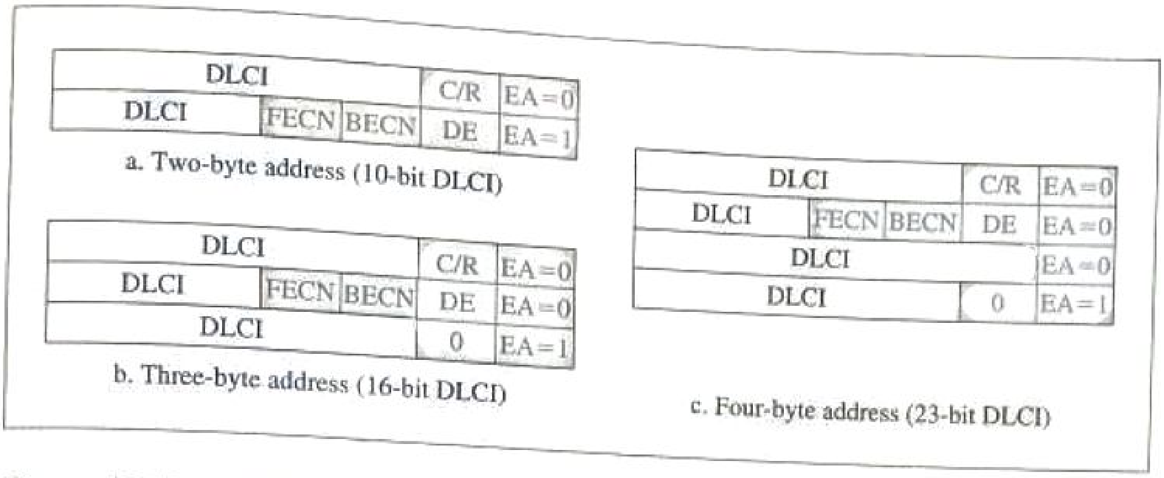
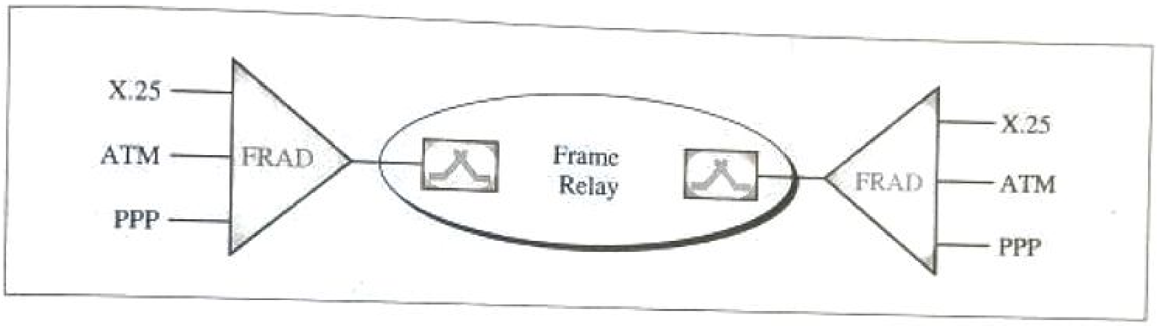


Figure 18.12 *FRAD*



A FRAD can be implemented as a separate device or as part of a switch. Figure 18.12 shows two FRADs connected to a Frame Relay network.

VOFR

Frame Relay networks offer an option called **Voice Over Frame Relay (VOFR)** that sends voice through the network. Voice is digitized using PCM and then compressed. The result is sent as data frames over the network. This feature allows the inexpensive sending of voice over long distances. However, note that the quality of voice is not as good as voice over a circuit-switched network such as the telephone network. Also, the varying delay mentioned earlier sometimes corrupts real-time voice.

LMI

Frame Relay was originally designed to provide PVC connections. There was not, therefore, a provision for controlling or managing interfaces. **Local management information (LMI)** is a protocol added recently to the Frame Relay protocol to provide more management features. In particular, LMI can provide

- A keep-alive mechanism to check if data are flowing.
- A multicast mechanism to allow a local end system to send frames to more than one remote end system.
- A mechanism to allow an end system to check the status of a switch (e.g., to see if the switch is congested).

Congestion Control and Quality of Service

One of the nice features of Frame Relay is that it provides **congestion control** and **quality of service.** We have not discussed these features yet. In Chapter 23, we introduce these two important aspects of networking and discuss how they are implemented in Frame Relay and some other networks.

18.3 ATM

Asynchronous Transfer Mode (ATM) is the **cell relay** protocol designed by the ATM Forum and adopted by the ITU-T. The combination of ATM and SONET will allow high-speed interconnection of all the world's networks. In fact, ATM can be thought of as the "highway" of the information superhighway.

Design Goals

Among the challenges faced by the designers of ATM, six stand out.

1. Foremost is the need for a transmission system to optimize the use of high-data-rate transmission media, in particular optical fiber. In addition to offering large bandwidths, newer transmission media and equipment are dramatically less susceptible to noise degradation. A technology is needed to take advantage of both factors and thereby maximize data rates.
2. The system must interface with existing systems and provide wide area interconnectivity between them without lowering their effectiveness or requiring their replacement.
3. The design must be implemented inexpensively so that cost would not be a barrier to adoption. If ATM is to become the backbone of international communications, as intended, it must be available at low cost to every user who wants it.
4. The new system must be able to work with and support the existing telecommunications hierarchies (local loops, local providers, long-distance carriers, and so on).
5. The new system must be connection-oriented to ensure accurate and predictable delivery.
6. Last but not least, one objective is to move as many of the functions to hardware as possible (for speed) and eliminate as many software functions as possible (again for speed).

Problems

Before we discuss the solutions to these design requirements, it is useful to examine some of the problems associated with existing systems.

Frame Networks

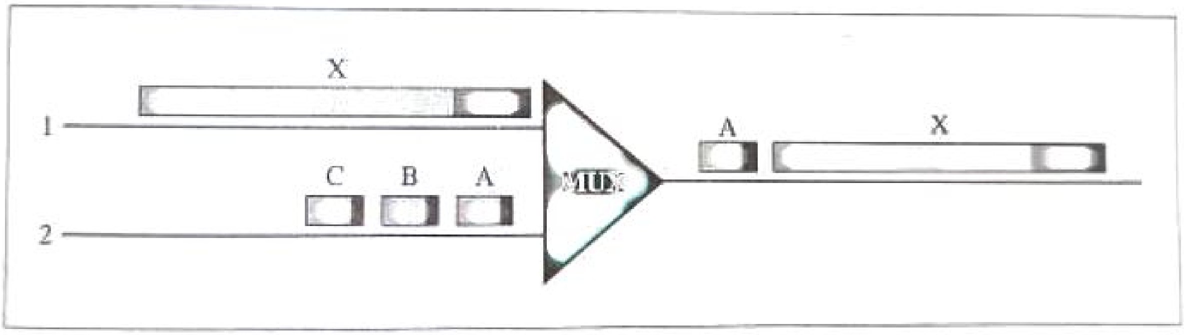
Before ATM, data communications at the data link layer had been based on frame switching and frame networks. Different protocols use frames of varying size and intricacy. As networks become more complex, the information that must be carried in the header becomes more extensive. The result is larger and larger headers relative to the size of the data unit. In response, some protocols have enlarged the size of the data unit to make header use more efficient (sending more data with the same size header). Unfortunately, large data fields create waste. If there is not much information to transmit, much of the field goes unused. To improve utilization, some protocols provide variable frame sizes to users.

Mixed Network Traffic

As you can imagine, the variety of frame sizes makes traffic unpredictable. Switches, multiplexers, and routers must incorporate elaborate software systems to manage the various sizes of frames. A great deal of header information must be read, and each bit counted and evaluated to ensure the integrity of every frame. Internetworking among the different frame networks is slow and expensive at best, and impossible at worst.

Another problem is that of providing consistent data rate delivery when frame sizes are unpredictable and can vary so dramatically. To get the most out of broadband technology, traffic must be time-division-multiplexed onto shared paths. Imagine the results of multiplexing frames from two networks with different requirements (and frame designs) onto one link (see Fig. 18.13). What happens when line 1 uses large frames (usually data frames) while line 2 uses very small frames (the norm for audio and video information)?

Figure 18.13 *Multiplexing using different frame sizes*



If line 1's gigantic frame X arrives at the multiplexer even a moment earlier than line 2's frames, the multiplexer puts frame X onto the new path first. After all, even if line 2's frames have priority, the multiplexer has no way of knowing to wait for them and processes the frame that has arrived. Frame A must therefore wait for the entire X bit stream to move into place before it can follow. The sheer size of X creates an unfair delay for frame A. The same imbalance can affect all the frames from line 2.

Because audio and video frames ordinarily are small, mixing them with conventional data traffic often creates unacceptable delays of this type and makes shared frame links unusable for audio and video information. Traffic must travel over different paths, in much the same way that automobile and train traffic does. But to fully utilize broad bandwidth links, we need to be able to send all kinds of traffic over the same links.

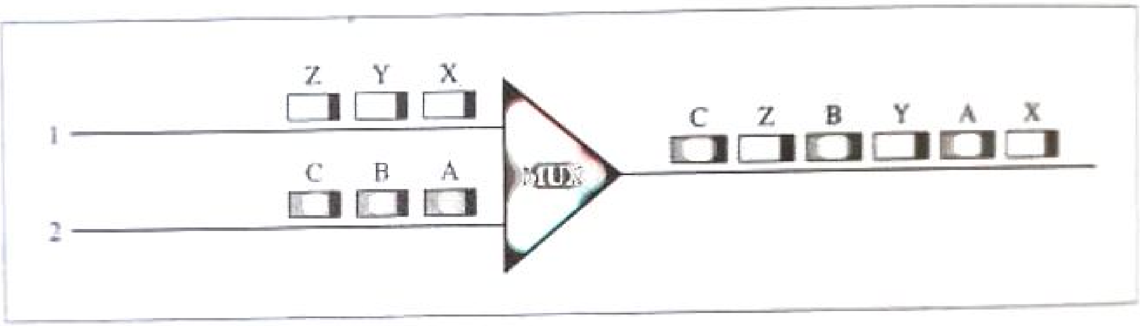
Cell Networks

Many of the problems associated with frame internetworking are solved by adopting a concept called cell networking. A cell is a small data unit of fixed size. In a **cell network,** which uses the **cell** as the basic unit of data exchange, all data are loaded into identical cells that can be transmitted with complete predictability and uniformity. As frames of different sizes and formats reach the cell network from a tributary network, they are split into multiple small data units of equal length and are loaded into cells. The cells are then multiplexed with other cells and routed through the cell network. Because each cell is the same size and all are small, the problems associated with multiplexing different-sized frames are avoided.

A cell network uses the cell as the basic unit of data exchange. A cell is defined as a small, fixed-sized block of information.

Figure 18.14 shows the multiplexer from Figure 18.13 with the two lines sending cells instead of frames. Frame X has been segmented into three cells: X, Y, and Z. Only the first cell from line 1 gets put on the link before the first cell from line 2. The cells from the two lines are interleaved so that none suffers a long delay.

Figure 18.14 *Multiplexing using cells*



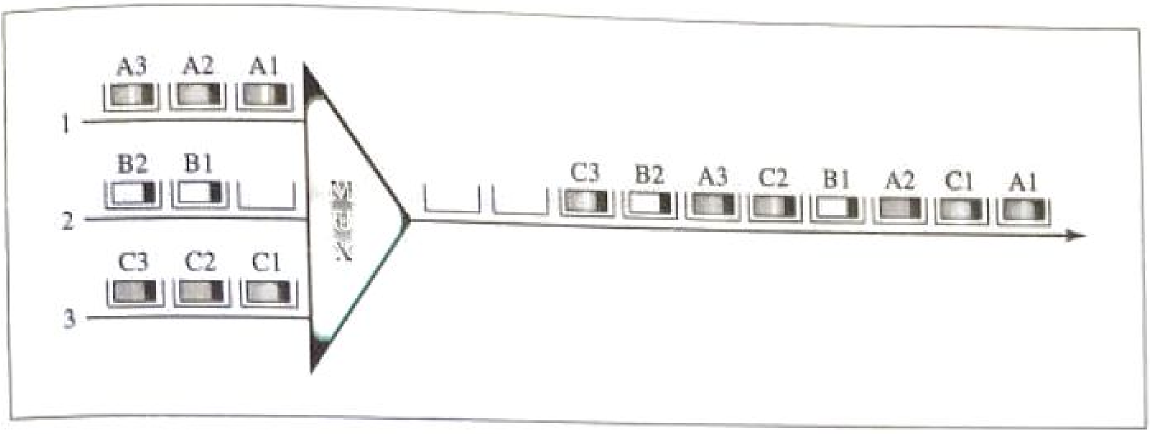
A second point in this same scenario is that the high speed of the links coupled with the small size of the cells means that, despite interleaving, cells from each line arrive at their respective destinations in an approximation of a continuous stream (much as a movie appears to your brain to be continuous action when in fact it is really a series of separate still photographs). In this way, a cell network can handle real-time transmissions, such as a phone call, without the parties being aware of the segmentation or multiplexing at all.

Asynchronous TDM

ATM uses asynchronous time-division multiplexing—that is why it is called Asynchronous Transfer Mode—to multiplex cells coming from different channels. It uses fixed-size slots (size of a cell). ATM multiplexers fill a slot with a cell from any input channel that has a cell; the slot is empty if none of the channels has a cell to send.

Figure 18.15 shows how cells from three inputs are multiplexed. At the first tick of the clock, channel 2 has no cell (empty input slot), so the multiplexer fills the slot with

Figure 18.15 *ATM multiplexing*

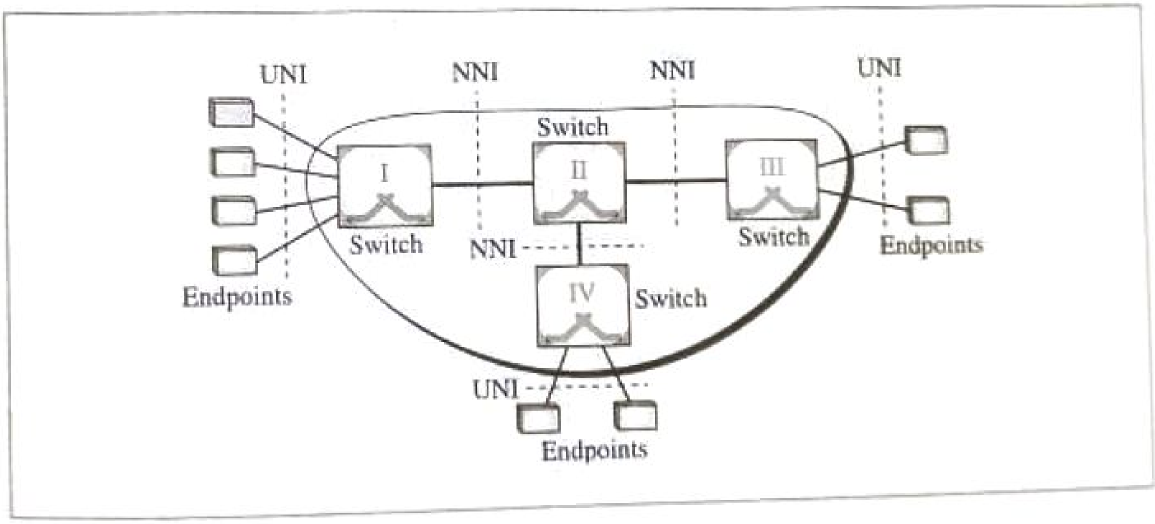


a cell from the third channel. When all the cells from all the channels are multiplexed, the output slots are empty.

Architecture

ATM is a cell-switched network. The user access devices, called the endpoints, are connected through a **user-to-network interface (UNI)** to the switches inside the network. The switches are connected through **network-to-network interfaces (NNIs).** Figure 18.16 shows an example of an ATM network.

Figure 18.16 *Architecture of an ATM network*



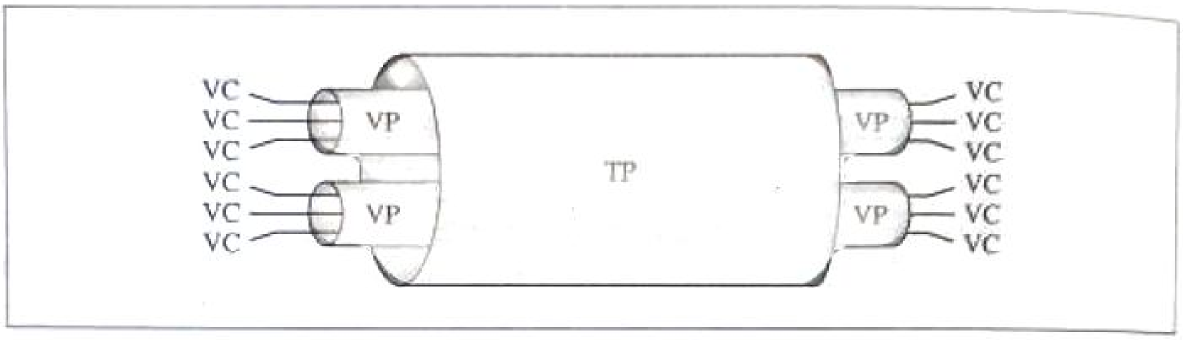
Virtual Connection

Connection between two endpoints is accomplished through transmission paths (TPs), virtual paths (VPs), and virtual circuits (VCs). A **transmission path (TP)** is the physical connection (wire, cable, satellite, and so on) between an endpoint and a switch or between two switches. Think of two switches as two cities. A transmission path is the set of all highways that directly connects the two cities.

A transmission path is divided into several virtual paths. A **virtual path (VP)** provides a connection or a set of connections between two switches. Think of a virtual path as a highway that connects two cities. Each highway is a virtual path; the set of all highways is the transmission path.

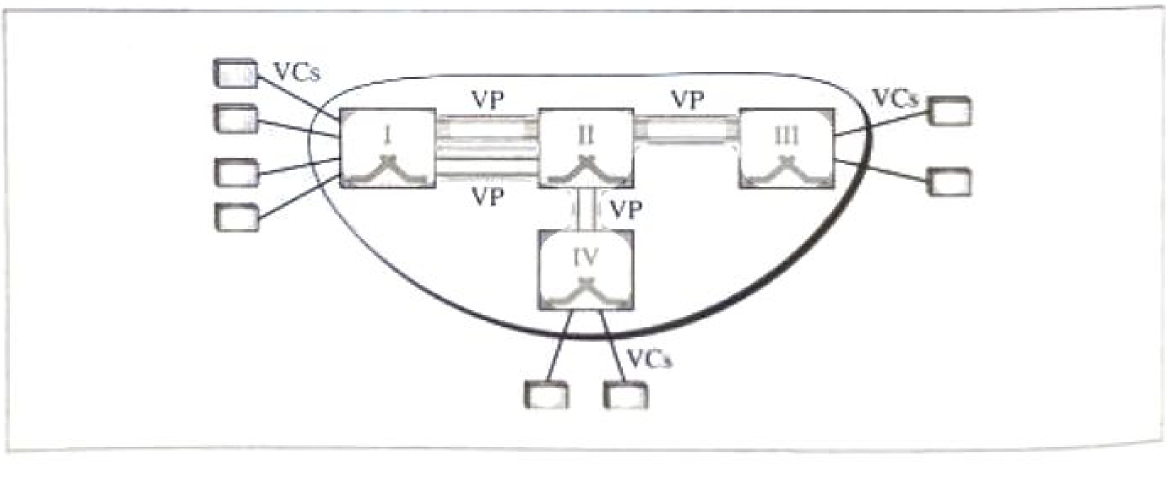
Cell networks are based on **virtual circuits (VCs).** All cells belonging to a single message follow the same virtual circuit and remain in their original order until they reach their destination. Think of a virtual circuit as the lanes of a highway (virtual path). Figure 18.17 shows the relationship between a transmission path (a physical connection), virtual paths (a combination of virtual circuits that are bundled together because parts of their paths are the same), and virtual circuits that logically connect two points.

Figure 18.17 *TP, VPs, and VCs*



To better understand the concept of VPs and VCs, look at Figure 18.18. In this figure, eight endpoints are communicating using four VCs. However, the first two VCs seem to share the same virtual path from switch I to switch III, so it is reasonable to bundle these two VCs together to form one VP. On the other hand, it is clear that the other two VCs share the same path from switch I to switch IV, so it is also reasonable to combine them to form one VP.

Figure 18.18 *Example of VPs and VCs*



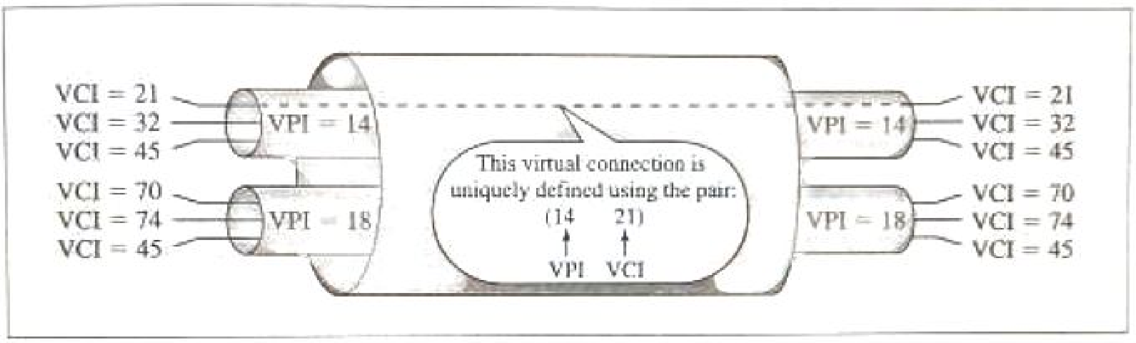
Identifiers

In a virtual circuit network, to route data from one endpoint to another, the virtual connections need to be identified. For this purpose, the designers of ATM created a hierarchical identifier with two levels: a **virtual path identifier (VPI)** and a **virtual circuit identifier (VCI).** The VPI defines the specific VP, and the VCI defines a particular VC inside the VP. The VPI is the same for all virtual connections that are bundled (logically) into one VP.

Note that a virtual connection is defined by a pair of numbers: the VPI and the VCI.

Figure 18.19 shows the VPIs and VCIs for a transmission path. The rationale for dividing an identifier into two parts will become clear when we discuss routing in an ATM network.

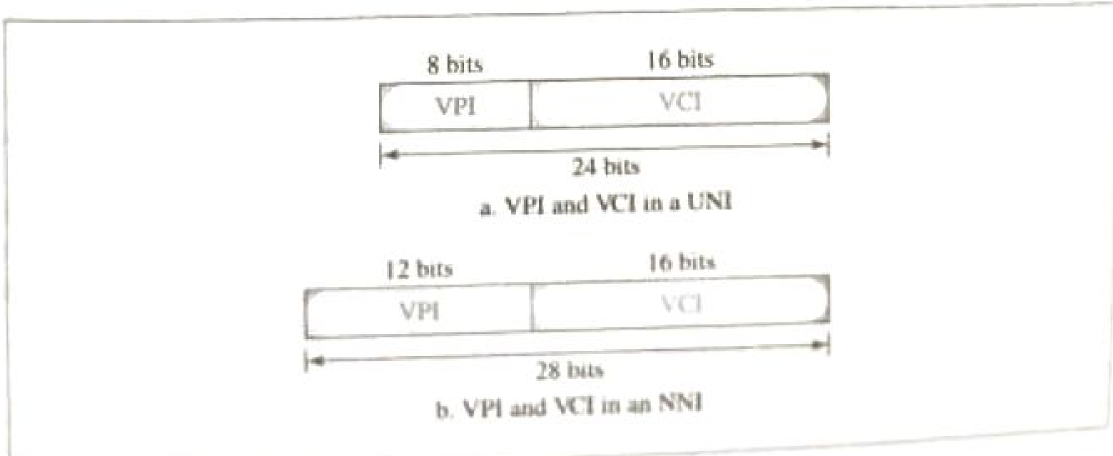
Figure 18.19 *Connection identifiers*



The lengths of the VPIs for UNIs and NNIs are different. In a UNI, the VPI is 8 bits, whereas in an NNI, the VPI is 12 bits. The length of the VCI is the same in both interfaces (16 bits). We therefore can say that a virtual connection is identified by 24 bits in a UNI and by 28 bits in an NNI (see Fig. 18.20).

The whole idea behind dividing a virtual connection identifier into two parts is to allow hierarchical routing. Most of the switches in a typical ATM network are routed using VPIs. The switches at the boundaries of the network, those that interact directly with the endpoint devices, use both VPIs and VCIs.

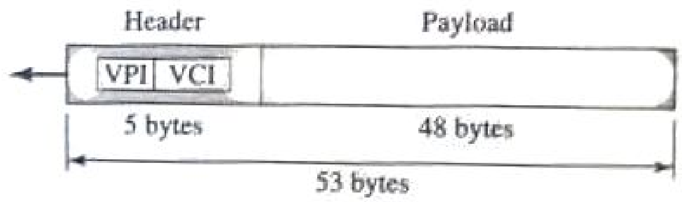
Figure 18.20 *Virtual connection identifiers in UNIs and NNIs*



Cells

The basic data unit in an ATM network is called a cell. A cell is only 53 bytes long with 5 bytes allocated to header and 48 bytes carrying payload (user data may be less than 48 bytes). We will study in detail the fields of a cell, but for the moment it suffices to say that most of the header is occupied by the VPI and VCI that define the virtual connection through which a cell should travel from an endpoint to a switch or from a switch to another switch. Figure 18.21 shows the cell structure.

Figure 18.21 *An ATM cell*



Connection Establishment and Release

Like Frame Relay, ATM uses two types of connections: PVC and SVC.

PVC A permanent virtual circuit connection is established between two endpoints by the network provider. The VPIs and VCIs are defined for the permanent connections, and the values are entered for the tables of each switch.

SVC In a switched virtual circuit connection, each time an endpoint wants to make a connection with another endpoint, a new virtual circuit must be established. ATM cannot do the job by itself, but needs network layer addresses and the services of another protocol (such as IP). The signaling mechanism of this other protocol makes a connection request using the network layer addresses of the two endpoints. The actual mechanism depends on the network layer protocol.

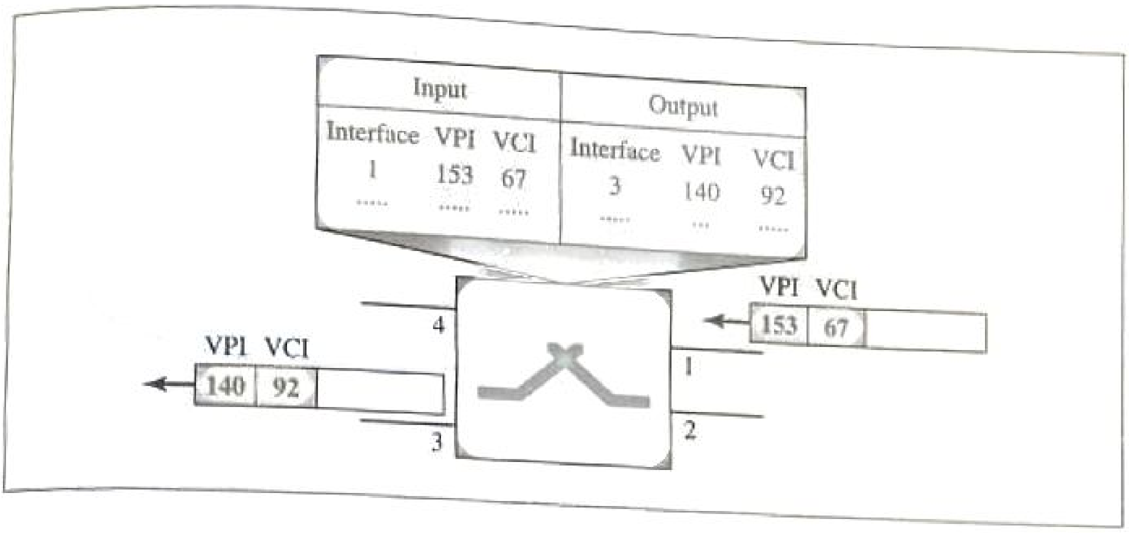
Switching

ATM uses switches to route the cell from a source endpoint to the destination endpoint. A switch routes the cell using both the VPIs and the VCIs. The routing requires the whole identifier. Figure 18.22 shows how a VPC switch routes the cell. A cell with a VPI of 153 and VCI of 67 arrives at switch interface (port) 1. The switch checks its switching table, which stores six pieces of information per row: arrival interface number, incoming VPI, incoming VCI, corresponding outgoing interface number, the new VPI, and the new VCI. The switch finds the entry with the interface 1, VPI 153, and VCI 67 and discovers that the combination corresponds to output interface 3, VPI 140, and VCI 92. It changes the VPI and VCI in the header to 140 and 92, respectively, and sends the cell out through interface 3.

Switching Fabric

The switching technology has created many interesting features to increase the speed of switches to handle data. Because switches are used in both data link layer and

Figure 18.22 *Routing with a switch*

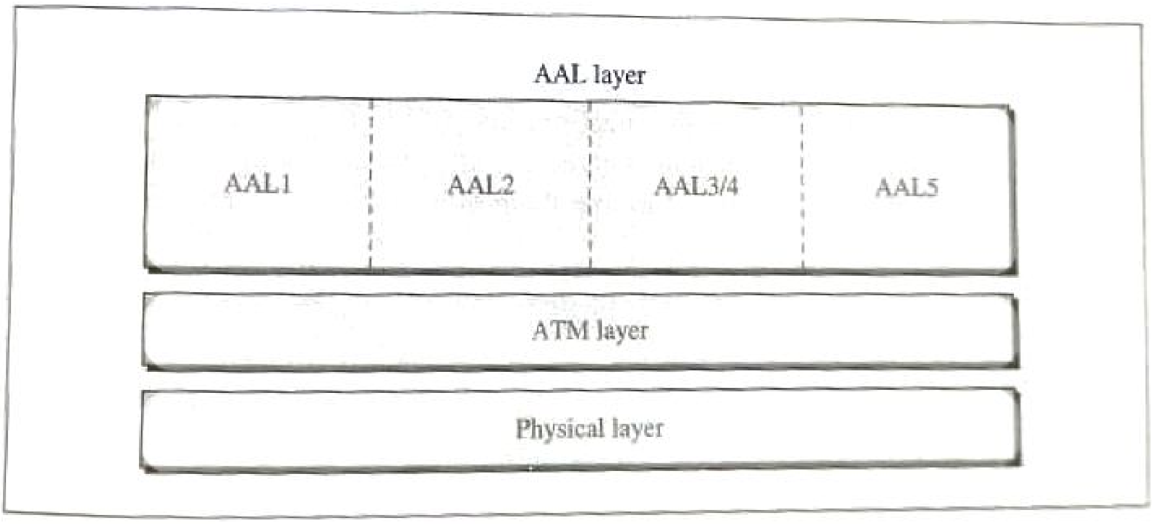


network layer, we do not discuss these variations here. For more information, see Appendix F.

ATM Layers

The ATM standard defines three layers. They are, from top to bottom, the application adaptation layer, the ATM layer, and the physical layer (see Fig. 18.23).

Figure 18.23 *ATM layers*



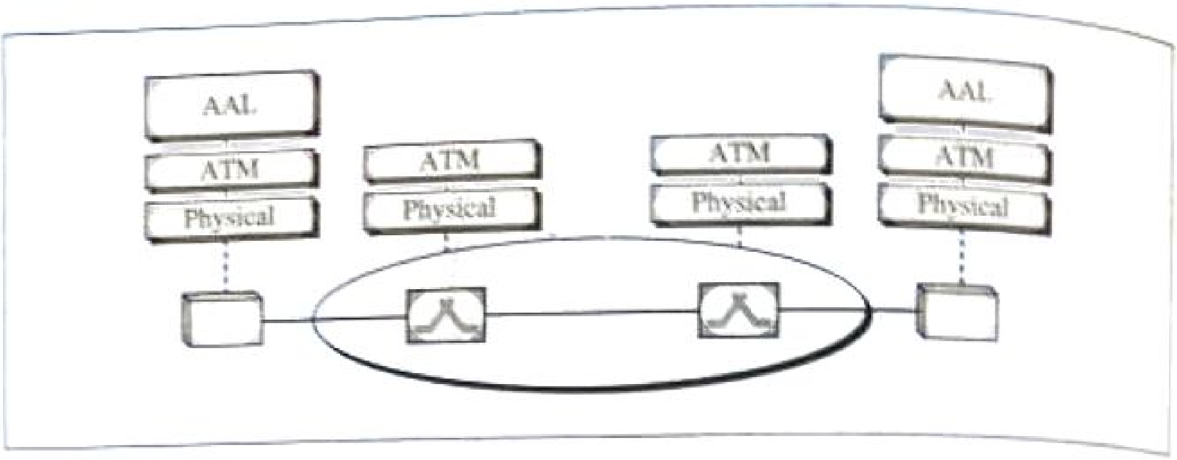
The endpoints use all three layers while the switches use only the two bottom layers (see Fig. 18.24).

Physical Layer

Like Ethernet and wireless LANs, ATM cells can be carried by any physical layer carrier.

SONET The original design of ATM was based on SONET (see Chapter 9) as the physical layer carrier. SONET is preferred for two reasons. First, the high data rate

Figure 18.24 *ATM layers in endpoint devices and switches*



of SONET's carrier reflects the design and philosophy of ATM. Second, in using SONET, the boundaries of cells can be clearly defined. As we saw in Chapter 9, SONET specifies the use of a pointer to define the beginning of a payload. If the beginning of the first ATM cell is defined, the rest of the cells in the same payload can easily be identified because there are no gaps between cells. Just count 53 bytes ahead to find the next cell.

Other Physical Technologies ATM does not limit the physical layer to SONET. Other technologies, even wireless, may be used. However, the problem of cell boundaries must be solved. One solution is for the receiver to guess the end of the cell and apply the CRC to the 5-byte header. If there is no error, the end of the cell is found, to a high probability, correctly. Count 52 bytes back to find the beginning of the cell.

ATM Layer

The **ATM layer** provides routing, traffic management, switching, and multiplexing services. It processes outgoing traffic by accepting 48-byte segments from the AAL sublayers and transforming them into 53-byte cells by the addition of a 5-byte header (see Fig. 18.25).

Header Format ATM uses two formats for this header, one for user-to-network interface (UNI) cells and another for network-to-network interface (NNI) cells. Figure 18.26

Figure 18.25 *ATM layer*

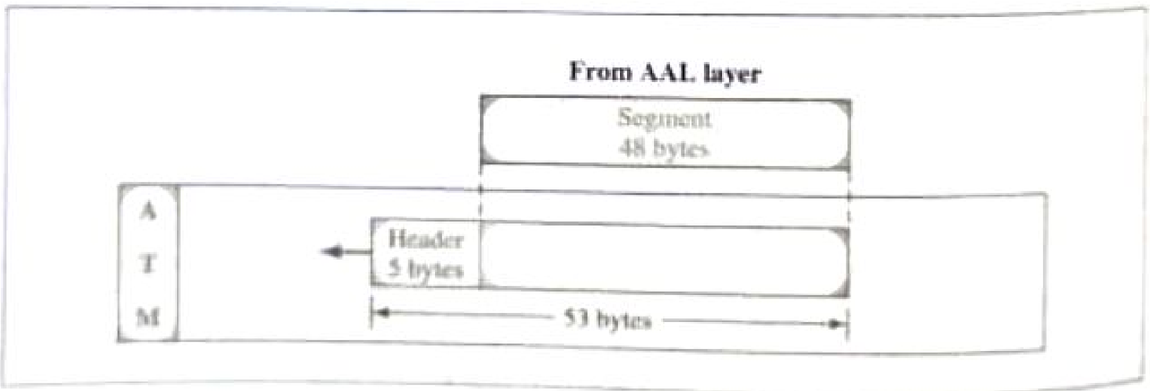
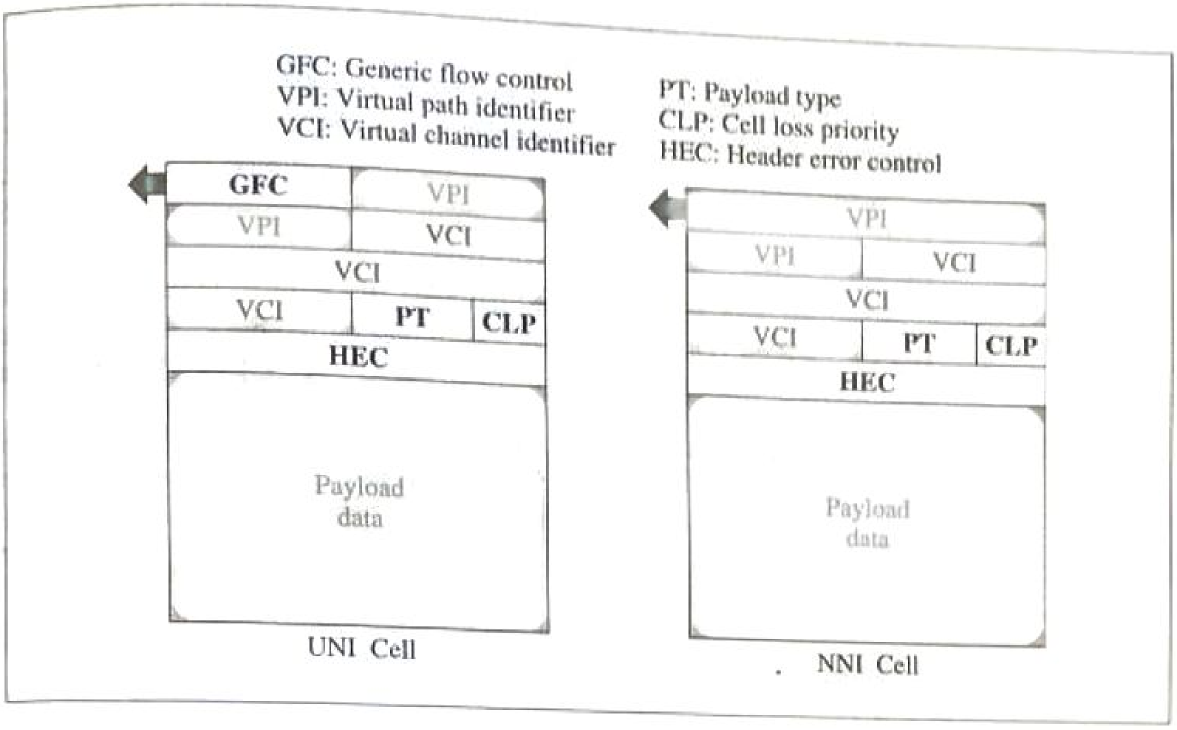


Figure 18.26 *ATM headers*



shows these headers in the byte-by-byte format preferred by the ITU-T (each row represents a byte).

- **Generic flow control (GFC).** The 4-bit GFC field provides flow control at the UNI level. The ITU-T has determined that this level of flow control is not necessary at the NNI level. In the NNI header, therefore, these bits are added to the VPI. The longer VPI allows more virtual paths to be defined at the NNI level. The format for this additional VPI has not yet been determined.
- **Virtual path identifier (VPI).** The VPI is an 8-bit field in a UNI cell and a 12-bit field in an NNI cell (see above).
- **Virtual channel identifier (VCI).** The VCI is a 16-bit field in both frames.
- **Payload type (PT).** In the three-bit PT field, the first bit defines the payload as user data or managerial information. The interpretation of the last 2 bits depends on the first bit.
- **Cell loss priority (CLP).** The 1-bit CLP field is provided for congestion control. A cell with its CLP bit set to 1 must be retained as long as there are cells with a CLP of 0. We discuss congestion control and quality of service in an ATM network in Chapter 23.
- **Header error correction (HEC).** The HEC is a code computed for the first 4 bytes of the header. It is a CRC with the divisor $x^8 + x^2 + x + 1$ that is used to correct single-bit errors and a large class of multiple-bit errors.

Application Adaptation Layer (AAL)

The **application adaptation layer (AAL)** was designed to enable two ATM concepts. First, ATM must accept any type of payload, both data frames and streams of bits. A data frame can come from an upper-layer protocol that creates a clearly defined frame to be sent to a carrier network such as ATM. A good example is the Internet. ATM must also carry multimedia payload. It can accept continuous bit streams and break them into chunks to be encapsulated into a cell at the ATM layer. AAL uses two sub-layers to accomplish these tasks.

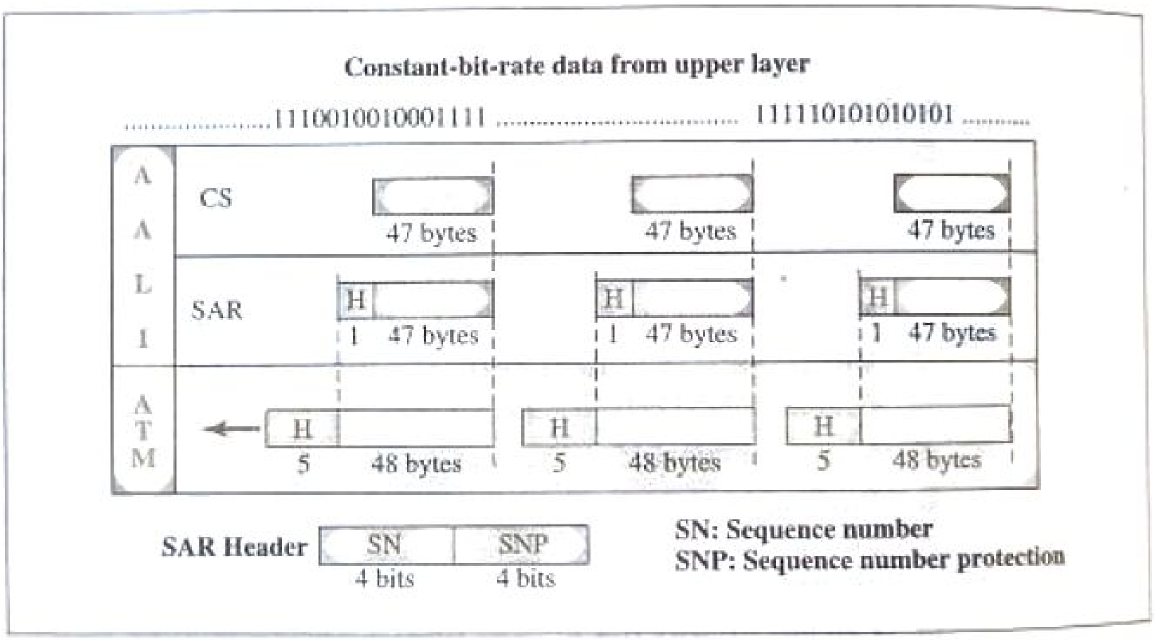
Whether the data are a data frame or a stream of bits, the payload must be segmented into 48-byte segments to be carried by a cell. At the destination, these segments need to be reassembled to recreate the original payload. The AAL defines a sublayer, called a **segmentation and reassembly (SAR)** sublayer, to do so. Segmentation is at the source; reassembly, at the destination.

Before data are segmented by SAR, they must be prepared to guarantee the integrity of the data. This is done by a sublayer called the **convergence sublayer (CS)**.

ATM defines four versions of the AAL: **AAL1, AAL2, AAL3/4,** and **AAL5.**

AAL1 **AAL1** supports applications that transfer information at constant bit rates, such as video and voice. It allows ATM to connect existing digital telephone networks such as voice channels and T-lines. Figure 18.27 shows how a bit stream of data is chopped into 47-byte chunks and encapsulated in cells.

Figure 18.27 *AAL1*



The CS sublayer divides the bit stream into 47-byte segments and passes them to the SAR sublayer below. Note that the CS sublayer does not add a header.

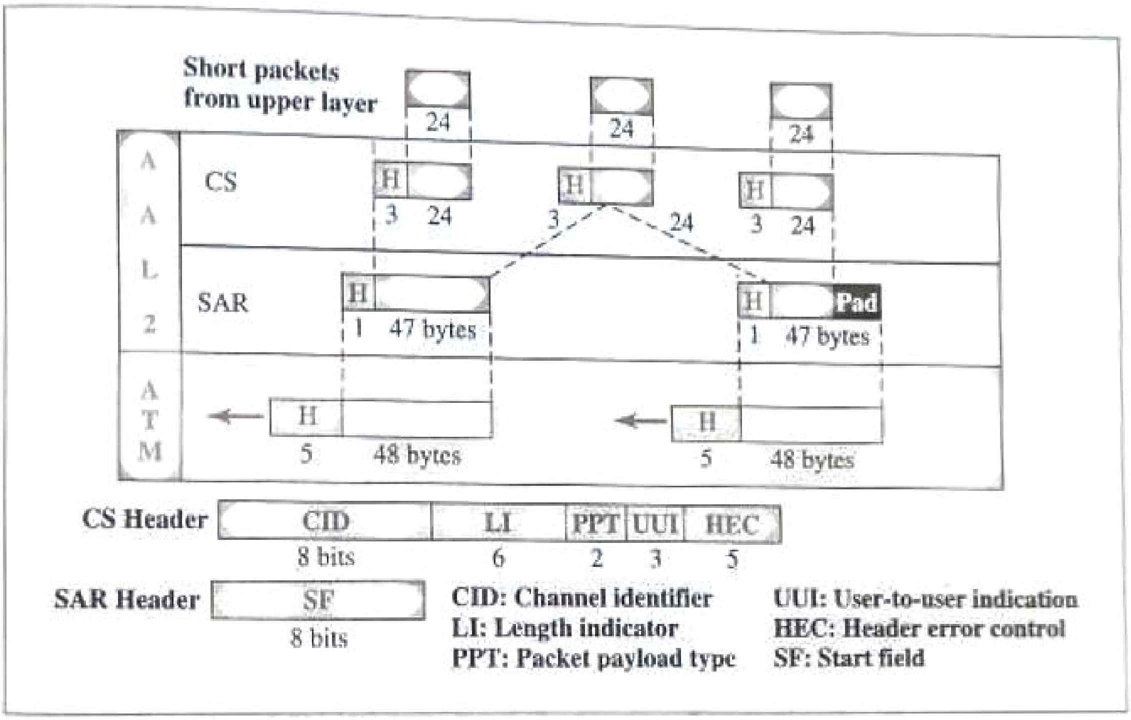
The SAR sublayer adds 1 byte of header and passes the 48-byte segment to the ATM layer. The header has two fields:

- **Sequence number (SN).** This 4-bit field defines a sequence number to order the bits. The first bit is sometimes used for timing, which leaves 3 bits for sequencing (modulo 8).
- **Sequence number protection (SNP).** The second 4-bit field protects the first field. The first 3 bits automatically correct the SN field. The last bit is a parity bit that detects error over all 8 bits.

AAL2 AAL2 was originally intended to support a variable-data-rate bit stream, but it has been redesigned. It is now used for low-bit-rate traffic and short-frame traffic such as audio (compressed or uncompressed), video, or fax. A good example of AAL2 use is in mobile telephony. AAL2 allows the multiplexing of short frames into one cell.

Figure 18.28 shows the process of encapsulating a short frame from the same source (the same user of mobile phone) or from several sources (several users of mobile telephones) into one cell.

Figure 18.28 *AAL2*



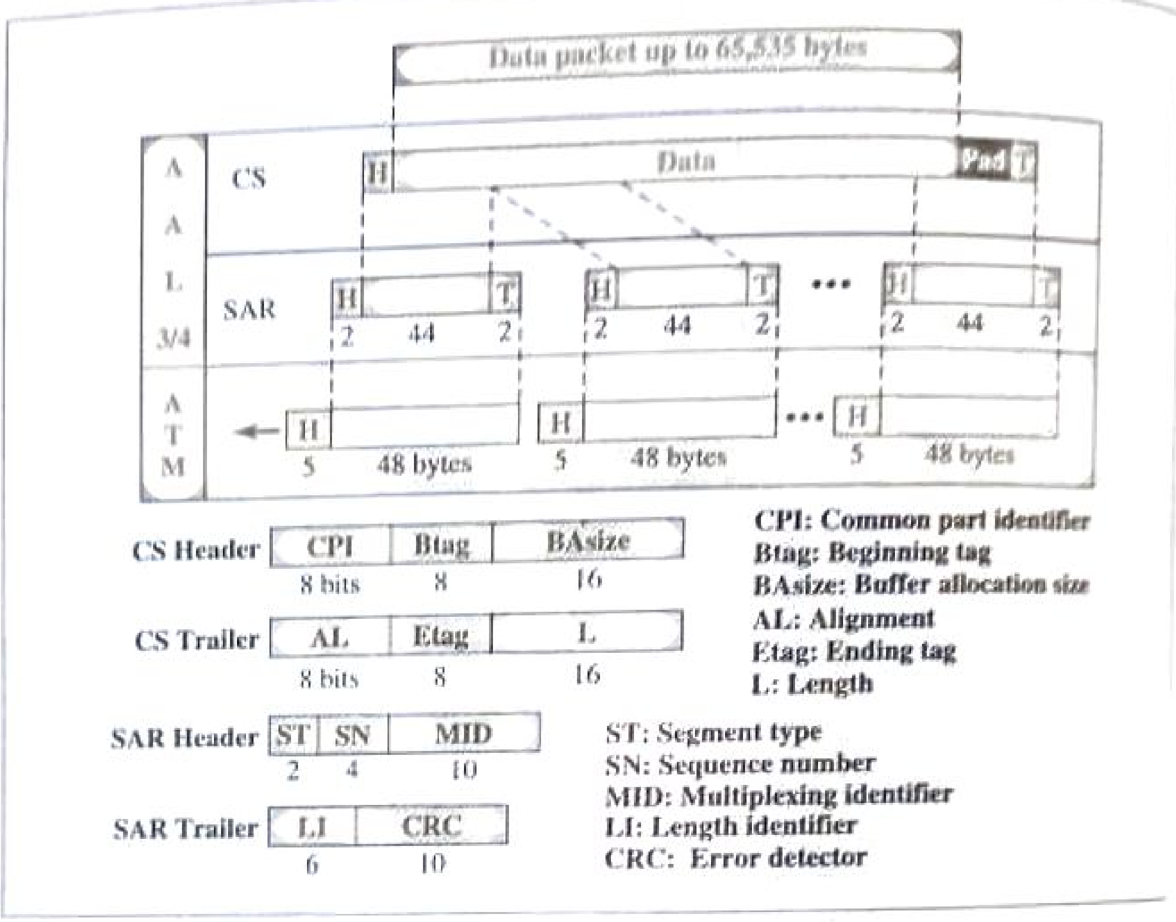
The CS layer overhead consists of five fields:

- **Channel identifier (CID).** The 8-bit CID field defines the channel (user) of the short packet.
- **Length indicator (LI).** The 6-bit LI field indicates how much of the final packet is data.
- **Packet payload type (PPT).** The PPT field defines the type of the packet.
- **User-to-user indicator (UUI).** The UUI field can be used by end-to-end users.
- **Header error control (HEC).** The last 5 bits are used to correct errors in the header.

The only overhead at the SAR layer is the start field (SF) that defines the offset from the beginning of the packet.

AAL3/4 Initially, AAL3 was intended to support connection-oriented data services and AAL4 to support connectionless services. As they evolved, however, it became evident that the fundamental issues of the two protocols were the same. They have therefore been combined into a single format called **AAL3/4.** Figure 18.29 shows the AAL3/4 sublayer

Figure 18.29 *AAL3/4*



The CS layer header and trailer consists of six fields:

- **Common part identifier (CPI).** The CPI defines how the subsequent fields are to be interpreted. The value at present is 0.
- **Begin tag (Btag).** The value of this field is repeated in each cell to identify all the cells beginning to the same packet. The value is the same as the Etag (see below).
- **Buffer allocation size (BAsize).** The 2-byte BA field tells the receiver what size buffer is needed for the coming data.
- **Alignment (AL).** The 1-byte AL field is included to make the rest of the trailer 4 bytes long.
- **Ending tag (Etag).** The 1-byte ET field serves as an ending flag. Its value is the same as that of the beginning tag.
- **Length (L).** The 2-byte L field indicates the length of the data unit.

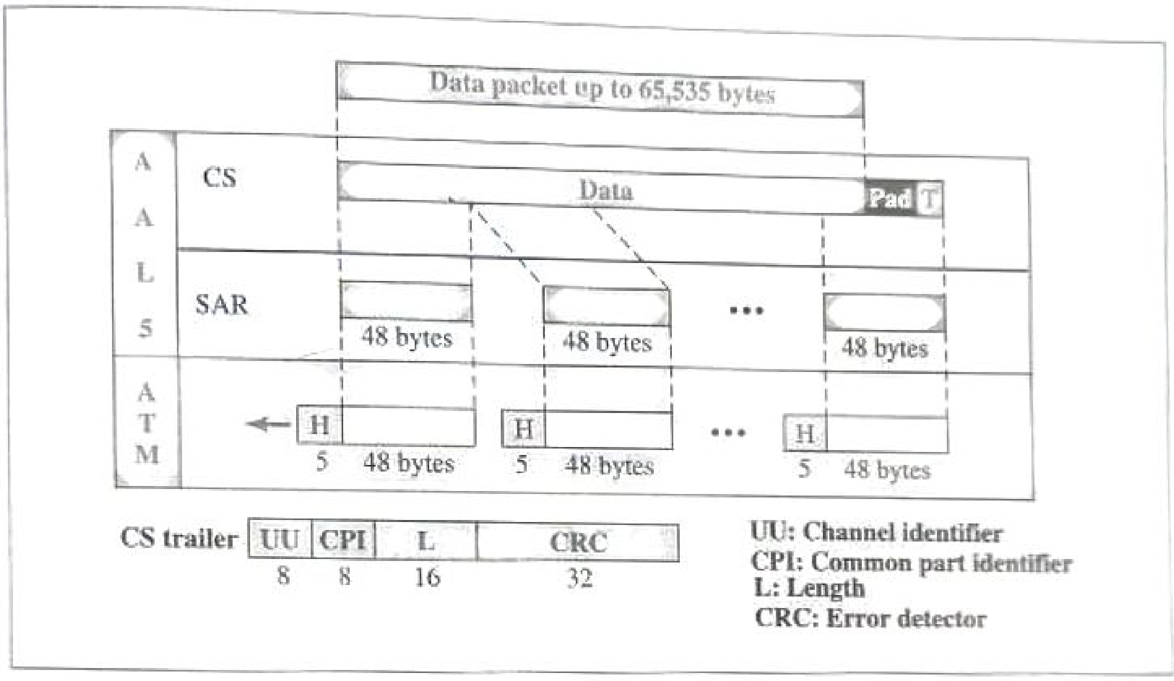
The SAR header and trailer consist of five fields:

- **Segment type (ST).** The 2-bit ST identifier specifies the position of the segment in the message: beginning (00), middle (01), or end (10). A single-segment message has an ST of 11.
- **Sequence number (SN).** This field is the same as defined previously.
- **Multiplexing identification (MID).** The 10-bit MID field identifies cells coming from different data flows and multiplexed on the same virtual connection.

- **Length indicator (LI).** This field defines how much of the packet is data, not padding.
- **CRC.** The last 10 bits of the trailer is a CRC for the entire data unit.

AAL5 AAL3/4 provides comprehensive sequencing and error control mechanisms that are not necessary for every application. For these applications, the designers of ATM have provided a fifth AAL sublayer, called the **simple and efficient adaptation layer (SEAL). AAL5** assumes that all cells belonging to a single message travel sequentially and that control functions are included in the upper layers of the sending application. Figure 18.30 shows the AAL5 sublayer.

Figure 18.30 *AAL5*



The four trailer fields in the CS layer are

- **User-to-user (UU).** This field is used by end users, as described previously.
- **Common part identifier (CPI).** This field is the same as defined previously.
- **Length (L).** The 2-byte L field indicates the length of the original data.
- **CRC.** The last 4 bytes are for error control on the entire data unit.

Congestion Control and Quality of Service

ATM has a very developed congestion control and quality of service that we discuss in Chapter 23.

ATM LANs

A lot of effort has been made to apply ATM technology to LANs. The result is the ATM LAN. We talk about ATM LANs in Appendix G.