

UNIT II

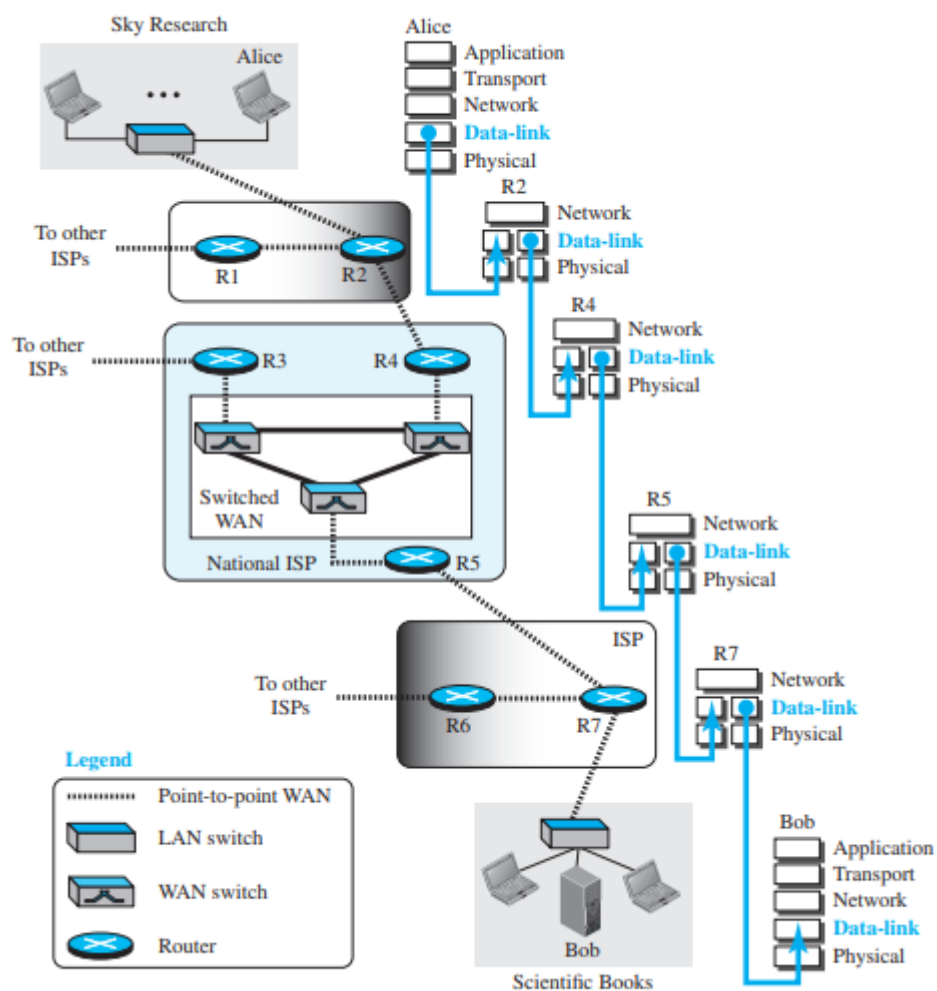
DATA-LINK LAYER & MEDIA ACCESS

Introduction – Link-Layer Addressing – DLC Services – Data-Link Layer Protocols – HDLC – PPP - Media Access Control - Wired LANs: Ethernet - Wireless LANs – Introduction – IEEE 802.11, Bluetooth – Connecting Devices.

INTRODUCTION

The Internet is a combination of networks glued together by connecting devices (routers or switches). If a packet is to travel from a host to another host, it needs to pass through these networks. Communication at the data-link layer is made up of five separate logical connections between the data-link layers in the path.

Figure 9.1 *Communication at the data-link layer*



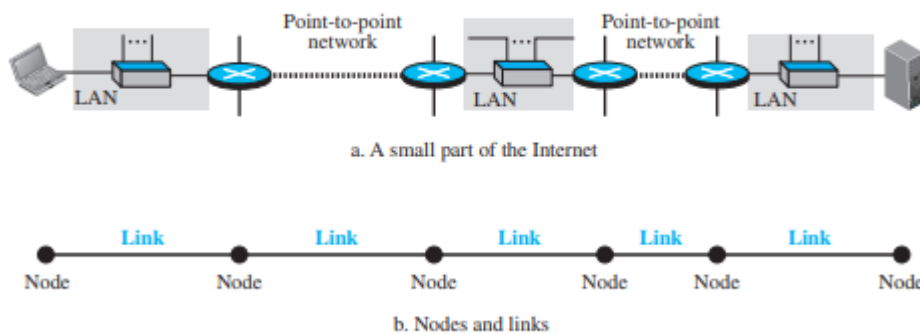
The data-link layer at Alice's computer communicates with the data-link layer at router R2. The data-link layer at router R2 communicates with the data-link layer at router R4, and so on. Finally, the data-link layer at router R7 communicates with the data-link layer at Bob's computer. Only one data-link layer is involved at the source or the destination, but two data-link

layers are involved at each router. The reason is that Alice's and Bob's computers are each connected to a single network, but each router takes input from one network and sends output to another network.

Nodes and Links

Communication at the data-link layer is node-to-node. A data unit from one point in the Internet needs to pass through many networks (LANs and WANs) to reach another point. These LANs and WANs are connected by routers. It is customary to refer to the two end hosts and the routers as nodes and the networks in between as links.

Figure 9.2 *Nodes and Links*

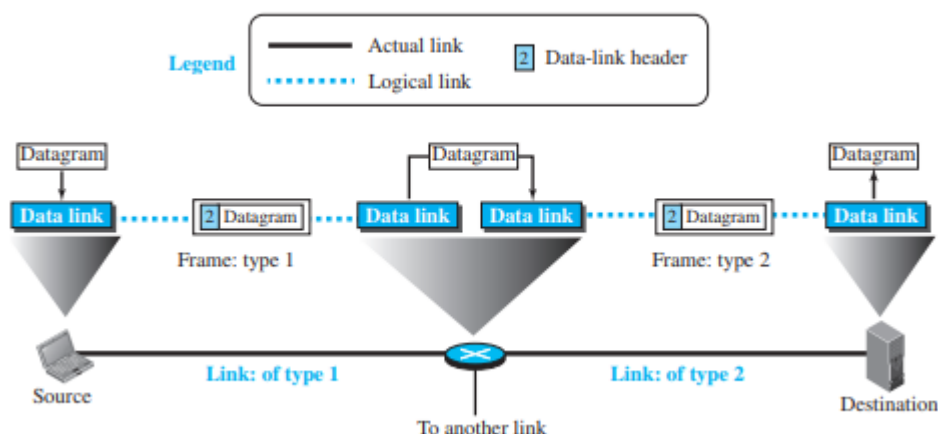


The first node is the source host; the last node is the destination host. The other four nodes are four routers. The first, the third, and the fifth links represent the three LANs; the second and the fourth links represent the two WANs.

Services

The data-link layer is located between the physical and the network layers. The datalink layer provides services to the network layer; it receives services from the physical layer.

Figure 9.3 *A communication with only three nodes*



The datagram received by the data-link layer of the source host is encapsulated in a frame. The frame is logically transported from the source host to the router. The frame is decapsulated at the data-link layer of the router and encapsulated at another frame. The new frame is logically transported from the router to the destination host. Note that, although we

have shown only two data-link layers at the router, the router actually has three data-link layers because it is connected to three physical links.

Framing

Definitely, the first service provided by the data-link layer is framing. The data-link layer at each node needs to encapsulate the datagram (packet received from the network layer) in a frame before sending it to the next node. The node also needs to decapsulate the datagram from the frame received on the logical channel. A packet at the data-link layer is normally called a frame.

Flow Control

The sending data-link layer at the end of a link is a producer of frames; the receiving data-link layer at the other end of a link is a consumer. If the rate of produced frames is higher than the rate of consumed frames, frames at the receiving end need to be buffered while waiting to be consumed (processed). Definitely, we cannot have an unlimited buffer size at the receiving side. We have two choices. The first choice is to let the receiving data-link layer drop the frames if its buffer is full. The second choice is to let the receiving data-link layer send a feedback to the sending data-link layer to ask it to stop or slow down.

Error Control

At the sending node, a frame in a data-link layer needs to be changed to bits, transformed to electromagnetic signals, and transmitted through the transmission media. At the receiving node, electromagnetic signals are received, transformed to bits, and put together to create a frame. Since electromagnetic signals are susceptible to error, a frame is susceptible to error. The error needs first to be detected. After detection, it needs to be either corrected at the receiver node or discarded and retransmitted by the sending node.

Congestion Control

Although a link may be congested with frames, which may result in frame loss, most data-link-layer protocols do not directly use a congestion control to alleviate congestion, although some wide-area networks do. In general, congestion control is considered an issue in the network layer or the transport layer because of its end-to-end nature.

Two Categories of Links

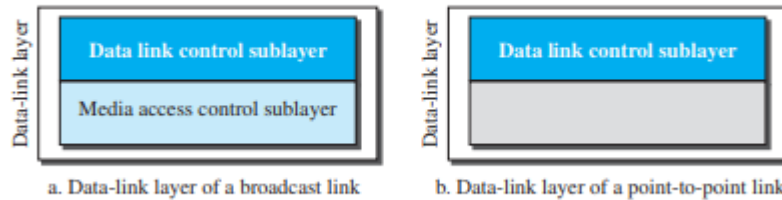
We can have a **point-to-point link** or a **broadcast link**. In a point-to-point link, the link is dedicated to the two devices; in a broadcast link, the link is shared between several pairs of devices. For example, when two friends use the traditional home phones to chat, they are using a point-to-point link; when the same two friends use their cellular phones, they are using a broadcast link.

Two Sublayers

To better understand the functionality of and the services provided by the link layer, we can divide the data-link layer into two sublayers: **data link control (DLC)** and **media access control (MAC)**. The data link control sublayer deals with all issues common to both point-to-

point and broadcast links; the media access control sublayer deals only with issues specific to broadcast links.

Figure 9.4 *Dividing the data-link layer into two sublayers*



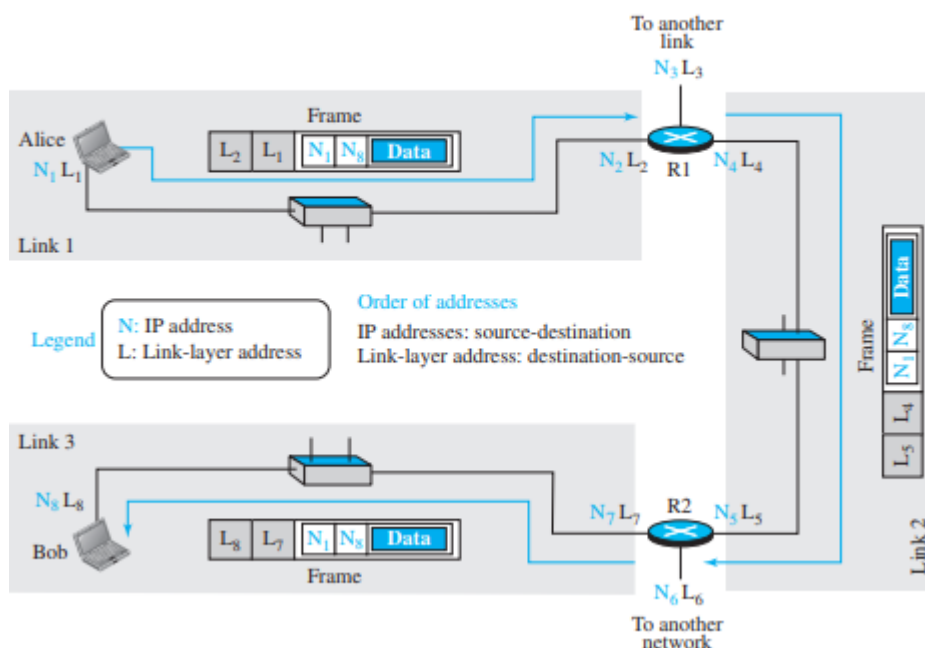
LINK-LAYER ADDRESSING

The source and destination IP addresses define the two ends but cannot define which links the datagram should pass through. We need to remember that the IP addresses in a datagram should not be changed. If the destination IP address in a datagram changes, the packet never reaches its destination; if the source IP address in a datagram changes, the destination host or a router can never communicate with the source if a response needs to be sent back or an error needs to be reported back to the source.

We need another addressing mechanism in a connectionless internetwork: the link-layer addresses of the two nodes. A link-layer address is sometimes called a link address, sometimes a physical address, and sometimes a MAC address.

Since a link is controlled at the data-link layer, the addresses need to belong to the data-link layer. When a datagram passes from the network layer to the data-link layer, the datagram will be encapsulated in a frame and two data-link addresses are added to the frame header. These two addresses are changed every time the frame moves from one link to another.

Figure 9.5 *IP addresses and link-layer addresses in a small internet*



In the internet in Figure 9.5, we have three links and two routers. We also have shown only two hosts: Alice (source) and Bob (destination). For each host, we have shown two addresses, the IP addresses (N) and the link-layer addresses (L). Note that a router has as many pairs of addresses as the number of links the router is connected to. We have shown three frames, one in each link. Each frame carries the same datagram with the same source and destination addresses (N1 and N8), but the link-layer addresses of the frame change from link to link. In link 1, the link-layer addresses are L1 and L2. In link 2, they are L4 and L5. In link 3, they are L7 and L8. Note that the IP addresses and the link-layer addresses are not in the same order. For IP addresses, the source address comes before the destination address; for link-layer addresses, the destination address comes before the source. The datagrams and frames are designed in this way, and we follow the design.

We may raise several questions:

- If the IP address of a router does not appear in any datagram sent from a source to a destination, why do we need to assign IP addresses to routers? The answer is that in some protocols a router may act as a sender or receiver of a datagram. For example, in routing protocols we will discuss in Chapters 20 and 21, a router is a sender or a receiver of a message. The communications in these protocols are between routers.
- Why do we need more than one IP address in a router, one for each interface? The answer is that an interface is a connection of a router to a link. We will see that an IP address defines a point in the Internet at which a device is connected. A router with n interfaces is connected to the Internet at n points. This is the situation of a house at the corner of a street with two gates; each gate has the address related to the corresponding street.
- How are the source and destination IP addresses in a packet determined? The answer is that the host should know its own IP address, which becomes the source IP address in the packet. As we will discuss in Chapter 26, the application layer uses the services of DNS to find the destination address of the packet and passes it to the network layer to be inserted in the packet.
- How are the source and destination link-layer addresses determined for each link? Again, each hop (router or host) should know its own link-layer address, as we discuss later in the chapter. The destination link-layer address is determined by using the Address Resolution Protocol, which we discuss shortly.
- What is the size of link-layer addresses? The answer is that it depends on the protocol used by the link. Although we have only one IP protocol for the whole Internet, we may be using different data-link protocols in different links. This means that we can define the size of the address when we discuss different link-layer protocols.

Three Types of addresses

Some link-layer protocols define three types of addresses: **unicast**, **multicast**, and **broadcast**.

Unicast Address

Each host or each interface of a router is assigned a unicast address. Unicasting means one-to-one communication. A frame with a unicast address destination is destined only for one entity in the link.

Unicast link-layer addresses in the most common LAN, Ethernet, are 48 bits (six bytes) that are presented as 12 hexadecimal digits separated by colons; for example, the following is a link-layer address of a computer.

A3:34:45:11:92:F1

Multicast Address

Some link-layer protocols define multicast addresses. Multicasting means one-to-many communication. The multicast link-layer addresses in the most common LAN, Ethernet, are 48 bits (six bytes) that are presented as 12 hexadecimal digits separated by colons. The second digit, however, needs to be an even number in hexadecimal. The following shows a multicast address:

A2:34:45:11:92:F1

Broadcast Address

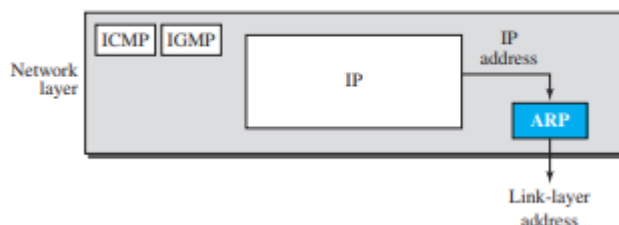
Some link-layer protocols define a broadcast address. Broadcasting means one-to-all communication. A frame with a destination broadcast address is sent to all entities in the link. The broadcast link-layer addresses in the most common LAN, Ethernet, are 48 bits, all 1s, that are presented as 12 hexadecimal digits separated by colons. The following shows a broadcast address:

FF:FF:FF:FF:FF:FF

Address Resolution Protocol (ARP)

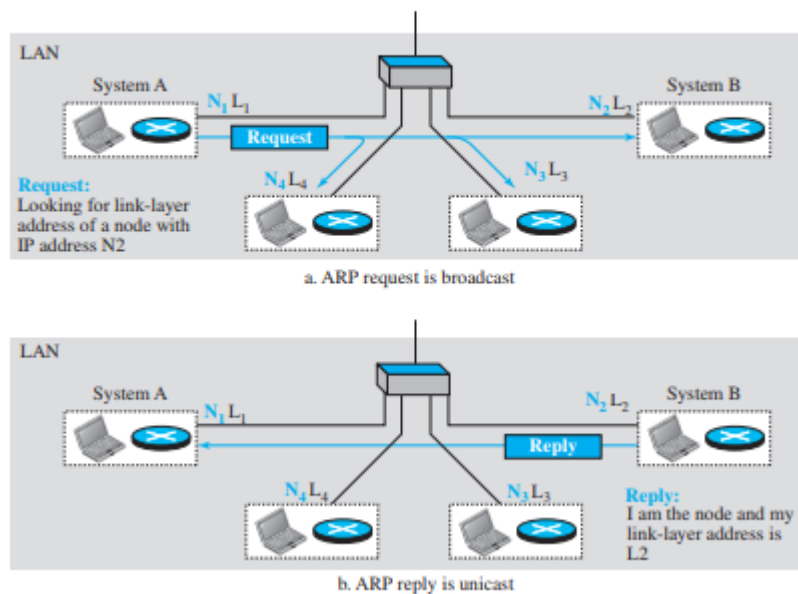
Anytime a node has an IP datagram to send to another node in a link, it has the IP address of the receiving node. The source host knows the IP address of the default router. Each router except the last one in the path gets the IP address of the next router by using its forwarding table. The last router knows the IP address of the destination host. However, the IP address of the next node is not helpful in moving a frame through a link; we need the link-layer address of the next node. This is the time when the Address Resolution Protocol (ARP) becomes helpful. It belongs to the network layer, but we discuss it in this chapter because it maps an IP address to a logical-link address. ARP accepts an IP address from the IP protocol, maps the address to the corresponding link-layer address, and passes it to the data-link layer.

Position of ARP in TCP/IP protocol suite



Anytime a host or a router needs to find the link-layer address of another host or router in its network, it sends an ARP request packet. The packet includes the link-layer and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the link-layer address of the receiver, the query is broadcast over the link using the link-layer broadcast address.

Figure 9.7 ARP operation



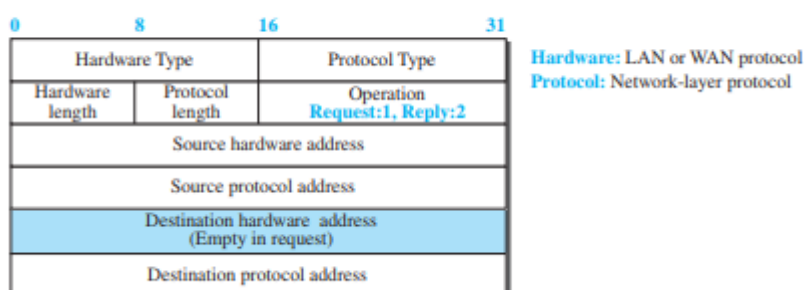
Every host or router on the network receives and processes the ARP request packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and link-layer addresses. The packet is unicast directly to the node that sent the request packet.

In Figure 9.7a, the system on the left (A) has a packet that needs to be delivered to another system (B) with IP address N_2 . System A needs to pass the packet to its data-link layer for the actual delivery, but it does not know the physical address of the recipient. It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of N_2 .

This packet is received by every system on the physical network, but only system B will answer it, as shown in Figure 9.7b. System B sends an ARP reply packet that includes its physical address. Now system A can send all the packets it has for this destination using the physical address it received.

Packet Format

Figure 9.8 ARP packet



The names of the fields are selfexplanatory. The hardware type field defines the type of the link-layer protocol; Ethernet is given the type 1. The protocol type field defines the network-layer protocol: IPv4 protocol is (0800)16. The source hardware and source protocol addresses are variable-length fields defining the link-layer and network-layer addresses of the sender. The

destination hardware address and destination protocol address fields define the receiver link-layer and network-layer addresses. An ARP packet is encapsulated directly into a data-link frame. The frame needs to have a field to show that the payload belongs to the ARP and not to the network-layer datagram.

DLC SERVICES

The data link control (DLC) deals with procedures for communication between two adjacent nodes—node-to-node communication—no matter whether the link is dedicated or broadcast. Data link control functions include framing and flow and error control.

Framing

Framing in the data-link layer separates a message from one source to a destination by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.

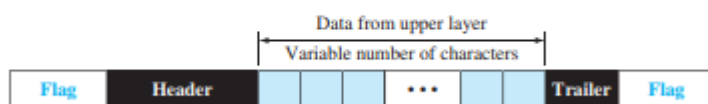
Frame Size

Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. In variable-size framing, we need a way to define the end of one frame and the beginning of the next. Historically, two approaches were used for this purpose: a character-oriented approach and a bit-oriented approach.

Character-Oriented Framing

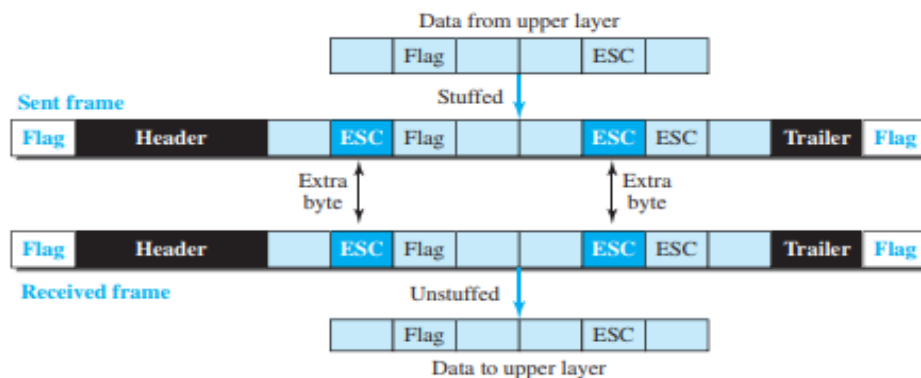
In character-oriented (or byte-oriented) framing, data to be carried are 8-bit characters from a coding system such as ASCII (see Appendix A). The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection redundant bits, are also multiples of 8 bits. To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame.

Figure 11.1 A frame in a character-oriented protocol



Character-oriented framing was popular when only text was exchanged by the data-link layers. The flag could be selected to be any character not used for text communication. Now, however, we send other types of information such as graphs, audio, and video; any character used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame. To fix this problem, a byte-stuffing strategy was added to character-oriented framing. In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC) and has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not as a delimiting flag.

Figure 11.2 *Byte stuffing and unstuffing*

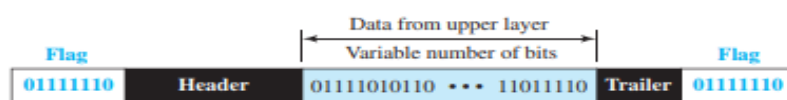


Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it creates another problem. What happens if the text contains one or more escape characters followed by a byte with the same pattern as the flag? The receiver removes the escape character, but keeps the next byte, which is incorrectly interpreted as the end of the frame. To solve this problem, the escape characters that are part of the text must also be marked by another escape character. In other words, if the escape character is part of the text, an extra one is added to show that the second one is part of the text.

Bit-Oriented Framing

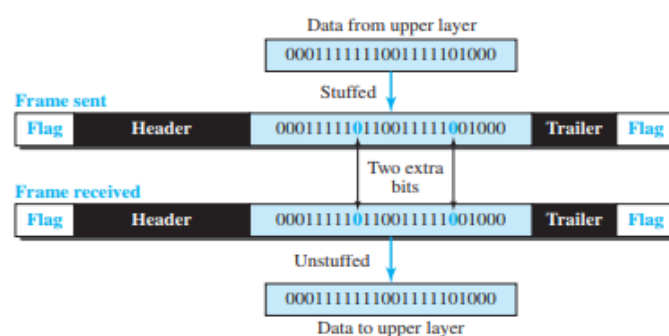
In bit-oriented framing, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on. However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag, 01111110, as the delimiter to define the beginning and the end of the frame.

Figure 11.3 *A frame in a bit-oriented protocol*



This flag can create the same type of problem we saw in the character-oriented protocols. That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag.

Figure 11.4 *Bit stuffing and unstuffing*



The strategy is called bit stuffing. In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver. Note that the extra bit is added after one 0 followed by five 1s regardless of the value of the next bit. This guarantees that the flag field sequence does not inadvertently appear in the frame.

Note that even if we have a 0 after five 1s, we still stuff a 0. The 0 will be removed by the receiver. This means that if the flaglike pattern 01111110 appears in the data, it will change to 011111010 (stuffed) and is not mistaken for a flag by the receiver. The real flag 01111110 is not stuffed by the sender and is recognized by the receiver.

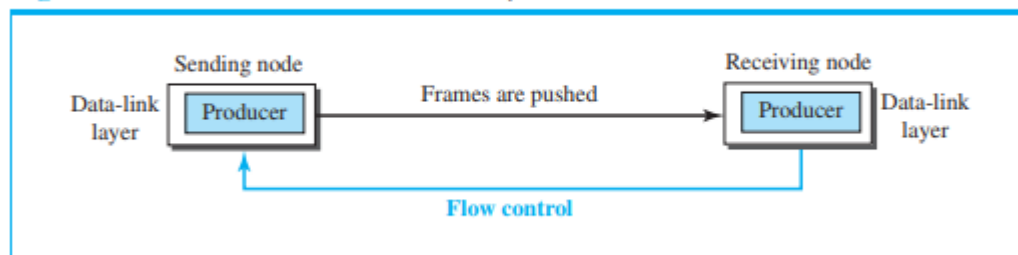
Flow and Error Control

One of the responsibilities of the data-link control sublayer is flow and error control at the data-link layer.

Flow Control

Whenever an entity produces items and another entity consumes them, there should be a balance between production and consumption rates. If the items are produced faster than they can be consumed, the consumer can be overwhelmed and may need to discard some items. If the items are produced more slowly than they can be consumed, the consumer must wait, and the system becomes less efficient. Flow control is related to the first issue. We need to prevent losing the data items at the consumer site.

Figure 11.5 Flow control at the data-link layer



The data-link layer at the sending node tries to push frames toward the data-link layer at the receiving node. If the receiving node cannot process and deliver the packet to its network at the same rate that the frames arrive, it becomes overwhelmed with frames. Flow control in this case can be feedback from the receiving node to the sending node to stop or slow down pushing frames.

Buffers

Although flow control can be implemented in several ways, one of the solutions is normally to use two buffers; one at the sending data-link layer and the other at the receiving data-link layer. A buffer is a set of memory locations that can hold packets at the sender and receiver. The flow control communication can occur by sending signals from the consumer to the producer. When the buffer of the receiving data-link layer is full, it informs the sending data-link layer to stop pushing frames.

Error Control

Since the underlying technology at the physical layer is not fully reliable, we need to implement error control at the data-link layer to prevent the receiving node from delivering

corrupted packets to its network layer. Error control at the data-link layer is normally very simple and implemented using one of the following two methods.

In both methods, a CRC is added to the frame header by the sender and checked by the receiver.

- In the first method, if the frame is corrupted, it is silently discarded; if it is not corrupted, the packet is delivered to the network layer. This method is used mostly in wired LANs such as Ethernet.
- In the second method, if the frame is corrupted, it is silently discarded; if it is not corrupted, an acknowledgment is sent (for the purpose of both flow and error control) to the sender.

Combination of Flow and Error Control

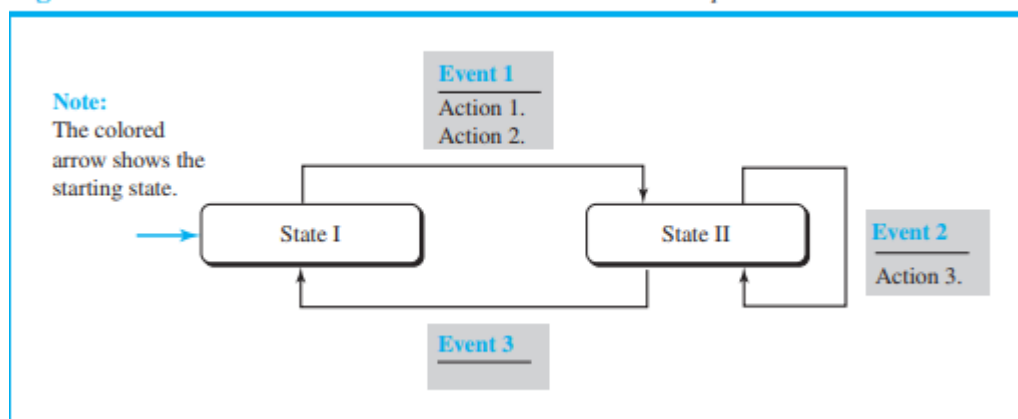
Flow and error control can be combined. In a simple situation, the acknowledgment that is sent for flow control can also be used for error control to tell the sender the packet has arrived uncorrupted. The lack of acknowledgment means that there is a problem in the sent frame. We show this situation when we discuss some simple protocols in the next section. A frame that carries an acknowledgment is normally called an ACK to distinguish it from the data frame.

DATA-LINK LAYER PROTOCOLS

Traditionally four protocols have been defined for the data-link layer to deal with flow and error control: **Simple, Stop-and-Wait, Go-Back-N, and Selective-Repeat**. Although the first two protocols still are used at the data-link layer, the last two have disappeared.

The behavior of a data-link-layer protocol can be better shown as a **finite state machine (FSM)**. An FSM is thought of as a machine with a finite number of states. The machine is always in one of the states until an event occurs. Each event is associated with two reactions: defining the list (possibly empty) of actions to be performed and determining the next state (which can be the same as the current state). One of the states must be defined as the initial state, the state in which the machine starts when it turns on.

Figure 11.6 Connectionless and connection-oriented service represented as FSMs



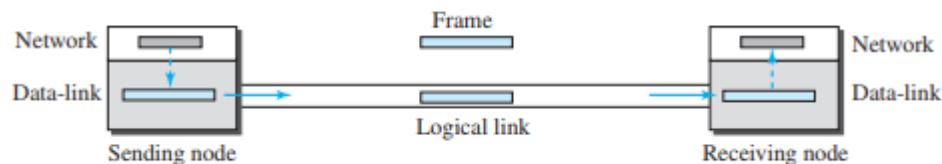
The figure shows a machine with three states. There are only three possible events and three possible actions. The machine starts in state I. If event 1 occurs, the machine performs actions 1 and 2 and moves to state II. When the machine is in state II, two events may occur. If

event 1 occurs, the machine performs action 3 and remains in the same state, state II. If event 3 occurs, the machine performs no action, but move to state I.

Simple Protocol

Our first protocol is a simple protocol with neither flow nor error control. We assume that the receiver can immediately handle any frame it receives.

Figure 11.7 *Simple protocol*

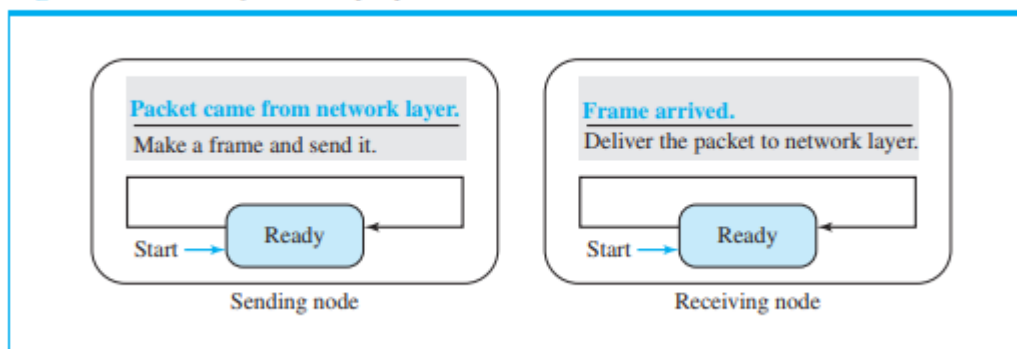


The data-link layer at the sender gets a packet from its network layer, makes a frame out of it, and sends the frame. The data-link layer at the receiver receives a frame from the link, extracts the packet from the frame, and delivers the packet to its network layer. The data-link layers of the sender and receiver provide transmission services for their network layers.

FSMs

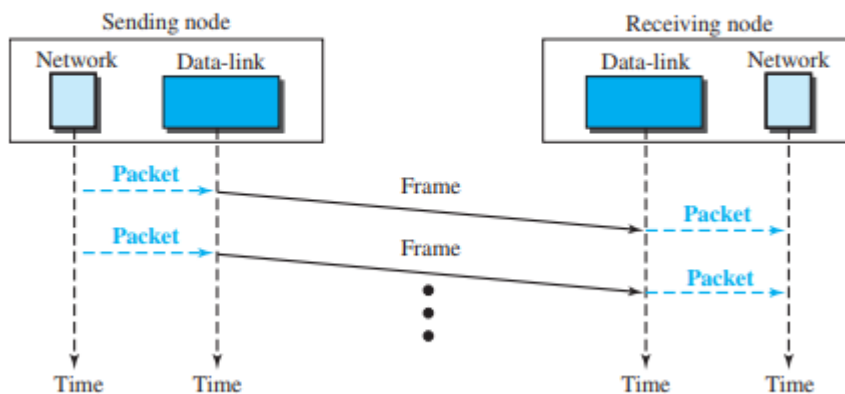
The sender site should not send a frame until its network layer has a message to send. The receiver site cannot deliver a message to its network layer until a frame arrives. We can show these requirements using two FSMs. Each FSM has only one state, the ready state. The sending machine remains in the ready state until a request comes from the process in the network layer. When this event occurs, the sending machine encapsulates the message in a frame and sends it to the receiving machine. The receiving machine remains in the ready state until a frame arrives from the sending machine. When this event occurs, the receiving machine decapsulates the message out of the frame and delivers it to the process at the network layer.

Figure 11.8 *FSMs for the simple protocol*



Below figure shows an example of communication using this protocol. It is very simple. The sender sends frames one after another without even thinking about the receiver.

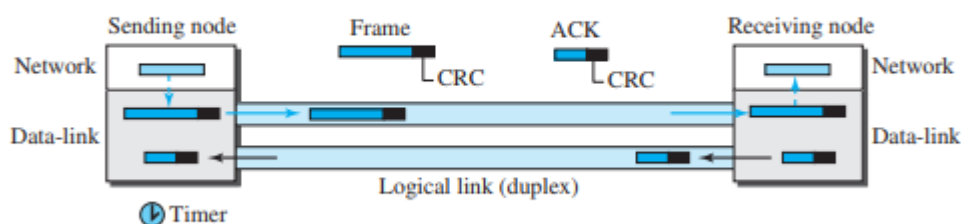
Figure 11.9 Flow diagram for Example 11.2



Stop-and-Wait Protocol

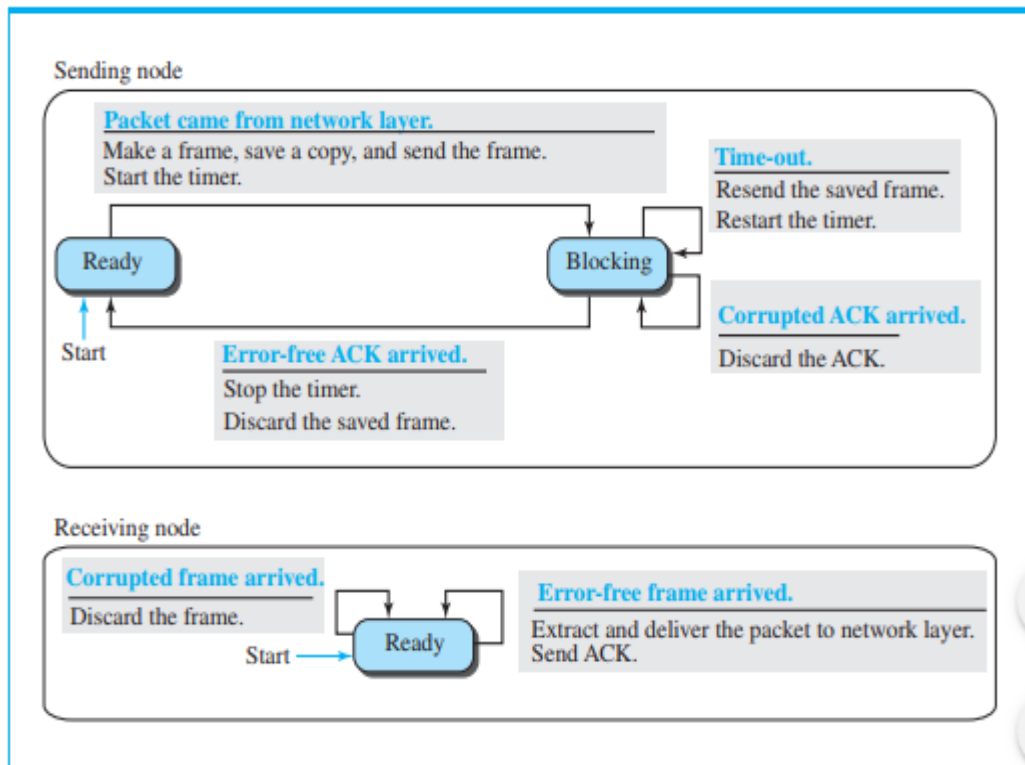
Our second protocol is called the Stop-and-Wait protocol, which uses both flow and error control. In this protocol, the sender sends one frame at a time and waits for an acknowledgment before sending the next one. To detect corrupted frames, we need to add a CRC to each data frame. When a frame arrives at the receiver site, it is checked. If its CRC is incorrect, the frame is corrupted and silently discarded. The silence of the receiver is a signal for the sender that a frame was either corrupted or lost. Every time the sender sends a frame, it starts a timer. If an acknowledgment arrives before the timer expires, the timer is stopped and the sender sends the next frame (if it has one to send). If the timer expires, the sender resends the previous frame, assuming that the frame was either lost or corrupted. This means that the sender needs to keep a copy of the frame until its acknowledgment arrives. When the corresponding acknowledgment arrives, the sender discards the copy and sends the next frame if it is ready.

Figure 11.10 Stop-and-Wait protocol



FSMs

Figure 11.11 *FSM for the Stop-and-Wait protocol*



Sender States

The sender is initially in the ready state, but it can move between the ready and blocking state.

- **Ready State.** When the sender is in this state, it is only waiting for a packet from the network layer. If a packet comes from the network layer, the sender creates a frame, saves a copy of the frame, starts the only timer and sends the frame. The sender then moves to the blocking state.
- **Blocking State.** When the sender is in this state, three events can occur:
 - a. If a time-out occurs, the sender resends the saved copy of the frame and restarts the timer.
 - b. If a corrupted ACK arrives, it is discarded.
 - c. If an error-free ACK arrives, the sender stops the timer and discards the saved copy of the frame. It then moves to the ready state.

Receiver

The receiver is always in the ready state. Two events may occur:

- a. If an error-free frame arrives, the message in the frame is delivered to the network layer and an ACK is sent.
- b. If a corrupted frame arrives, the frame is discarded.

Figure 11.12 Flow diagram for Example 11.3

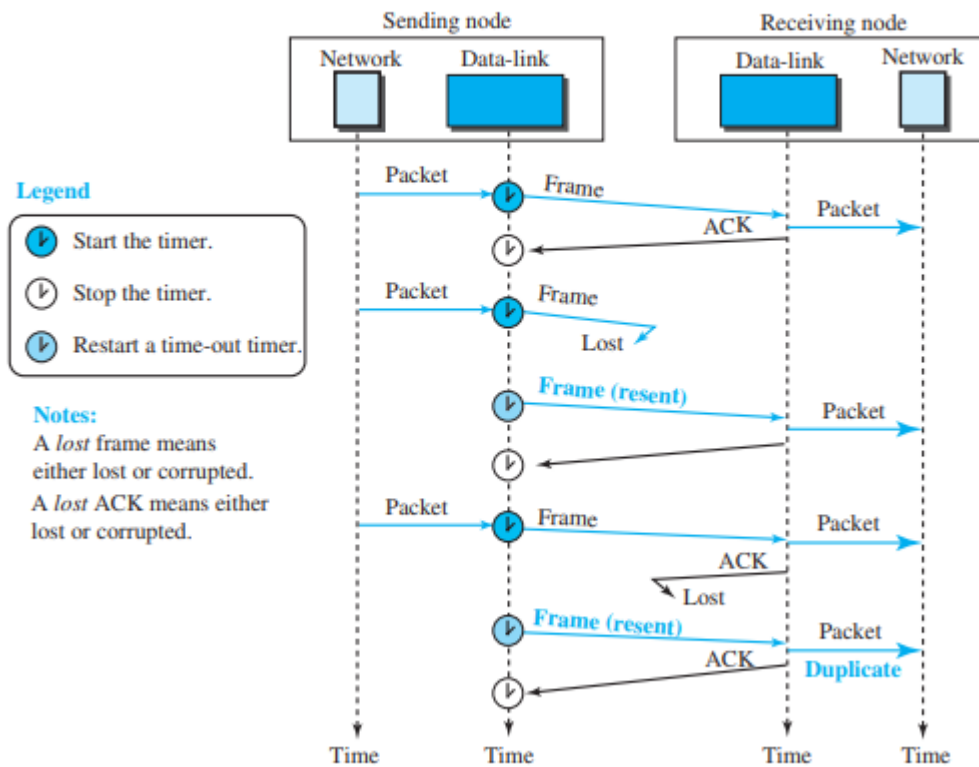


Figure 11.12 shows an example. The first frame is sent and acknowledged. The second frame is sent, but lost. After time-out, it is resent. The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent. However, there is a problem with this scheme. The network layer at the receiver site receives two copies of the third packet, which is not right. In the next section, we will see how we can correct this problem using sequence numbers and acknowledgment numbers.

Sequence and Acknowledgment Numbers

We saw a problem in Example 11.3 that needs to be addressed and corrected. Duplicate packets, as much as corrupted packets, need to be avoided. As an example, assume we are ordering some item online. If each packet defines the specification of an item to be ordered, duplicate packets mean ordering an item more than once. To correct the problem in Example 11.3, we need to add sequence numbers to the data frames and acknowledgment numbers to the ACK frames. However, numbering in this case is very simple. Sequence numbers are 0, 1, 0, 1, 0, 1, ...; the acknowledgment numbers can also be 1, 0, 1, 0, 1, 0, ... In other words, the sequence numbers start with 0, the acknowledgment numbers start with 1. An acknowledgment number always defines the sequence number of the next frame to receive.

Figure 11.13 Flow diagram for Example 11.4

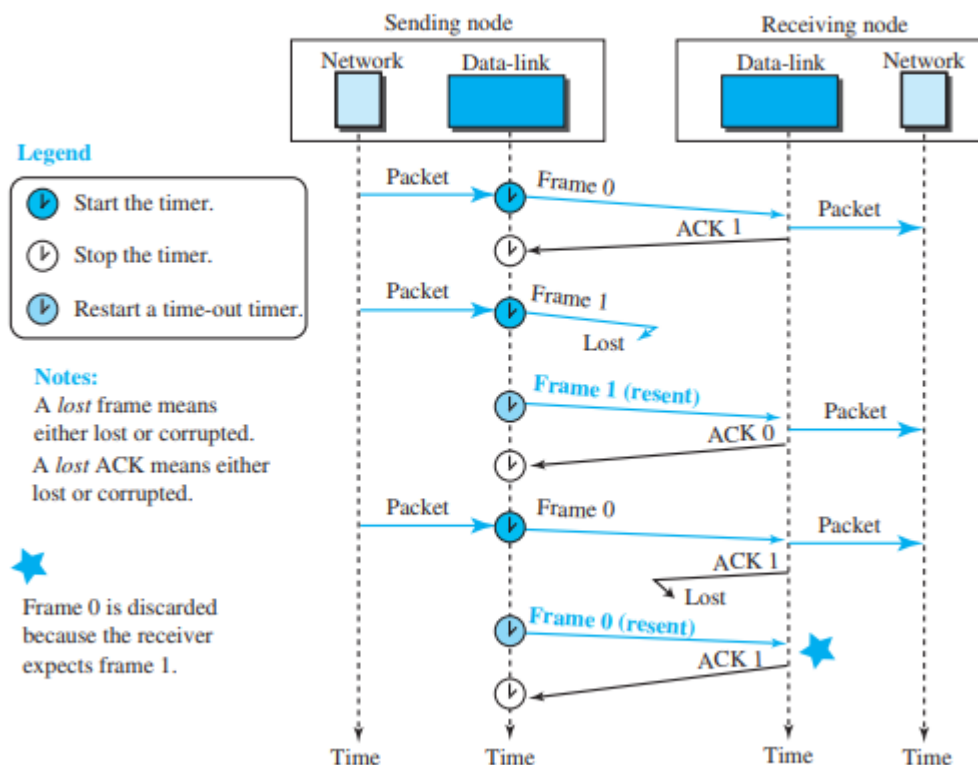


Figure 11.13 shows how adding sequence numbers and acknowledgment numbers can prevent duplicates. The first frame is sent and acknowledged. The second frame is sent, but lost. After time-out, it is resent. The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent.

Piggybacking

The two protocols we discussed in this section are designed for unidirectional communication, in which data is flowing only in one direction although the acknowledgment may travel in the other direction. Protocols have been designed in the past to allow data to flow in both directions. However, to make the communication more efficient, the data in one direction is piggybacked with the acknowledgment in the other direction. In other words, when node A is sending data to node B, Node A also acknowledges the data received from node B.

HDLC

High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links.

Configurations and Transfer Modes

HDLC provides two common transfer modes that can be used in different configurations: normal response mode (NRM) and asynchronous balanced mode (ABM). In normal response mode (NRM), the station configuration is unbalanced. We have one primary

station and multiple secondary stations. A primary station can send commands; a secondary station can only respond. The NRM is used for both point-to-point and multipoint links

Figure 11.14 *Normal response mode*

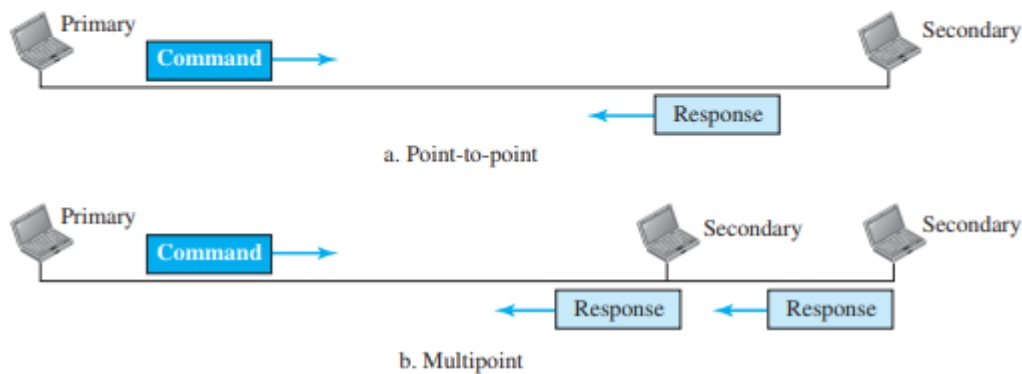


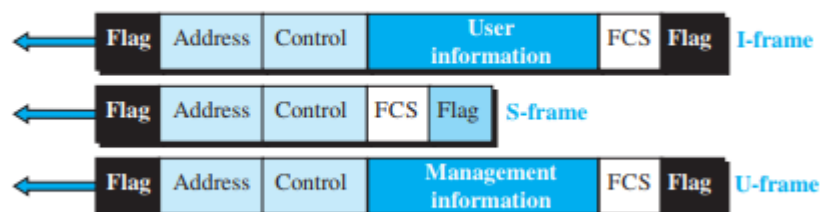
Figure 11.15 *Asynchronous balanced mode*



Framing

To provide the flexibility necessary to support all the options possible in the modes and configurations just described, HDLC defines three types of frames: information frames (I-frames), supervisory frames (S-frames), and unnumbered frames (U-frames). Each type of frame serves as an envelope for the transmission of a different type of message. Iframes are used to data-link user data and control information relating to user data (piggybacking). S-frames are used only to transport control information. U-frames are reserved for system management. Information carried by U-frames is intended for managing the link itself. Each frame in HDLC may contain up to six fields, a beginning flag field, an address field, a control field, an information field, a frame check sequence (FCS) field, and an ending flag field. In multiple-frame transmissions, the ending flag of one frame can serve as the beginning flag of the next frame.

Figure 11.16 *HDLC frames*

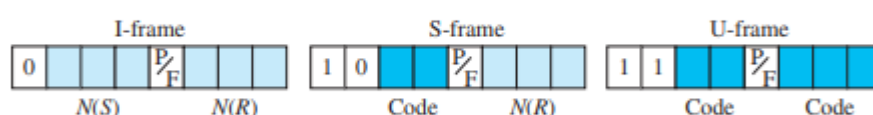


Let us now discuss the fields and their use in different frame types.

- **Flag field.** This field contains synchronization pattern 01111110, which identifies both the beginning and the end of a frame.
- **Address field.** This field contains the address of the secondary station. If a primary station created the frame, it contains a to address. If a secondary station creates the frame, it contains a from address. The address field can be one byte or several bytes long, depending on the needs of the network.
- **Control field.** The control field is one or two bytes used for flow and error control. The interpretation of bits are discussed later.
- **Information field.** The information field contains the user's data from the network layer or management information. Its length can vary from one network to another.
- **FCS field.** The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte CRC.

The control field determines the type of frame and defines its functionality.

Figure 11.17 Control field format for the different frame types



Control Field for I-Frames

I-frames are designed to carry user data from the network layer. In addition, they can include flow- and error-control information (piggybacking). The subfields in the control field are used to define these functions. The first bit defines the type. If the first bit of the control field is 0, this means the frame is an I-frame. The next 3 bits, called $N(S)$, define the sequence number of the frame. Note that with 3 bits, we can define a sequence number between 0 and 7. The last 3 bits, called $N(R)$, correspond to the acknowledgment number when piggybacking is used. The single bit between $N(S)$ and $N(R)$ is called the P/F bit. The P/F field is a single bit with a dual purpose. It has meaning only when it is set (bit = 1) and can mean poll or final. It means poll when the frame is sent by a primary station to a secondary (when the address field contains the address of the receiver). It means final when the frame is sent by a secondary to a primary (when the address field contains the address of the sender).

Control Field for S-Frames

Supervisory frames are used for flow and error control whenever piggybacking is either impossible or inappropriate. S-frames do not have information fields. If the first 2 bits of the control field are 10, this means the frame is an S-frame. The last 3 bits, called $N(R)$, correspond to the acknowledgment number (ACK) or negative acknowledgment number (NAK), depending on the type of S-frame. The 2 bits called code are used to define the type of S-frame itself. With 2 bits, we can have four types of S-frames, as described below:

- **Receive ready (RR).** If the value of the code subfield is 00, it is an RR S-frame
- **Receive not ready (RNR).** If the value of the code subfield is 10, it is an RNR Sframe.

- **Reject (REJ).** If the value of the code subfield is 01, it is an REJ S-frame.
- **Selective reject (SREJ).** If the value of the code subfield is 11, it is an SREJ Sframe.

Control Field for U-Frames

Unnumbered frames are used to exchange session management and control information between connected devices. Unlike S-frames, U-frames contain an information field, but one used for system management information, not user data. As with S-frames, however, much of the information carried by U-frames is contained in codes included in the control field. U-frame codes are divided into two sections: a 2-bit prefix before the P/F bit and a 3-bit suffix after the P/F bit. Together, these two segments (5 bits) can be used to create up to 32 different types of U-frames.

POINT-TO-POINT PROTOCOL (PPP)

One of the most common protocols for point-to-point access is the **Point-to-Point Protocol (PPP)**.

Services Provided by PPP

PPP defines the format of the frame to be exchanged between devices. It also defines how two devices can negotiate the establishment of the link and the exchange of data. PPP is designed to accept payloads from several network layers (not only IP). Authentication is also provided in the protocol, but it is optional. The new version of PPP, called Multilink PPP, provides connections over multiple links. One interesting feature of PPP is that it provides network address configuration.

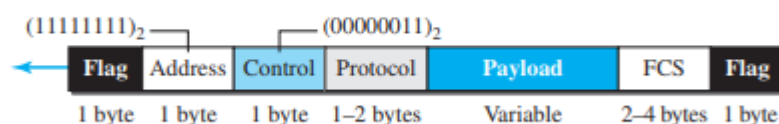
Services Not Provided by PPP

PPP does not provide flow control. A sender can send several frames one after another with no concern about overwhelming the receiver. PPP has a very simple mechanism for error control. A CRC field is used to detect errors. If the frame is corrupted, it is silently discarded; the upper-layer protocol needs to take care of the problem. Lack of error control and sequence numbering may cause a packet to be received out of order. PPP does not provide a sophisticated addressing mechanism to handle frames in a multipoint configuration.

Framing

PPP uses a character-oriented (or byte-oriented) frame.

Figure 11.20 *PPP frame format*



The description of each field follows:

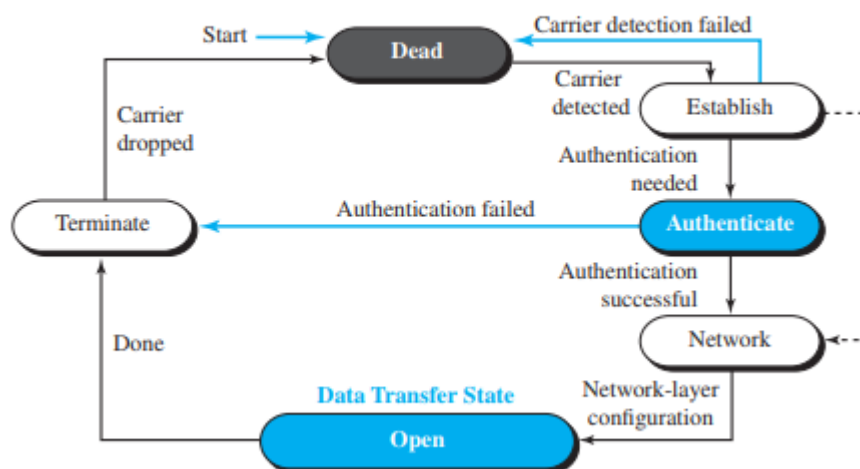
- **Flag.** A PPP frame starts and ends with a 1-byte flag with the bit pattern 01111110.
- **Address.** The address field in this protocol is a constant value and set to 11111111 (broadcast address).

- **Control.** This field is set to the constant value 00000011 (imitating unnumbered frames in HDLC). PPP does not provide any flow control. Error control is also limited to error detection.
- **Protocol.** The protocol field defines what is being carried in the data field: either user data or other information. This field is by default 2 bytes long, but the two parties can agree to use only 1 byte.
- **Payload field.** This field carries either the user data or other information that we will discuss shortly. The data field is a sequence of bytes with the default of a maximum of 1500 bytes; but this can be changed during negotiation. The data field is byte-stuffed if the flag byte pattern appears in this field. Because there is no field defining the size of the data field, padding is needed if the size is less than the maximum default value or the maximum negotiated value.
- **FCS.** The frame check sequence (FCS) is simply a 2-byte or 4-byte standard CRC.

Transition Phases

A PPP connection goes through phases which can be shown in a transition phase diagram (see Figure 11.21). The transition diagram, which is an FSM, starts with the dead state. In this state, there is no active carrier (at the physical layer) and the line is quiet. When one of the two nodes starts the communication, the connection goes into the establish state. In this state, options are negotiated between the two parties. If the two parties agree that they need authentication (for example, if they do not know each other), then the system needs to do authentication (an extra step); otherwise, the parties can simply start communication. The link-control protocol packets, discussed shortly, are used for this purpose. Several packets may be exchanged here. Data transfer takes place in the open state. When a connection reaches this state, the exchange of data packets can be started. The connection remains in this state until one of the endpoints wants to terminate the connection. In this case, the system goes to the terminate state. The system remains in this state until the carrier (physical-layer signal) is dropped, which moves the system to the dead state again.

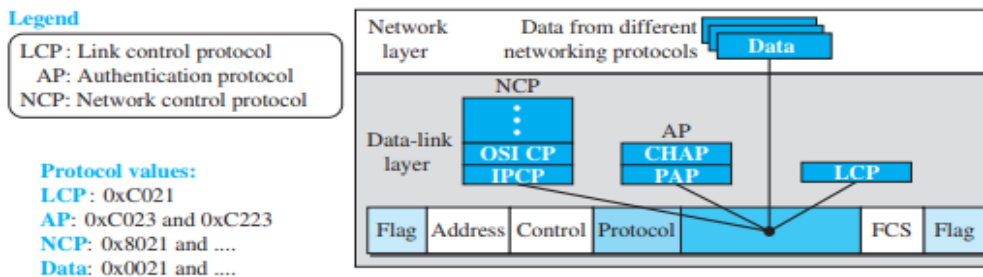
Figure 11.21 Transition phases



Multiplexing

Although PPP is a link-layer protocol, it uses another set of protocols to establish the link, authenticate the parties involved, and carry the network-layer data. Three sets of protocols are defined to make PPP powerful: the Link Control Protocol (LCP), two Authentication Protocols (APs), and several Network Control Protocols (NCPs). At any moment, a PPP packet can carry data from one of these protocols in its data field.

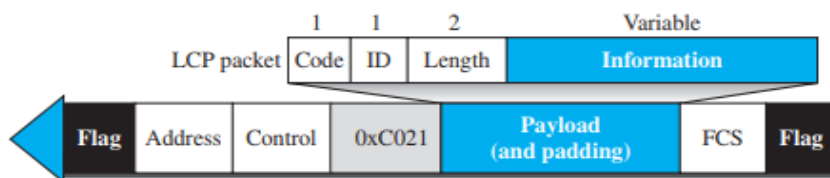
Figure 11.22 Multiplexing in PPP



Link Control Protocol

The Link Control Protocol (LCP) is responsible for establishing, maintaining, configuring, and terminating links. All LCP packets are carried in the payload field of the PPP frame with the protocol field set to C021 in hexadecimal.

Figure 11.23 LCP packet encapsulated in a frame



The code field defines the type of LCP packet.

Table 11.1 LCP packets

Code	Packet Type	Description
0x01	Configure-request	Contains the list of proposed options and their values
0x02	Configure-ack	Accepts all options proposed
0x03	Configure-nak	Announces that some options are not acceptable
0x04	Configure-reject	Announces that some options are not recognized
0x05	Terminate-request	Request to shut down the line
0x06	Terminate-ack	Accept the shutdown request
0x07	Code-reject	Announces an unknown code
0x08	Protocol-reject	Announces an unknown protocol
0x09	Echo-request	A type of hello message to check if the other end is alive
0x0A	Echo-reply	The response to the echo-request message
0x0B	Discard-request	A request to discard the packet

The ID field holds a value that matches a request with a reply. One endpoint inserts a value in this field, which will be copied into the reply packet. The length field defines the length of the entire LCP packet. The information field contains information, such as options, needed for some LCP packets.

Authentication Protocols

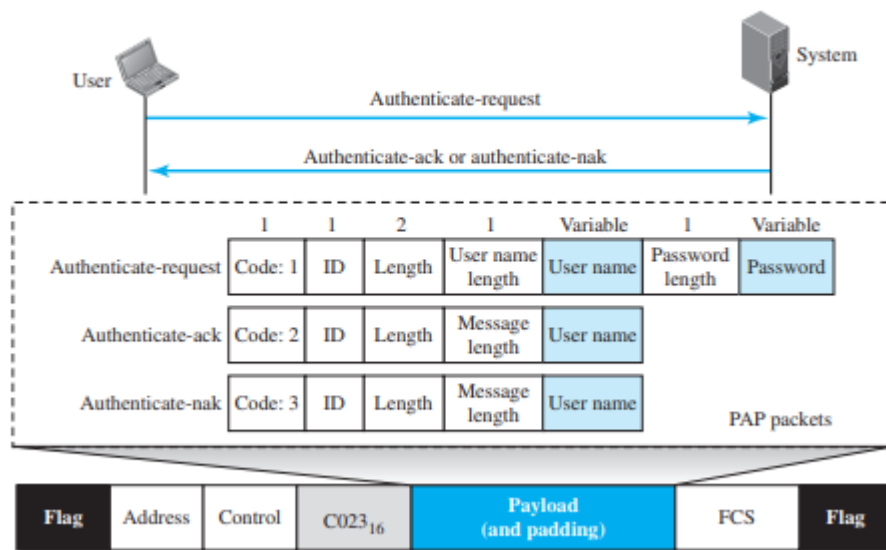
Authentication plays a very important role in PPP because PPP is designed for use over dial-up links where verification of user identity is necessary. Authentication means validating the identity of a user who needs to access a set of resources. PPP has created two protocols for authentication: Password Authentication Protocol and Challenge Handshake Authentication Protocol.

PAP

The **Password Authentication Protocol (PAP)** is a simple authentication procedure with a two-step process:

- The user who wants to access a system sends an authentication identification (usually the user name) and a password.
- The system checks the validity of the identification and password and either accepts or denies connection.

Figure 11.24 PAP packets encapsulated in a PPP frame



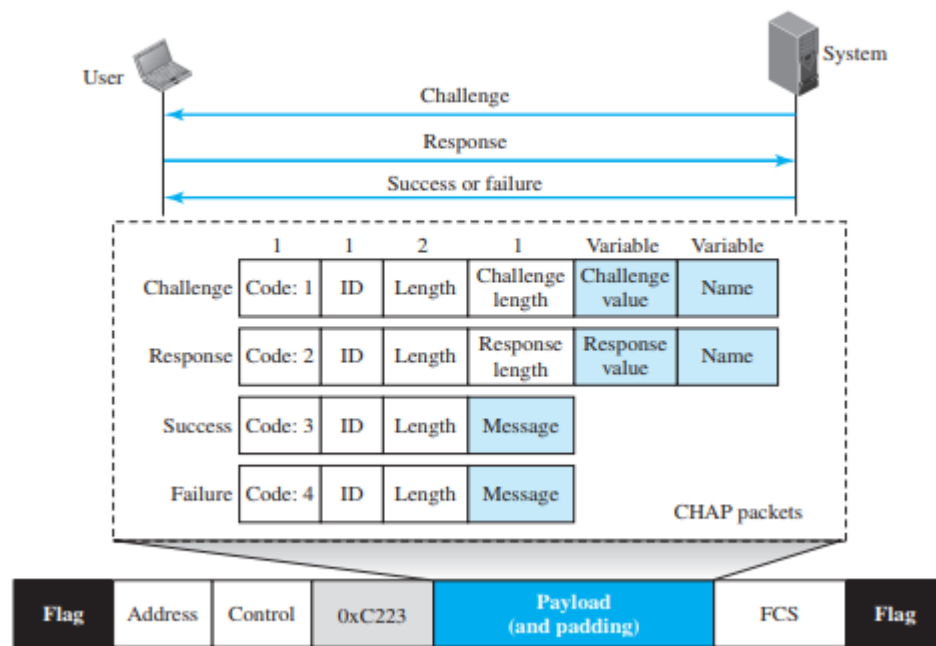
CHAP

The **Challenge Handshake Authentication Protocol (CHAP)** is a three-way handshaking authentication protocol that provides greater security than PAP. In this method, the password is kept secret; it is never sent online.

- The system sends the user a challenge packet containing a challenge value, usually a few bytes.
- The user applies a predefined function that takes the challenge value and the user's own password and creates a result. The user sends the result in the response packet to the system.

- c. The system does the same. It applies the same function to the password of the user (known to the system) and the challenge value to create a result. If the result created is the same as the result sent in the response packet, access is granted; otherwise, it is denied. CHAP is more secure than PAP, especially if the system continuously changes the challenge value. Even if the intruder learns the challenge value and the result, the password is still secret.

Figure 11.25 CHAP packets encapsulated in a PPP frame

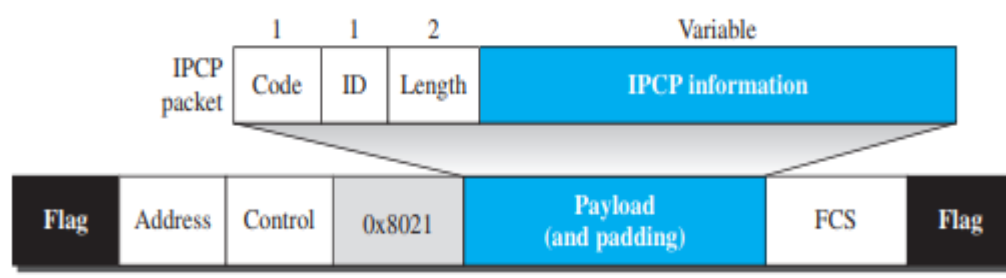


CHAP packets are encapsulated in the PPP frame with the protocol value C223 in hexadecimal. There are four CHAP packets: challenge, response, success, and failure. The first packet is used by the system to send the challenge value. The second is used by the user to return the result of the calculation. The third is used by the system to allow access to the system. The fourth is used by the system to deny access to the system.

IPCP

One NCP protocol is the **Internet Protocol Control Protocol (IPCP)**. This protocol configures the link used to carry IP packets in the Internet. IPCP is especially of interest to us.

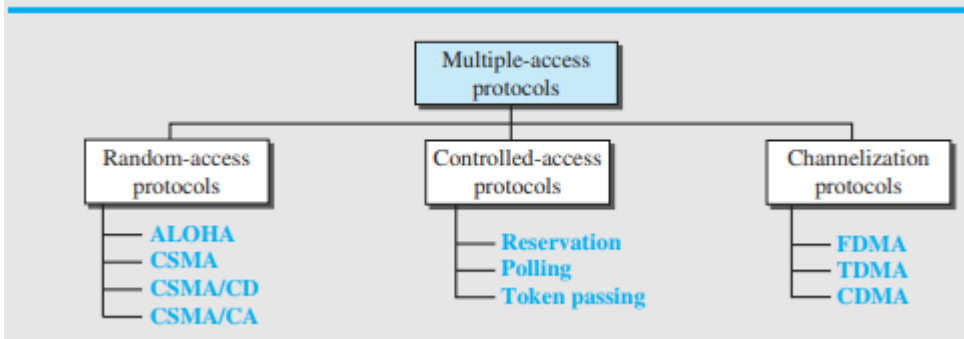
Figure 11.26 IPCP packet encapsulated in PPP frame



MEDIA ACCESS CONTROL

When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link.

Figure 12.1 *Taxonomy of multiple-access protocols*



RANDOM ACCESS

In random-access or contention methods, no station is superior to another station and none is assigned control over another. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.

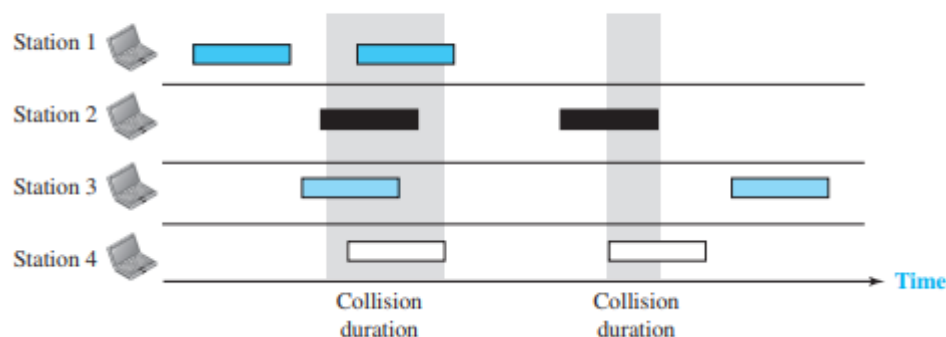
Two features give this method its name. First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called random access. Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called contention methods.

ALOHA

Pure ALOHA

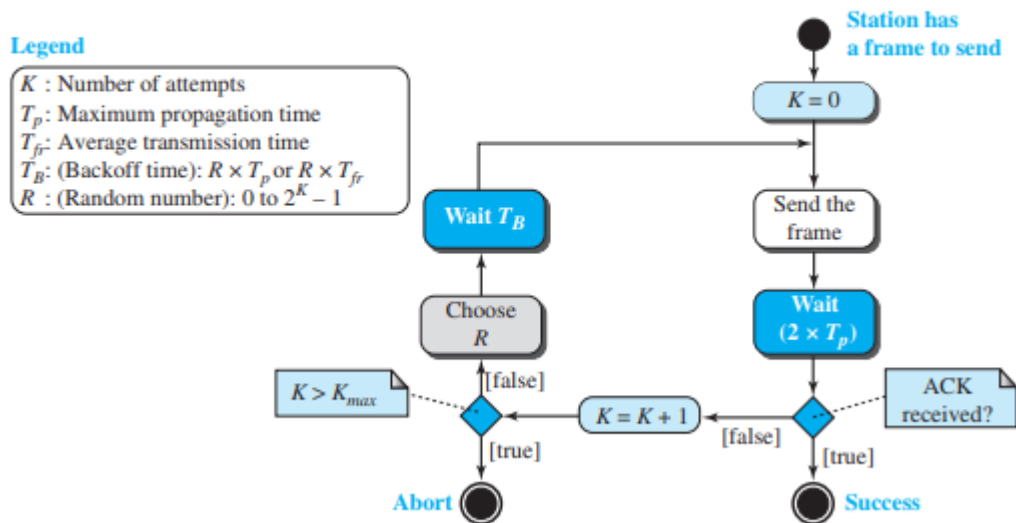
The original ALOHA protocol is called pure ALOHA. This is a simple but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send (multiple access). However, since there is only one channel to share, there is the possibility of collision between frames from different stations.

Figure 12.2 *Frames in a pure ALOHA network*



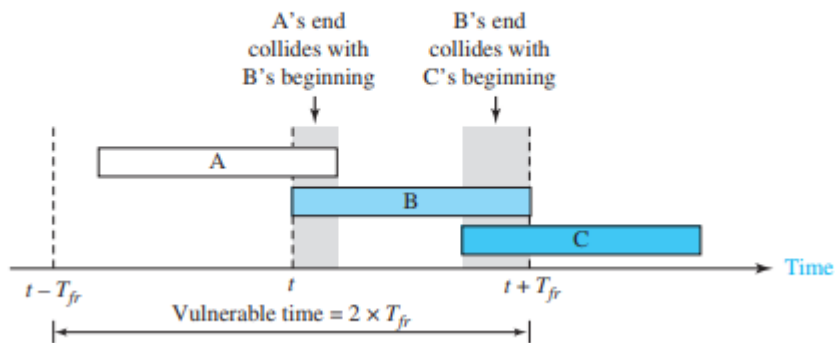
A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. We call this time the backoff time TB.

Figure 12.3 Procedure for pure ALOHA protocol



Vulnerable time

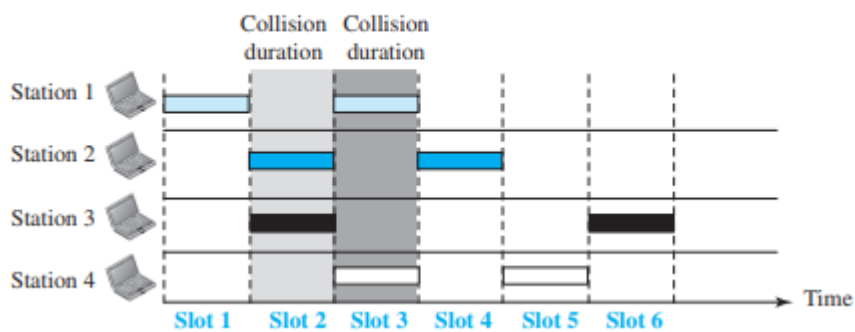
Figure 12.4 Vulnerable time for pure ALOHA protocol



Slotted ALOHA

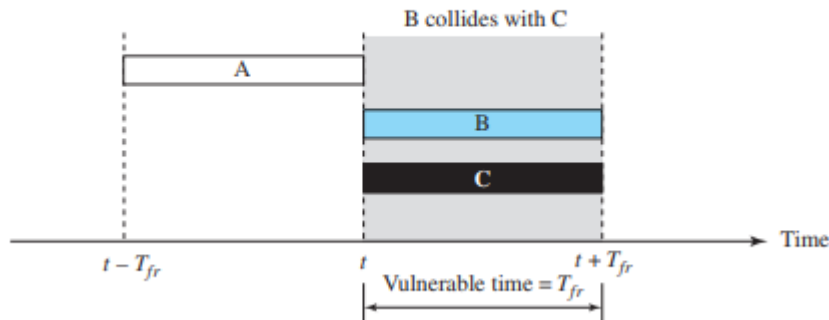
Pure ALOHA has a vulnerable time of $2 \times T_{fr}$. This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or just before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA. In slotted ALOHA we divide the time into slots of T_{fr} seconds and force the station to send only at the beginning of the time slot.

Figure 12.5 Frames in a slotted ALOHA network



Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to T_{fr}

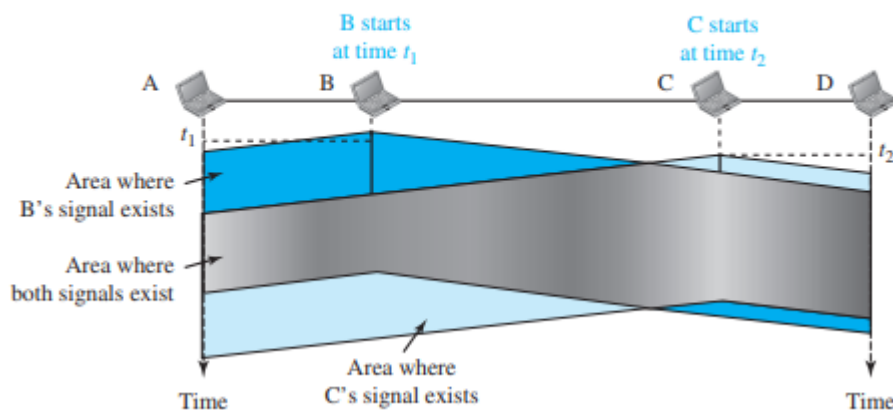
Figure 12.6 Vulnerable time for slotted ALOHA protocol



CSMA

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle “sense before transmit” or “listen before talk.”

Figure 12.7 Space/time model of a collision in CSMA



Persistence Methods

What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions: the 1-persistent method, the nonpersistent method, and the p-persistent method.

Figure 12.9 Behavior of three persistence methods

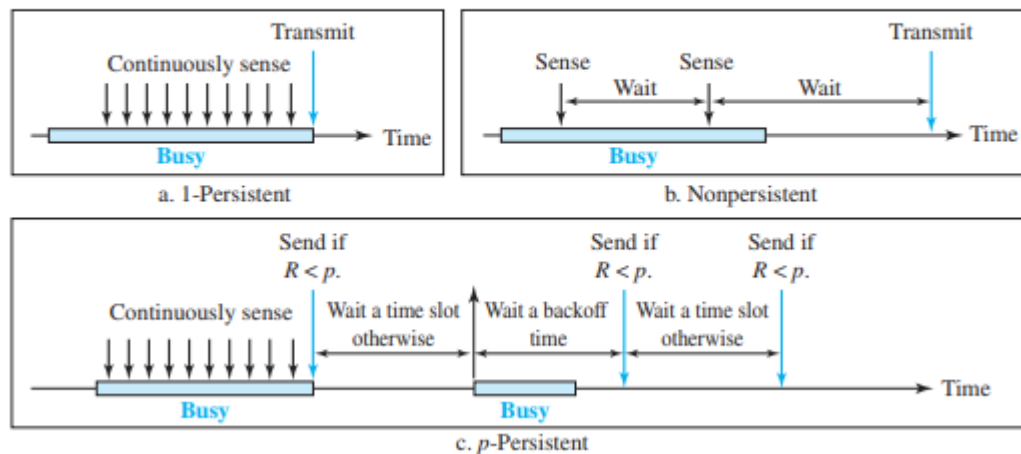
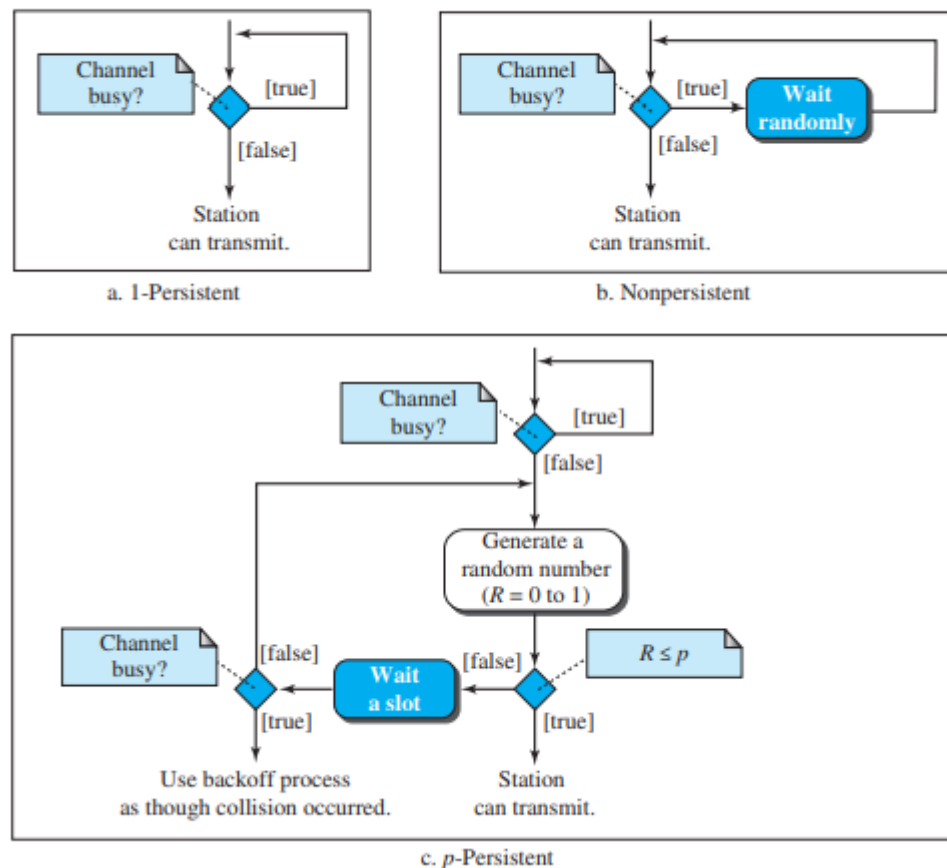


Figure 12.10 Flow diagram for three persistence methods



1-Persistent

The 1-persistent method is simple and straightforward. In this method, after the station finds the line idle, it sends its frame immediately (with probability 1). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.

Nonpersistent

In the nonpersistent method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. The nonpersistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

p-Persistent

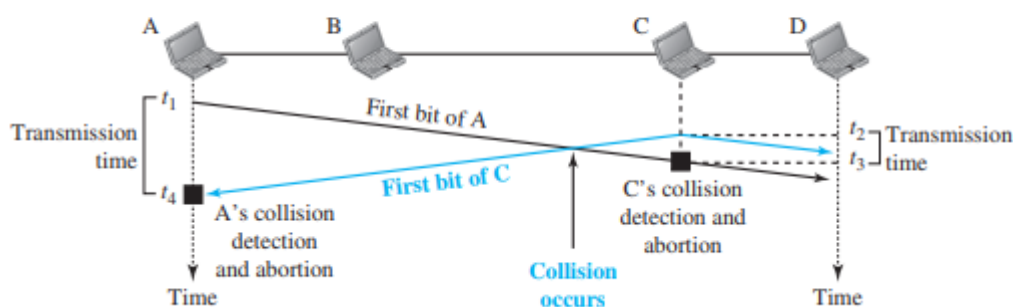
The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency. In this method, after the station finds the line idle it follows these steps:

1. With probability p , the station sends its frame.
2. With probability $q = 1 - p$, the station waits for the beginning of the next time slot and checks the line again.
 - a. If the line is idle, it goes to step 1.
 - b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

CSMA/CD

The CSMA method does not specify the procedure following a collision. Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision. In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again.

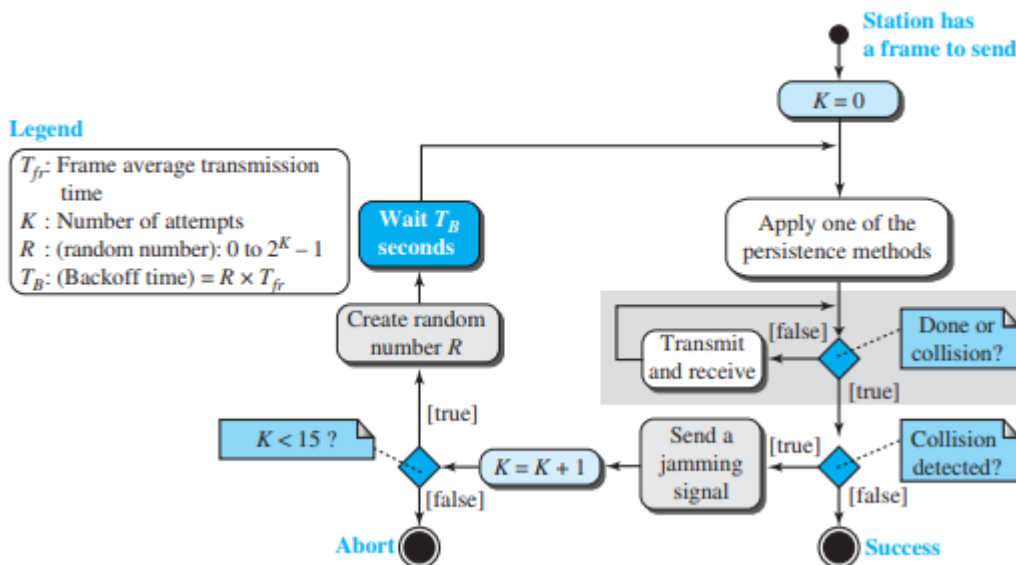
Figure 12.11 Collision of the first bits in CSMA/CD



It is similar to the one for the ALOHA protocol, but there are differences.

- a. The first difference is the addition of the persistence process.
- b. The second difference is the frame transmission.
- c. The third difference is the sending of a short jamming signal to make sure that all other stations become aware of the collision.

Figure 12.13 Flow diagram for the CSMA/CD

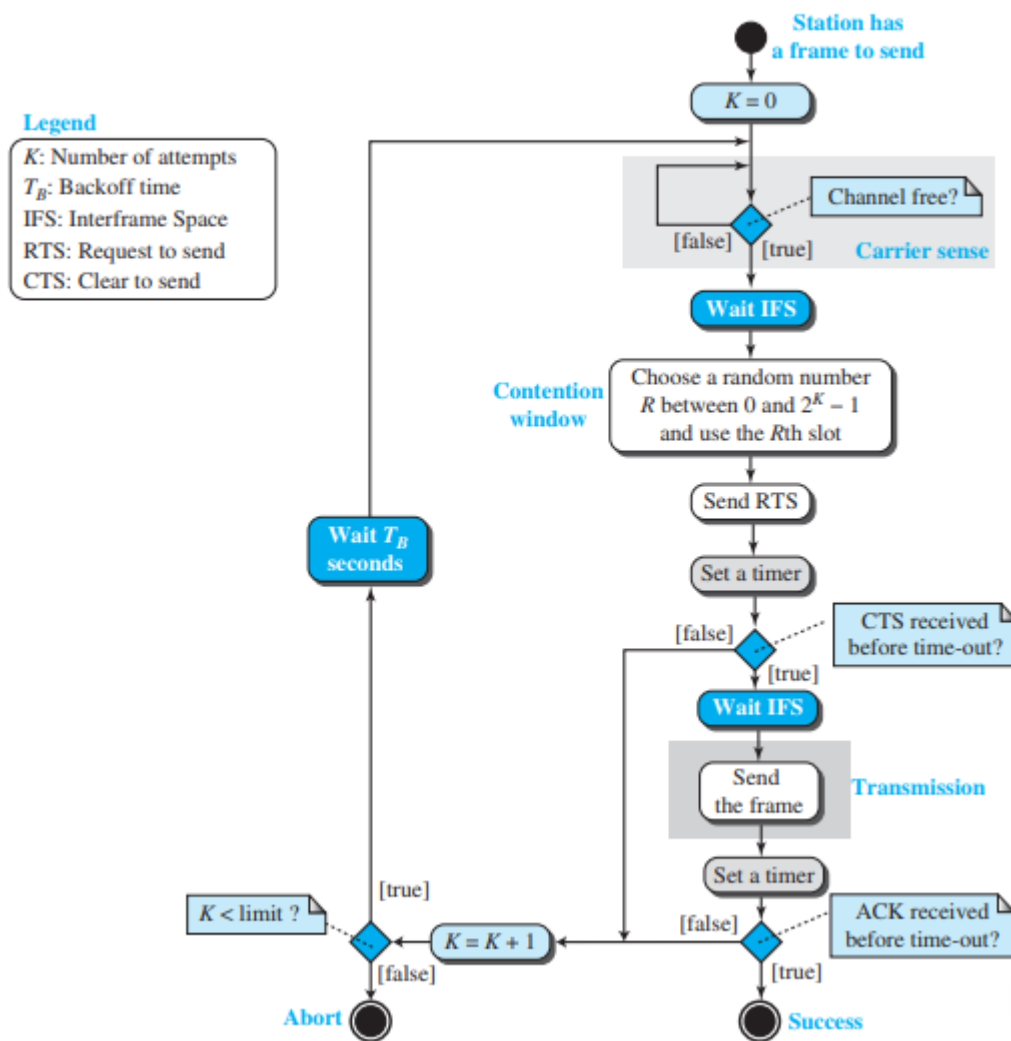


CSMA/CA

Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for wireless networks. Collisions are avoided through the use of CSMA/CA's three strategies: the interframe space, the contention window, and acknowledgments.

- **Interframe Space (IFS).** First, collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the interframe space or IFS.
- **Contention Window.** The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential backoff strategy.
- **Acknowledgment.** With all these precautions, there still may be a collision resulting in destroyed data. In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.

Figure 12.15 Flow diagram of CSMA/CA



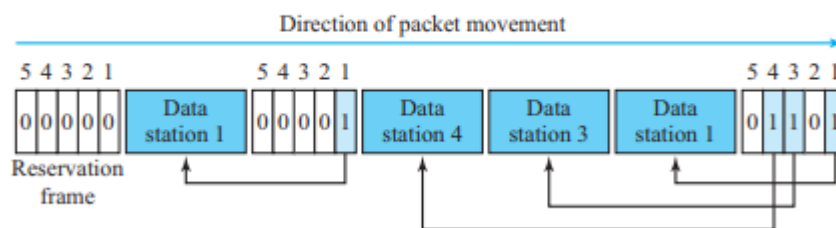
CONTROLLED ACCESS

In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations.

Reservation

In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval. If there are N stations in the system, there are exactly N reservation minislots in the reservation frame. Each minislot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own minislot. The stations that have made reservations can send their data frames after the reservation frame.

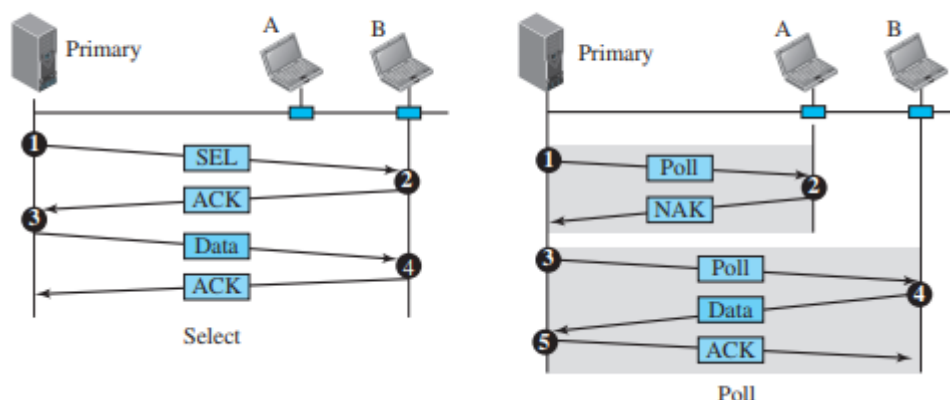
Figure 12.18 *Reservation access method*



Polling

Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations. All data exchanges must be made through the primary station even when the ultimate destination is a secondary device. The primary device controls the link; the secondary devices follow its instructions. It is up to the primary device to determine which device is allowed to use the channel at a given time. The primary device, therefore, is always the initiator of a session. This method uses poll and select functions to prevent collisions. However, the drawback is if the primary station fails, the system goes down.

Figure 12.19 *Select and poll functions in polling-access method*



Select

The select function is used whenever the primary device has something to send

Poll

The poll function is used by the primary device to solicit transmissions from the secondary devices.

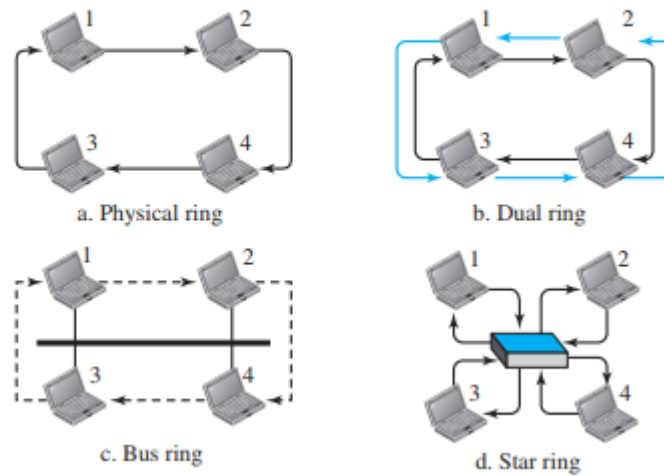
Token Passing

In the token-passing method, the stations in a network are organized in a logical ring. In other words, for each station, there is a predecessor and a successor. The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring. The current station is the one that is accessing the channel now. The right to this access has been passed from the predecessor to the current station. The right will be passed to the successor when the current station has no more data to send.

Logical Ring

In a token-passing network, stations do not have to be physically connected in a ring; the ring can be a logical one.

Figure 12.20 Logical ring and physical topology in token-passing access method



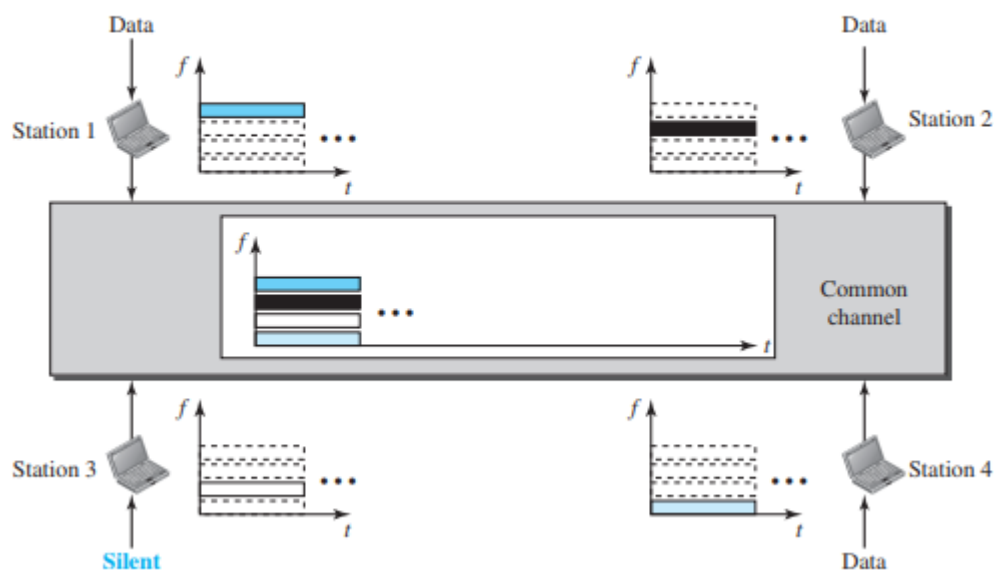
CHANNELIZATION

Channelization (or channel partition, as it is sometimes called) is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, among different stations. In this section, we discuss three channelization protocols: **FDMA**, **TDMA**, and **CDMA**.

FDMA

In **frequency-division multiple access (FDMA)**, the available bandwidth is divided into frequency bands. Each station is allocated a band to send its data. In other words, each band is reserved for a specific station, and it belongs to the station all the time. Each station also uses a bandpass filter to confine the transmitter frequencies. To prevent station interferences, the allocated bands are separated from one another by small guard bands.

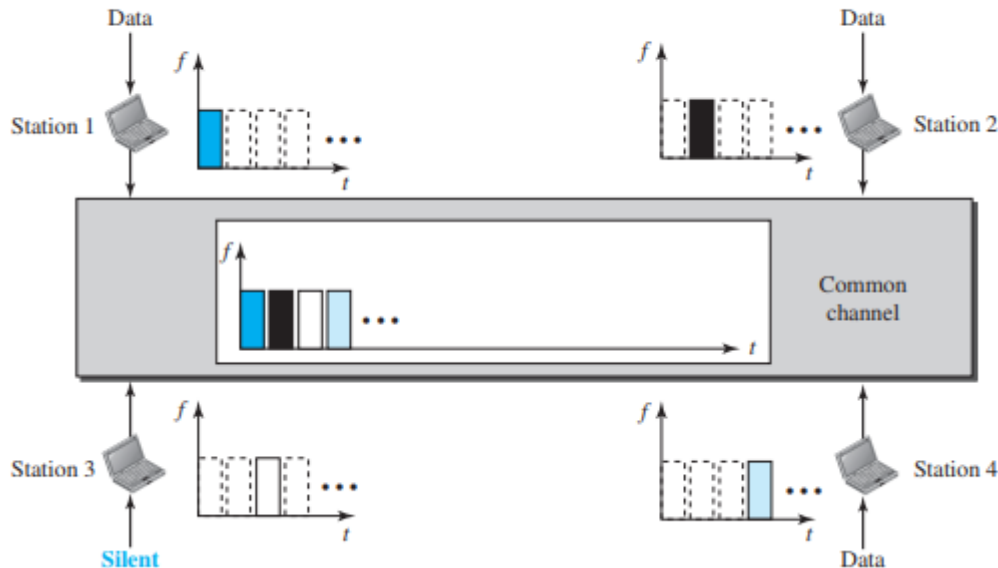
Figure 12.21 Frequency-division multiple access (FDMA)



TDMA

In **time-division multiple access (TDMA)**, the stations share the bandwidth of the channel in time. Each station is allocated a time slot during which it can send data. Each station transmits its data in its assigned time slot.

Figure 12.22 *Time-division multiple access (TDMA)*



The main problem with TDMA lies in achieving synchronization between the different stations. Each station needs to know the beginning of its slot and the location of its slot. This may be difficult because of propagation delays introduced in the system if the stations are spread over a large area. To compensate for the delays, we can insert guard times. Synchronization is normally accomplished by having some synchronization bits (normally referred to as preamble bits) at the beginning of each slot.

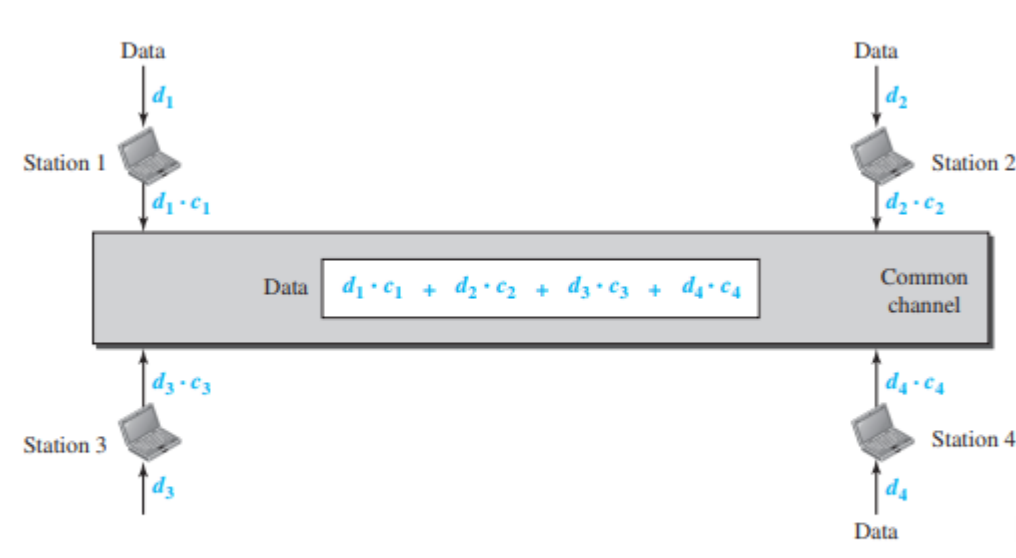
CDMA

Code-division multiple access (CDMA) was conceived several decades ago. Recent advances in electronic technology have finally made its implementation possible. CDMA differs from FDMA in that only one channel occupies the entire bandwidth of the link. It differs from TDMA in that all stations can send data simultaneously; there is no timesharing. CDMA simply means communication with different codes.

Idea

Let us assume we have four stations, 1, 2, 3, and 4, connected to the same channel. The data from station 1 are d_1 , from station 2 are d_2 , and so on. The code assigned to the first station is c_1 , to the second is c_2 , and so on. We assume that the assigned codes have two properties. 1. If we multiply each code by another, we get 0. 2. If we multiply each code by itself, we get 4 (the number of stations). With these two properties in mind, let us see how the above four stations can send data using the same common channel.

Figure 12.23 Simple idea of communication with code

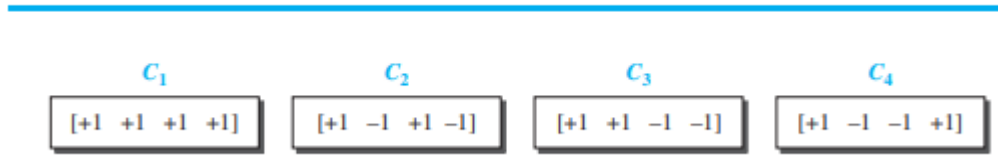


$$\begin{aligned} \text{data} &= (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4) \cdot c_1 \\ &= d_1 \cdot c_1 \cdot c_1 + d_2 \cdot c_2 \cdot c_1 + d_3 \cdot c_3 \cdot c_1 + d_4 \cdot c_4 \cdot c_1 = 4 \times d_1 \end{aligned}$$

Chips

CDMA is based on coding theory. Each station is assigned a code, which is a sequence of numbers called chips

Figure 12.24 Chip sequences



WIRED LANS: ETHERNET

ETHERNET PROTOCOL

In the 1980s and 1990s several different types of LANs were used. All of these LANs used a media-access method to solve the problem of sharing the media. The Ethernet used the CSMA/CD approach. The Token Ring, Token Bus, and FDDI (Fiber Distribution Data Interface) used the token-passing approach. During this period, another LAN technology, ATM LAN, which deployed the high speed WAN technology (ATM), appeared in the market.

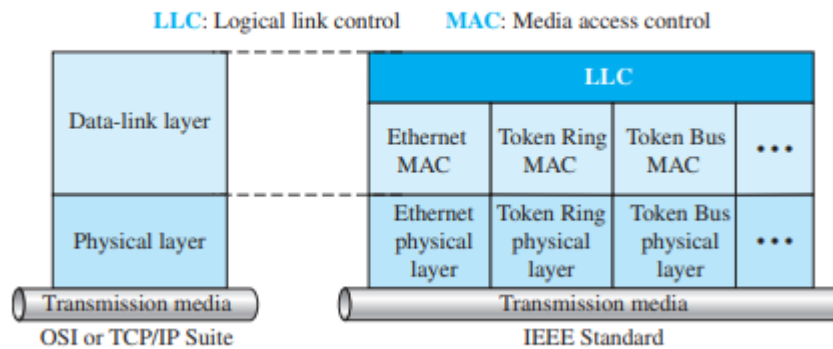
Almost every LAN except Ethernet has disappeared from the marketplace because Ethernet was able to update itself to meet the needs of the time.

It is natural that an organization that has used an Ethernet LAN in the past and now needs a higher data rate would update to the new generation instead of switching to another technology, which might cost more.

IEEE Project 802

In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers. The relationship of the 802 Standard to the TCP/IP protocol suite is shown in Figure.

Figure 13.1 IEEE standard for LANs



The IEEE has subdivided the data-link layer into two sublayers: **logical link control (LLC)** and **media access control (MAC)**. IEEE has also created several physical-layer standards for different LAN protocols.

Logical Link Control (LLC)

In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sublayer called the logical link control (LLC). Framing is handled in both the LLC sublayer and the MAC sublayer.

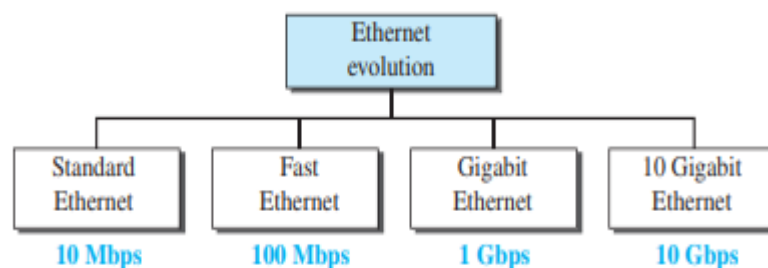
Media Access Control (MAC)

IEEE Project 802 has created a sublayer called media access control that defines the specific access method for each LAN.

Ethernet Evolution

The Ethernet LAN was developed in the 1970s by Robert Metcalfe and David Boggs. Since then, it has gone through four generations: Standard Ethernet (10 Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (1 Gbps), and 10 Gigabit Ethernet (10 Gbps)

Figure 13.2 Ethernet evolution through four generations



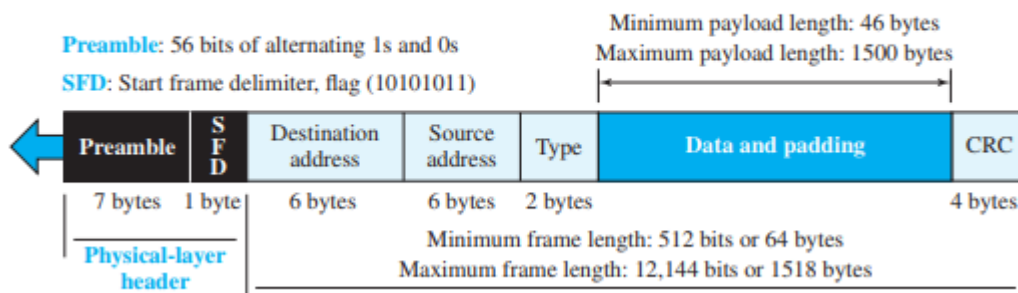
STANDARD ETHERNET

Characteristics

- The data rate of Standard Ethernet is 10 Mbps.
- Ethernet provides a connectionless service, which means each frame sent is independent of the previous or next frame.
- Ethernet is also unreliable like IP and UDP. If a frame is corrupted during transmission and the receiver finds out about the corruption, the receiver drops the frame silently.

Frame Format

Figure 13.3 Ethernet frame



- **Preamble.** This field contains 7 bytes (56 bits) of alternating 0s and 1s that alert the receiving system to the coming frame and enable it to synchronize its clock if it's out of synchronization.
- **Start frame delimiter (SFD).** This field (1 byte: 10101011) signals the beginning of the frame.
- **Destination address (DA).** This field is six bytes (48 bits) and contains the linklayer address of the destination station or stations to receive the packet.
- **Source address (SA).** This field is also six bytes and contains the link-layer address of the sender of the packet.
- **Type.** This field defines the upper-layer protocol whose packet is encapsulated in the frame. This protocol can be IP, ARP, OSPF, and so on.
- **Data.** This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes.
- **CRC.** The last field contains error detection information, in this case a CRC-32. The CRC is calculated over the addresses, types, and data field. If the receiver calculates the CRC and finds that it is not zero (corruption in transmission), it discards the frame.

Frame Length

- Minimum frame length: 64 bytes
- Maximum frame length: 1518 bytes
- Minimum data length: 46 bytes
- Maximum data length: 1500 bytes

Addressing

Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC). The NIC fits inside the station and provides the station with a link-layer address. The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes. For example, the following shows an Ethernet MAC address:

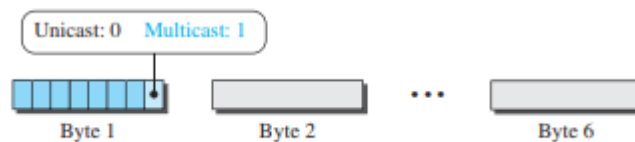
47:20:1B:2E:08:EE

The address is sent left to right, byte by byte; for each byte, it is sent right to left, bit by bit, as shown below:

Hexadecimal	47	20	1B	2E	08	EE
Binary	01000111	00100000	00011011	00101110	00001000	11101110
Transmitted ←	11100010	00000100	11011000	01110100	00010000	01110111

A source address is always a unicast address—the frame comes from only one station. The destination address, however, can be unicast, multicast, or broadcast. If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast.

Figure 13.4 Unicast and multicast addresses



Access Method:

Since the network that uses the standard Ethernet protocol is a broadcast network, we need to use an access method to control access to the sharing medium. The standard Ethernet chose CSMA/CD with 1-persistent method.

FAST ETHERNET (100 MBPS)

In the 1990s, some LAN technologies with transmission rates higher than 10 Mbps, such as FDDI and Fiber Channel, appeared on the market. If the Standard Ethernet wanted to survive, it had to compete with these technologies. Ethernet made a big jump by increasing the transmission rate to 100 Mbps, and the new generation was called the Fast Ethernet. The designers of the Fast Ethernet needed to make it compatible with the Standard Ethernet. The MAC sublayer was left unchanged, which meant the frame format and the maximum and minimum size could also remain unchanged. By increasing the transmission rate, features of the Standard Ethernet that depend on the transmission rate, access method, and implementation had to be reconsidered.

The goals of Fast Ethernet can be summarized as follows:

1. Upgrade the data rate to 100 Mbps.
2. Make it compatible with Standard Ethernet.
3. Keep the same 48-bit address.
4. Keep the same frame format.

Access Method

Operation of the CSMA/CD depends on the transmission rate, the minimum size of the frame, and the maximum network length. If we want to keep the minimum size of the frame, the maximum length of the network should be changed. So the Fast Ethernet came with two solutions:

1. The first solution was to totally drop the bus topology and use a passive hub and star topology but make the maximum size of the network 250 meters instead of 2500 meters as in the Standard Ethernet.
2. The second solution is to use a link-layer switch (discussed later in the chapter) with a buffer to store frames and a full-duplex connection to each host to make the transmission medium private for each host.

Autonegotiation

A new feature added to Fast Ethernet is called autonegotiation. It allows a station or a hub a range of capabilities. Autonegotiation allows two devices to negotiate the mode or data rate of operation. It was designed particularly to allow incompatible devices to connect to one another. We can summarize the goal of autonegotiation as follows. It was designed particularly for these purposes:

- To allow incompatible devices to connect to one another. For example, a device with a maximum capacity of 10 Mbps can communicate with a device with a 100 Mbps capacity (but which can work at a lower rate).
- To allow one device to have multiple capabilities.
- To allow a station to check a hub's capabilities.

Physical Layer

To be able to handle a 100 Mbps data rate, several changes need to be made at the physical layer.

Topology

Fast Ethernet is designed to connect two or more stations. If there are only two stations, they can be connected point-to-point. Three or more stations need to be connected in a star topology with a hub or a switch at the center.

Encoding

Manchester encoding needs a 200-Mbaud bandwidth for a data rate of 100 Mbps, which makes it unsuitable for a medium such as twisted-pair cable. For this reason, the Fast Ethernet designers sought some alternative encoding/decoding scheme. However, it was found that one scheme would not perform equally well for all three implementations. Therefore, three different encoding schemes were chosen.

1. 100Base-TX uses two pairs of twisted-pair cable.
2. 100Base-FX uses two pairs of fiber-optic cables.
3. 100Base-T4, was designed to use category 3 or higher UTP.

Table 13.2 Summary of Fast Ethernet implementations

Implementation	Medium	Medium Length	Wires	Encoding
100Base-TX	UTP or STP	100 m	2	4B5B + MLT-3
100Base-FX	Fiber	185 m	2	4B5B + NRZ-I
100Base-T4	UTP	100 m	4	Two 8B/6T

GIGABIT ETHERNET

The need for an even higher data rate resulted in the design of the Gigabit Ethernet Protocol (1000 Mbps). The IEEE committee calls it the Standard 802.3z. The goals of the Gigabit Ethernet were to upgrade the data rate to 1 Gbps, but keep the address length, the frame format, and the maximum and minimum frame length the same. The goals of the Gigabit Ethernet design can be summarized as follows:

1. Upgrade the data rate to 1 Gbps.
2. Make it compatible with Standard or Fast Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format.
5. Keep the same minimum and maximum frame lengths.
6. Support autonegotiation as defined in Fast Ethernet.

MAC Sublayer

A main consideration in the evolution of Ethernet was to keep the MAC sublayer untouched. However, to achieve a data rate of 1 Gbps, this was no longer possible. Gigabit Ethernet has two distinctive approaches for medium access: **half-duplex and full-duplex**.

Full-Duplex Mode

In the full-duplex mode of Gigabit Ethernet, there is no collision; the maximum length of the cable is determined by the signal attenuation in the cable.

Half-Duplex Mode

Gigabit Ethernet can also be used in half-duplex mode, although it is rare. In this case, a switch can be replaced by a hub, which acts as the common cable in which a collision might occur. The half-duplex approach uses CSMA/CD.

Three methods have been defined:

1. traditional,
2. carrier extension, and
3. frame bursting.

Physical Layer

The physical layer in Gigabit Ethernet is more complicated than that in Standard or Fast Ethernet.

Topology

Gigabit Ethernet is designed to connect two or more stations. If there are only two stations, they can be connected point-to-point. Three or more stations need to be connected in a star topology with a hub or a switch at the center. Another possible configuration is to connect several star topologies or let one star topology be part of another.

Implementation

Gigabit Ethernet can be categorized as either a two-wire or a four-wire implementation. The two-wire implementations use fiber-optic cable (1000Base-SX, short-wave, or 1000Base-LX,

long-wave), or STP (1000Base-CX). The four-wire version uses category 5 twisted-pair cable (1000Base-T).

Table 13.3 *Summary of Gigabit Ethernet implementations*

Implementation	Medium	Medium Length	Wires	Encoding
1000Base-SX	Fiber S-W	550 m	2	8B/10B + NRZ
1000Base-LX	Fiber L-W	5000 m	2	8B/10B + NRZ
1000Base-CX	STP	25 m	2	8B/10B + NRZ
1000Base-T4	UTP	100 m	4	4D-PAM5

10 GIGABIT ETHERNET

In recent years, there has been another look into the Ethernet for use in metropolitan areas. The idea is to extend the technology, the data rate, and the coverage distance so that the Ethernet can be used as LAN and MAN (metropolitan area network). The IEEE committee created 10 Gigabit Ethernet and called it Standard 802.3ae. The goals of the 10 Gigabit Ethernet design can be summarized as upgrading the data rate to 10 Gbps, keeping the same frame size and format, and allowing the interconnection of LANs, MANs, and WAN possible. This data rate is possible only with fiber-optic technology at this time. The standard defines two types of physical layers: LAN PHY and WAN PHY. The first is designed to support existing LANs; the second actually defines a WAN with links connected through SONET OC-192.

Implementation

10 Gigabit Ethernet operates only in full-duplex mode, which means there is no need for contention; CSMA/CD is not used in 10 Gigabit Ethernet. Four implementations are the most common: 10GBase-SR, 10GBase-LR, 10GBase-EW, and 10GBase-X4.

Table 13.4 *Summary of 10 Gigabit Ethernet implementations*

Implementation	Medium	Medium Length	Number of wires	Encoding
10GBase-SR	Fiber 850 nm	300 m	2	64B66B
10GBase-LR	Fiber 1310 nm	10 Km	2	64B66B
10GBase-EW	Fiber 1350 nm	40 Km	2	SONET
10GBase-X4	Fiber 1310 nm	300 m to 10 Km	2	8B10B

WIRELESS LANS

There are two types of wireless LANs.

- The first is the wireless LAN defined by the IEEE 802.11 project (sometimes called wireless Ethernet);
- The second is a personal wireless LAN, Bluetooth, that is sometimes called personal area network or PAN.

INTRODUCTION

Wireless communication is one of the fastest-growing technologies. The demand for connecting devices without the use of cables is increasing everywhere. Wireless LANs can be found on college campuses, in office buildings, and in many public areas.

Architectural Comparison

Medium

The first difference we can see between a wired and a wireless LAN is the medium. In a wired LAN, we use wires to connect hosts. In a switched LAN, with a link-layer switch, the communication between the hosts is point-to-point and full-duplex (bidirectional). In a wireless LAN, the medium is air, the signal is generally broadcast.

Hosts

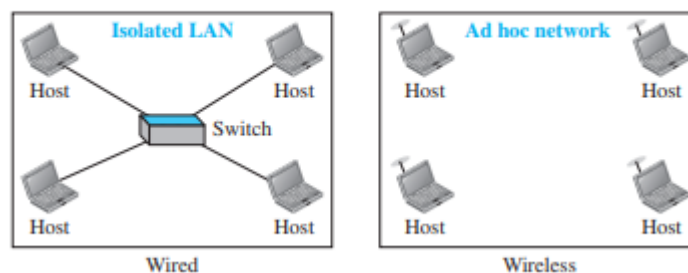
In a wired LAN, a host is always connected to its network at a point with a fixed link-layer address related to its network interface card (NIC).

In a wireless LAN, a host is not physically connected to the network; it can move freely (as we'll see) and can use the services provided by the network.

Isolated LANs

The concept of a wired isolated LAN also differs from that of a wireless isolated LAN. A wired isolated LAN is a set of hosts connected via a link-layer switch (in the recent generation of Ethernet). A wireless isolated LAN, called an ad hoc network in wireless LAN terminology, is a set of hosts that communicate freely with each other.

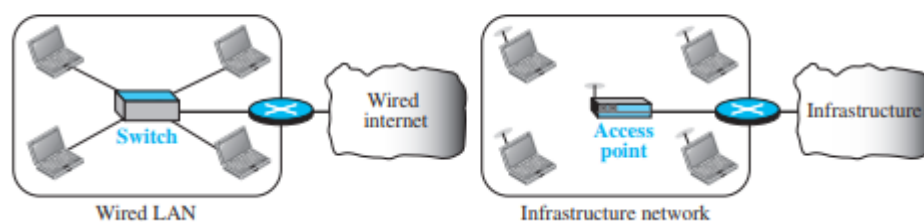
Figure 15.1 *Isolated LANs: wired versus wireless*



Connection to Other Networks

A wired LAN can be connected to another network or an internetwork such as the Internet using a router. A wireless LAN may be connected to a wired infrastructure network, to a wireless infrastructure network, or to another wireless LAN.

Figure 15.2 *Connection of a wired LAN and a wireless LAN to other networks*



In this case, the wireless LAN is referred to as an infrastructure network, and the connection to the wired infrastructure, such as the Internet, is done via a device called an access point (AP). Note that the role of the access point is completely different from the role of a link-layer switch in the wired environment. An access point is gluing two different environments together: one wired and one wireless. Communication between the AP and the wireless host occurs in a wireless environment; communication between the AP and the infrastructure occurs in a wired environment.

Characteristics

- **Attenuation** The strength of electromagnetic signals decreases rapidly because the signal disperses in all directions; only a small portion of it reaches the receiver. The situation becomes worse with mobile senders that operate on batteries and normally have small power supplies.
- **Interference** Another issue is that a receiver may receive signals not only from the intended sender, but also from other senders if they are using the same frequency band.
- **Multipath Propagation** A receiver may receive more than one signal from the same sender because electromagnetic waves can be reflected back from obstacles such as walls, the ground, or objects. The result is that the receiver receives some signals at different phases (because they travel different paths). This makes the signal less recognizable.
- **Error** With the above characteristics of a wireless network, we can expect that errors and error detection are more serious issues in a wireless network than in a wired network. If we think about the error level as the measurement of signal-to-noise ratio (SNR), we can better understand why error detection and error correction and retransmission are more important in a wireless network.

Access Control

In this method, each host contends to access the medium and sends its frame if it finds the medium idle. If a collision occurs, it is detected and the frame is sent again. Collision detection in CSMA/CD serves two purposes. If a collision is detected, it means that the frame has not been received and needs to be resent. If a collision is not detected, it is a kind of acknowledgment that the frame was received. The CSMA/CD algorithm does not work in wireless LANs for three reasons:

1. To detect a collision, a host needs to send and receive at the same time (sending the frame and receiving the collision signal), which means the host needs to work in a duplex mode. Wireless hosts do not have enough power to do so (the power is supplied by batteries). They can only send or receive at one time.
2. Because of the hidden station problem, in which a station may not be aware of another station's transmission due to some obstacles or range problems, collision may occur but not be detected.
3. The distance between stations can be great. Signal fading could prevent a station at one end from hearing a collision at the other end.

To overcome the above three problems, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) was invented for wireless LANs.

IEEE 802.11

IEEE has defined the specifications for a wireless LAN, called IEEE 802.11, which covers the physical and data-link layers. It is sometimes called wireless Ethernet.

In some countries, including the United States, the public uses the term WiFi (short for wireless fidelity) as a synonym for wireless LAN.

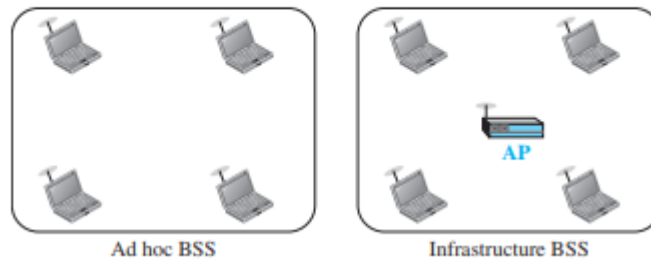
Architecture:

The standard defines two kinds of services: the **basic service set (BSS)** and the **extended service set (ESS)**.

Basic Service Set

IEEE 802.11 defines the basic service set (BSS) as the building blocks of a wireless LAN. A basic service set is made of stationary or mobile wireless stations and an optional central base station, known as the access point (AP).

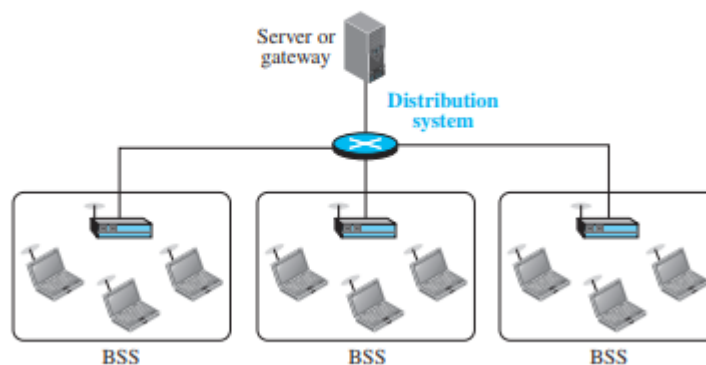
Figure 15.4 Basic service sets (BSSs)



The BSS without an AP is a stand-alone network and cannot send data to other BSSs. It is called an ad hoc architecture. In this architecture, stations can form a network without the need of an AP; they can locate one another and agree to be part of a BSS. A BSS with an AP is sometimes referred to as an infrastructure BSS.

Extended Service Set

Figure 15.5 Extended service set (ESS)



An extended service set (ESS) is made up of two or more BSSs with APs. In this case, the BSSs are connected through a distribution system, which is a wired or a wireless network. The distribution system connects the APs in the BSSs. IEEE 802.11 does not restrict the distribution system; it can be any IEEE LAN such as an Ethernet. Note that the extended service set uses two types of stations: mobile and stationary. The mobile stations are normal stations inside a BSS. The stationary stations are AP stations that are part of a wired LAN.

Station Types

IEEE 802.11 defines three types of stations based on their mobility in a wireless LAN: **no-transition**, **BSS-transition**, and **ESS-transition** mobility. A station with no-transition mobility is either stationary (not moving) or moving only inside a BSS. A station with BSS-transition mobility can move from one BSS to another, but the movement is confined inside one ESS. A station with ESS-transition mobility can move from one ESS to another. However, IEEE 802.11 does not guarantee that communication is continuous during the move.

MAC Sublayer

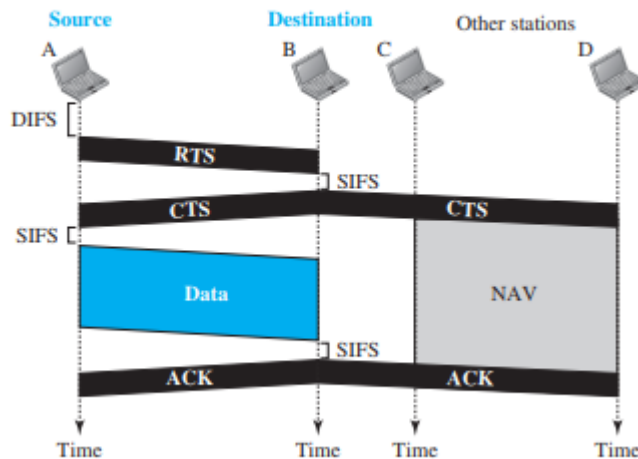
IEEE 802.11 defines two MAC sublayers: the **distributed coordination function (DCF)** and **point coordination function (PCF)**.

Distributed Coordination Function:

- DCF uses CSMA/CA as the access method.

Frame Exchange Time Line

Figure 15.7 CSMA/CA and NAV



1. Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency.
 - a. The channel uses a persistence strategy with backoff until the channel is idle.
 - b. After the station is found to be idle, the station waits for a period of time called the distributed interframe space (DIFS); then the station sends a control frame called the request to send (RTS).
2. After receiving the RTS and waiting a period of time called the short interframe space (SIFS), the destination station sends a control frame, called the clear to send (CTS), to the source station. This control frame indicates that the destination station is ready to receive data.
3. The source station sends data after waiting an amount of time equal to SIFS.
4. The destination station, after waiting an amount of time equal to SIFS, sends an acknowledgment to show that the frame has been received. Acknowledgment is needed in this protocol because the station does not have any means to check for the successful arrival of its data at the destination. On the other hand, the lack of collision in CSMA/CD is a kind of indication to the source that data have arrived.

Network Allocation Vector

When a station sends an RTS frame, it includes the duration of time that it needs to occupy the channel. The stations that are affected by this transmission create a timer called a network allocation vector (NAV) that shows how much time must pass before these stations are allowed to check the channel for idleness. Each time a station accesses the system and sends an

RTS frame, other stations start their NAV. In other words, each station, before sensing the physical medium to see if it is idle, first checks its NAV to see if it has expired.

Collision During Handshaking

What happens if there is a collision during the time when RTS or CTS control frames are in transition, often called the handshaking period? Two or more stations may try to send RTS frames at the same time. These control frames may collide. However, because there is no mechanism for collision detection, the sender assumes there has been a collision if it has not received a CTS frame from the receiver. The backoff strategy is employed, and the sender tries again.

Hidden-Station Problem

The solution to the hidden station problem is the use of the handshake frames (RTS and CTS). Figure 15.7 also shows that the RTS message from B reaches A, but not C. However, because both B and C are within the range of A, the CTS message, which contains the duration of data transmission from B to A, reaches C. Station C knows that some hidden station is using the channel and refrains from transmitting until that duration is over.

Point Coordination Function (PCF)

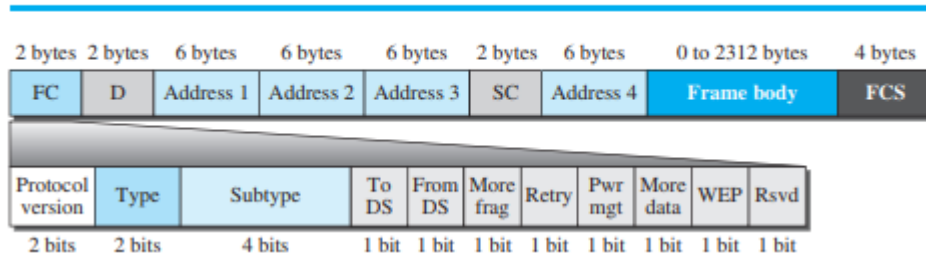
The point coordination function (PCF) is an optional access method that can be implemented in an infrastructure network (not in an ad hoc network). It is implemented on top of the DCF and is used mostly for time-sensitive transmission. PCF has a centralized, contention-free polling access method. The AP performs polling for stations that are capable of being polled. The stations are polled one after another, sending any data they have to the AP. To give priority to PCF over DCF, another interframe space, PIFS, has been defined. PIFS (PCF IFS) is shorter than DIFS. This means that if, at the same time, a station wants to use only DCF and an AP wants to use PCF, the AP has priority. Due to the priority of PCF over DCF, stations that only use DCF may not gain access to the medium. To prevent this, a repetition interval has been designed to cover both contention-free PCF and contention-based DCF traffic. The repetition interval, which is repeated continuously, starts with a special control frame, called a beacon frame. When the stations hear the beacon frame, they start their NAV for the duration of the contention-free period of the repetition interval.

Fragmentation

The wireless environment is very noisy, so frames are often corrupted. A corrupt frame has to be retransmitted. The protocol, therefore, recommends fragmentation—the division of a large frame into smaller ones. It is more efficient to resend a small frame than a large one.

Frame Format

Figure 15.9 *Frame format*



- **Frame control (FC).** The FC field is 2 bytes long and defines the type of frame and some control information.
- **D.** This field defines the duration of the transmission that is used to set the value of NAV. In one control frame, it defines the ID of the frame.
- **Addresses.** There are four address fields, each 6 bytes long. The meaning of each address field depends on the value of the To DS and From DS subfields.
- **Sequence control.** This field, often called the SC field, defines a 16-bit value. The first four bits define the fragment number; the last 12 bits define the sequence number, which is the same in all fragments.
- **Frame body.** This field, which can be between 0 and 2312 bytes, contains information based on the type and the subtype defined in the FC field.
- **FCS.** The FCS field is 4 bytes long and contains a CRC-32 error-detection sequence.

Frame Types

A wireless LAN defined by IEEE 802.11 has three categories of frames: **management frames, control frames, and data frames.**

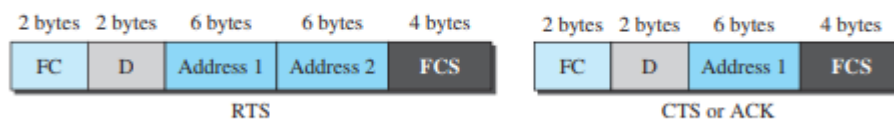
Management Frames

Management frames are used for the initial communication between stations and access points.

Control Frames

Control frames are used for accessing the channel and acknowledging frames.

Figure 15.10 Control frames



Data Frames

Data frames are used for carrying data and control information.

Addressing Mechanism

The IEEE 802.11 addressing mechanism specifies four cases, defined by the value of the two flags in the FC field, To DS and From DS. Each flag can be either 0 or 1, resulting in four different situations. The interpretation of the four addresses (address 1 to address 4) in the MAC frame depends on the value of these flags.

Table 15.3 Addresses

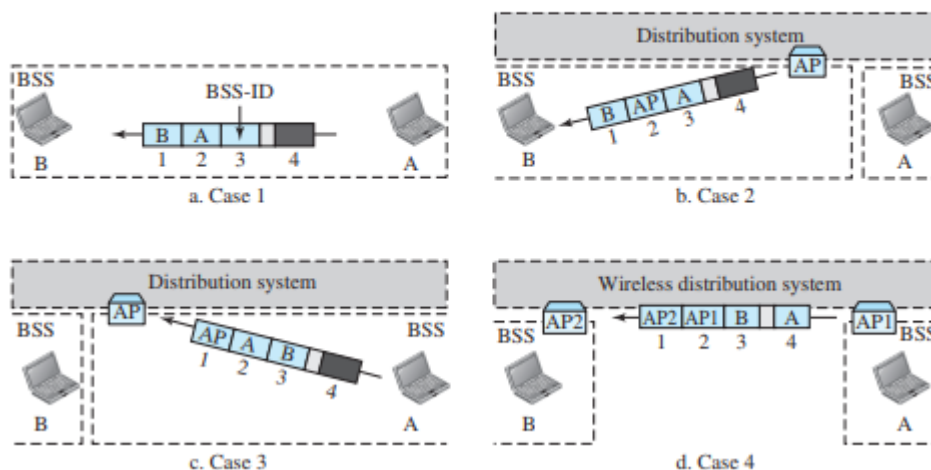
To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	Destination	Source	BSS ID	N/A
0	1	Destination	Sending AP	Source	N/A
1	0	Receiving AP	Source	Destination	N/A
1	1	Receiving AP	Sending AP	Destination	Source

- **Case 1: 00** In this case, To DS = 0 and From DS = 0. This means that the frame is not going to a distribution system (To DS = 0) and is not coming from a distribution system

(From DS = 0). The frame is going from one station in a BSS to another without passing through the distribution system.

- **Case 2: 01** In this case, To DS = 0 and From DS = 1. This means that the frame is coming from a distribution system (From DS = 1). The frame is coming from an AP and going to a station. The addresses are as shown in Figure 15.11. Note that address 3 contains the original sender of the frame (in another BSS).
- **Case 3: 10** In this case, To DS = 1 and From DS = 0. This means that the frame is going to a distribution system (To DS = 1). The frame is going from a station to an AP. The ACK is sent to the original station. The addresses are as shown in Figure 15.11. Note that address 3 contains the final destination of the frame in the distribution system.
- **Case 4: 11** In this case, To DS = 1 and From DS = 1. This is the case in which the distribution system is also wireless. The frame is going from one AP to another AP in a wireless distribution system. Here, we need four addresses to define the original sender, the final destination, and two intermediate APs. Figure 15.11 shows the situation.

Figure 15.11 Addressing mechanisms



BLUETOOTH

Bluetooth is a wireless LAN technology designed to connect devices of different functions such as telephones, notebooks, computers (desktop and laptop), cameras, printers, and even coffee makers when they are at a short distance from each other. A Bluetooth LAN is an ad hoc network, which means that the network is formed spontaneously; the devices, sometimes called gadgets, find each other and make a network called a piconet. A Bluetooth LAN can even be connected to the Internet if one of the gadgets has this capability. A Bluetooth LAN, by nature, cannot be large. If there are many gadgets that try to connect, there is chaos.

Bluetooth technology has several applications. Peripheral devices such as a wireless mouse or keyboard can communicate with the computer through this technology. Monitoring devices can communicate with sensor devices in a small health care center. Home security devices can use this technology to connect different sensors to the main security controller. Conference attendees can synchronize their laptop computers at a conference.

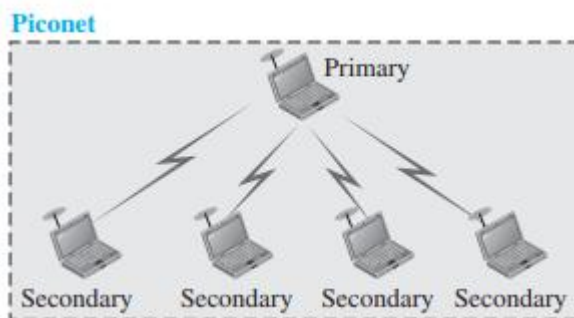
Architecture

Bluetooth defines two types of networks: piconet and scatternet.

Piconets

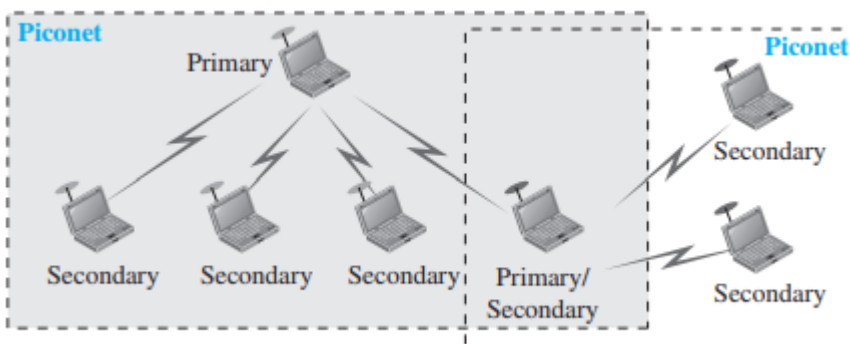
A Bluetooth network is called a piconet, or a small net. A piconet can have up to eight stations, one of which is called the primary; the rest are called secondaries. All the secondary stations synchronize their clocks and hopping sequence with the primary. Note that a piconet can have only one primary station. The communication between the primary and secondary stations can be one-to-one or one-to-many.

Although a piconet can have a maximum of seven secondaries, additional secondaries can be in the parked state. A secondary in a parked state is synchronized with the primary, but cannot take part in communication until it is moved from the parked state to the active state. Because only eight stations can be active in a piconet, activating a station from the parked state means that an active station must go to the parked state.



Scatternet

Piconets can be combined to form what is called a scatternet. A secondary station in one piconet can be the primary in another piconet. This station can receive messages from the primary in the first piconet (as a secondary) and, acting as a primary, deliver them to secondaries in the second piconet. A station can be a member of two piconets.

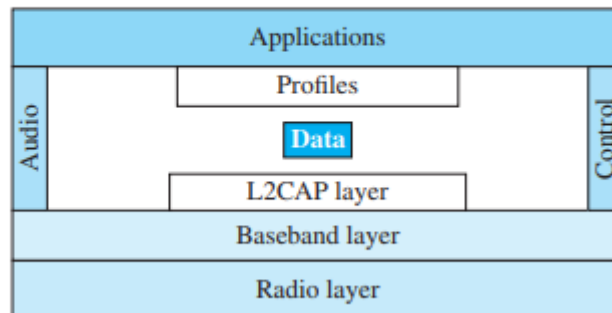


Bluetooth Devices

A Bluetooth device has a built-in short-range radio transmitter. The current data rate is 1 Mbps with a 2.4-GHz bandwidth. This means that there is a possibility of interference between the IEEE 802.11b wireless LANs and Bluetooth LANs.

Bluetooth Layers

Figure 15.19 *Bluetooth layers*



L2CAP

The Logical Link Control and Adaptation Protocol, or L2CAP (L2 here means LL), is roughly equivalent to the LLC sublayer in LANs. It is used for data exchange on an ACL link; SCO channels do not use L2CAP. Figure shows the format of the data packet at this level.

Figure 15.20 *L2CAP data packet format*



The 16-bit length field defines the size of the data, in bytes, coming from the upper layers. Data can be up to 65,535 bytes. The channel ID (CID) defines a unique identifier for the virtual channel created at this level (see below). The L2CAP has specific duties: multiplexing, segmentation and reassembly, quality of service (QoS), and group management.

Baseband Layer

The baseband layer is roughly equivalent to the MAC sublayer in LANs. The access method is TDMA (discussed later). The primary and secondary stations communicate with each other using time slots. The length of a time slot is exactly the same as the dwell time, 625 μ s. This means that during the time that one frequency is used, a primary sends a frame to a secondary, or a secondary sends a frame to the primary. Note that the communication is only between the primary and a secondary; secondaries cannot communicate directly with one another.

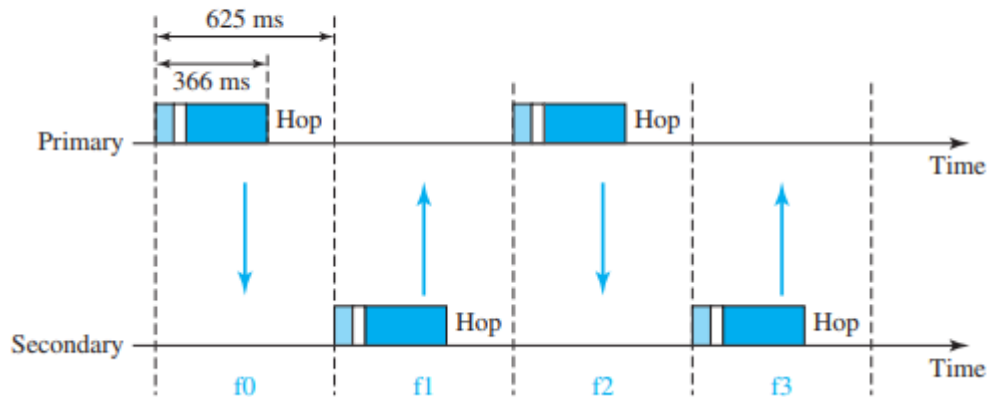
TDMA

Bluetooth uses a form of TDMA that is called TDD-TDMA (time-division duplex TDMA). TDD-TDMA is a kind of half-duplex communication in which the sender and receiver send and receive data, but not at the same time (half-duplex); however, the communication for each direction uses different hops. This is similar to walkie-talkies using different carrier frequencies.

- **Single-Secondary Communication** If the piconet has only one secondary, the TDMA operation is very simple. The time is divided into slots of 625 μ s. The primary uses even-numbered slots (0, 2, 4, . . .); the secondary uses odd-numbered slots (1, 3, 5, . . .). TDD-TDMA allows the primary and the secondary to communicate in half-duplex mode. In

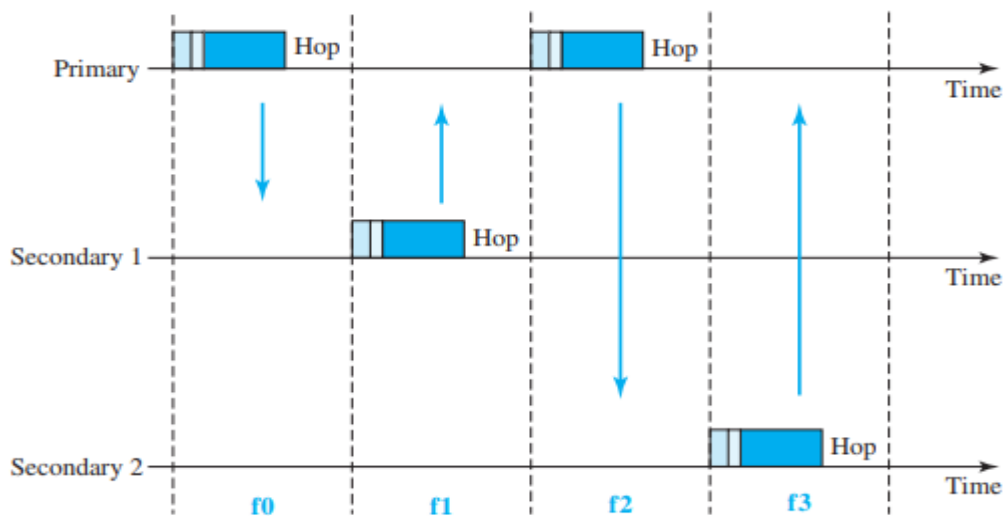
slot 0, the primary sends and the secondary receives; in slot 1, the secondary sends and the primary receives. The cycle is repeated. Figure 15.21 shows the concept.

Figure 15.21 *Single-secondary communication*



- **Multiple-Secondary Communication** The process is a little more involved if there is more than one secondary in the piconet. Again, the primary uses the even-numbered slots, but a secondary sends in the next odd-numbered slot if the packet in the previous slot was addressed to it. All secondaries listen on even-numbered slots, but only one secondary sends in any odd-numbered slot.

Figure 15.22 *Multiple-secondary communication*



Let us elaborate on the figure.

1. In slot 0, the primary sends a frame to secondary 1.
2. In slot 1, only secondary 1 sends a frame to the primary because the previous frame was addressed to secondary 1; other secondaries are silent.
3. In slot 2, the primary sends a frame to secondary 2.
4. In slot 3, only secondary 2 sends a frame to the primary because the previous frame was addressed to secondary 2; other secondaries are silent.
5. The cycle continues.

We can say that this access method is similar to a poll/select operation with reservations. When the primary selects a secondary, it also polls it. The next time slot is reserved for the polled station to send its frame. If the polled secondary has no frame to send, the channel is silent.

Links

Two types of links can be created between a primary and a secondary: SCO links and ACL links.

- **SCO A synchronous connection-oriented (SCO)** link is used when avoiding latency (delay in data delivery) is more important than integrity (error-free delivery). In an SCO link, a physical link is created between the primary and a secondary by reserving specific slots at regular intervals. The basic unit of connection is two slots, one for each direction. If a packet is damaged, it is never retransmitted. SCO is used for real-time audio where avoiding delay is all-important. A secondary can create up to three SCO links with the primary, sending digitized audio (PCM) at 64 kbps in each link.
- **ACL An asynchronous connectionless link (ACL)** is used when data integrity is more important than avoiding latency. In this type of link, if a payload encapsulated in the frame is corrupted, it is retransmitted. A secondary returns an ACL frame in the available odd-numbered slot if the previous slot has been addressed to it. ACL can use one, three, or more slots and can achieve a maximum data rate of 721 kbps.

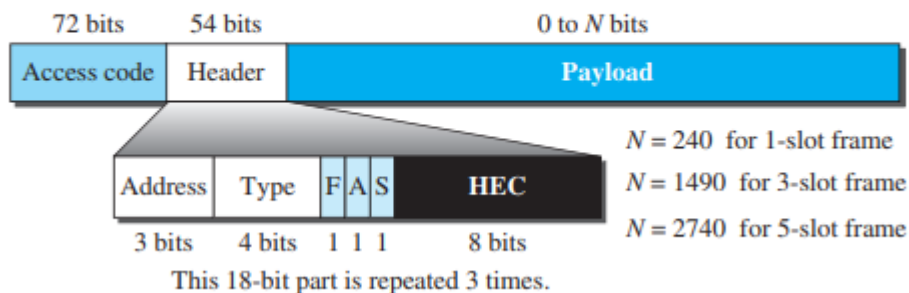
Frame Format

A frame in the baseband layer can be one of three types: one-slot, three-slot, or fiveslot. A slot, as we said before, is 625 μ s. However, in a one-slot frame exchange, 259 μ s is needed for hopping and control mechanisms. This means that a one-slot frame can last only 625 – 259, or 366 μ s. With a 1-MHz bandwidth and 1 bit/Hz, the size of a oneslot frame is 366 bits.

A three-slot frame occupies three slots. However, since 259 μ s is used for hopping, the length of the frame is $3 \times 625 - 259 = 1616 \mu$ s or 1616 bits. A device that uses a three-slot frame remains at the same hop (at the same carrier frequency) for three slots. Even though only one hop number is used, three hop numbers are consumed. That means the hop number for each frame is equal to the first slot of the frame.

A five-slot frame also uses 259 bits for hopping, which means that the length of the frame is $5 \times 625 - 259 = 2866$ bits.

Figure 15.23 Frame format types



The following describes each field:

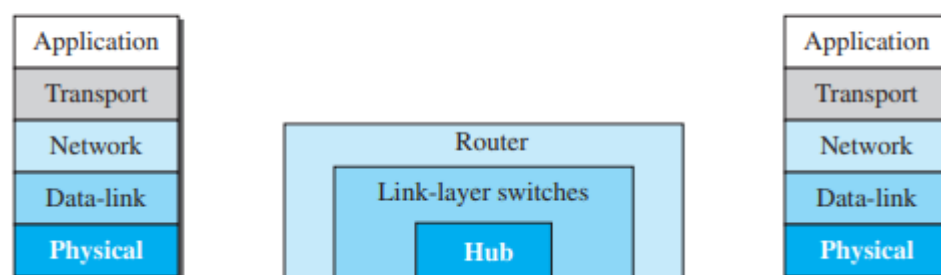
- **Access code.** This 72-bit field normally contains synchronization bits and the identifier of the primary to distinguish the frame of one piconet from that of another.
- **Header.** This 54-bit field is a repeated 18-bit pattern. Each pattern has the following subfields:
 - a. **Address.** The 3-bit address subfield can define up to seven secondaries (1 to 7). If the address is zero, it is used for broadcast communication from the primary to all secondaries.
 - b. **Type.** The 4-bit type subfield defines the type of data coming from the upper layers. We discuss these types later.
 - c. **F.** This 1-bit subfield is for flow control. When set (1), it indicates that the device is unable to receive more frames (buffer is full).
 - d. **A.** This 1-bit subfield is for acknowledgment. Bluetooth uses Stop-and-Wait ARQ; 1 bit is sufficient for acknowledgment.
 - e. **S.** This 1-bit subfield holds a sequence number. Bluetooth uses Stop-and-Wait ARQ; 1 bit is sufficient for sequence numbering.
 - f. **HEC.** The 8-bit header error correction subfield is a checksum to detect errors in each 18-bit header section. The header has three identical 18-bit sections. The receiver compares these three sections, bit by bit. If each of the corresponding bits is the same, the bit is accepted; if not, the majority opinion rules. This is a form of forward error correction (for the header only). This double error control is needed because the nature of the communication, via air, is very noisy. Note that there is no retransmission in this sublayer.
- **Payload.** This subfield can be 0 to 2740 bits long. It contains data or control information coming from the upper layers.

CONNECTING DEVICES

Hosts and networks do not normally operate in isolation. We use connecting devices to connect hosts together to make a network or to connect networks together to make an internet. Connecting devices can operate in different layers of the Internet model. We discuss three kinds of connecting devices: **hubs, link-layer switches, and routers.**

Hubs today operate in the first layer of the Internet model. Link-layer switches operate in the first two layers. Routers operate in the first three layers.

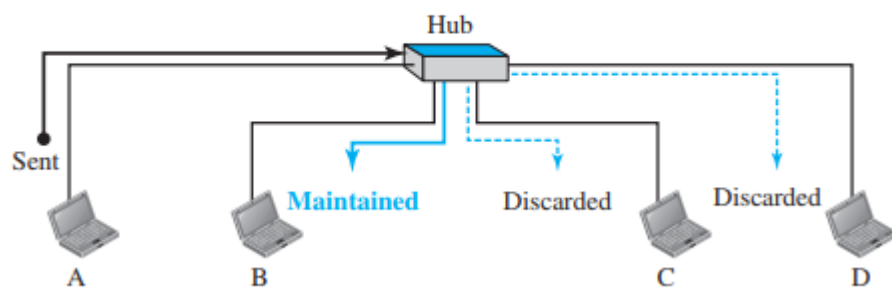
Figure 17.1 *Three categories of connecting devices*



Hubs

A hub is a device that operates only in the physical layer. Signals that carry information within a network can travel a fixed distance before attenuation endangers the integrity of the data. A repeater receives a signal and, before it becomes too weak or corrupted, regenerates and retimes the original bit pattern. The repeater then sends the refreshed signal. In the past, when Ethernet LANs were using bus topology, a repeater was used to connect two segments of a LAN to overcome the length restriction of the coaxial cable. Today, however, Ethernet LANs use star topology. In a star topology, a repeater is a multiport device, often called a hub, that can be used to serve as the connecting point and at the same time function as a repeater.

Figure 17.2 A hub

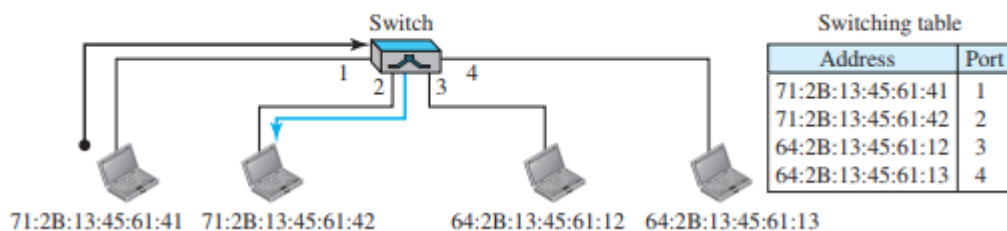


when a packet from station A to station B arrives at the hub, the signal representing the frame is regenerated to remove any possible corrupting noise, but the hub forwards the packet from all outgoing ports except the one from which the signal was received. In other words, the frame is broadcast. All stations in the LAN receive the frame, but only station B keeps it. The rest of the stations discard it.

Link-Layer Switches

A link-layer switch (or switch) operates in both the physical and the data-link layers. As a physical-layer device, it regenerates the signal it receives. As a link-layer device, the link-layer switch can check the MAC addresses (source and destination) contained in the frame.

Figure 17.3 Link-layer switch



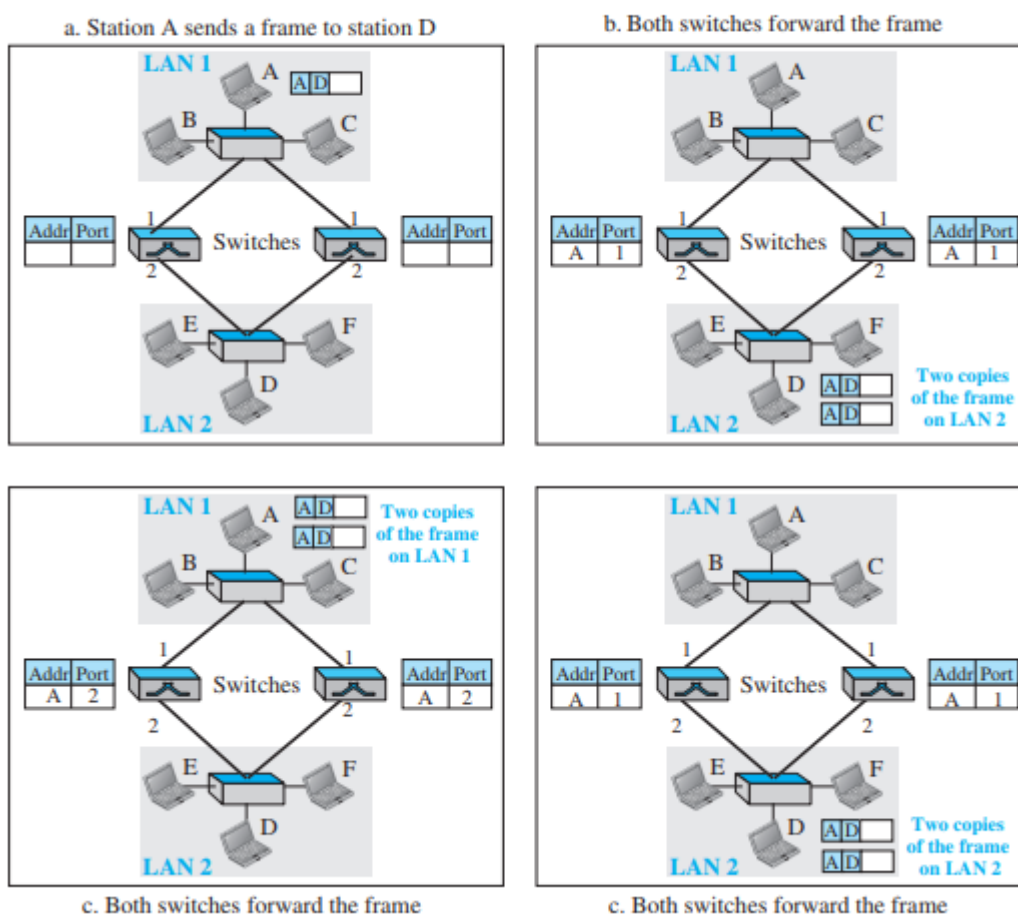
we have a LAN with four stations that are connected to a link-layer switch. If a frame destined for station 71:2B:13:45:61:42 arrives at port 1, the link-layer switch consults its table to

find the departing port. According to its table, frames for 71:2B:13:45:61:42 should be sent out only through port 2; therefore, there is no need for forwarding the frame through other ports.

Loop Problem

Transparent switches work fine as long as there are no redundant switches in the system. Systems administrators, however, like to have redundant switches (more than one switch between a pair of LANs) to make the system more reliable. If a switch fails, another switch takes over until the failed one is repaired or replaced. Redundancy can create loops in the system, which is very undesirable. Loops can be created only when two or more broadcasting LANs (those using hubs, for example) are connected by more than one switch.

Figure 17.5 Loop problem in a learning switch



Spanning Tree Algorithm

To solve the looping problem, the IEEE specification requires that switches use the spanning tree algorithm to create a loopless topology. In graph theory, a spanning tree is a graph in which there is no loop.

The process for finding the spanning tree involves three steps:

1. Every switch has a built-in ID (normally the serial number, which is unique). Each switch broadcasts this ID so that all switches know which one has the smallest ID. The switch

with the smallest ID is selected as the root switch (root of the tree). We assume that switch S1 has the smallest ID. It is, therefore, selected as the root switch.

2. The algorithm tries to find the shortest path (a path with the shortest cost) from the root switch to every other switch or LAN. The shortest path can be found by examining the total cost from the root switch to the destination. Figure 17.7 shows the shortest paths. We have used the Dijkstra algorithm described in Chapter 20.
3. The combination of the shortest paths creates the shortest tree, which is also shown in Figure 17.7.
4. Based on the spanning tree, we mark the ports that are part of it, the forwarding ports, which forward a frame that the switch receives. We also mark those ports that are not part of the spanning tree, the blocking ports, which block the frames received by the switch. Figure 17.8 shows the logical systems of LANs with forwarding ports (solid lines) and blocking ports (broken lines).

Routers

A router is a three-layer device; it operates in the physical, data-link, and network layers. As a physical-layer device, it regenerates the signal it receives. As a link-layer device, the router checks the physical addresses (source and destination) contained in the packet. As a network-layer device, a router checks the network-layer addresses.

A router can connect networks. In other words, a router is an internetworking device; it connects independent networks to form an internetwork. According to this definition, two networks connected by a router become an internetwork or an internet. There are three major differences between a router and a repeater or a switch.

1. A router has a physical and logical (IP) address for each of its interfaces.
2. A router acts only on those packets in which the link-layer destination address matches the address of the interface at which the packet arrives.
3. A router changes the link-layer address of the packet (both source and destination) when it forwards the packet.

Figure 17.9 *Routing example*

