

3.5.5 Compounding High- and Low-Resolution Packets

- Another solution is to create a duplicate of each packet with a low-resolution redundancy and combine the redundant version with the next packet.
- For example (Figure 10.22):

We can create 4 low-resolution packets out of 5 high-resolution packets and send them (Fig 10.22).

- If a packet is lost, we can use the low-resolution version from the next packet.
- In this method, if the last packet is lost, it cannot be recovered, but we use the low-resolution version of a packet if the lost packet is not the last one.
- The audio and video reproduction does not have the same quality, but the lack of quality is not recognized most of the time.

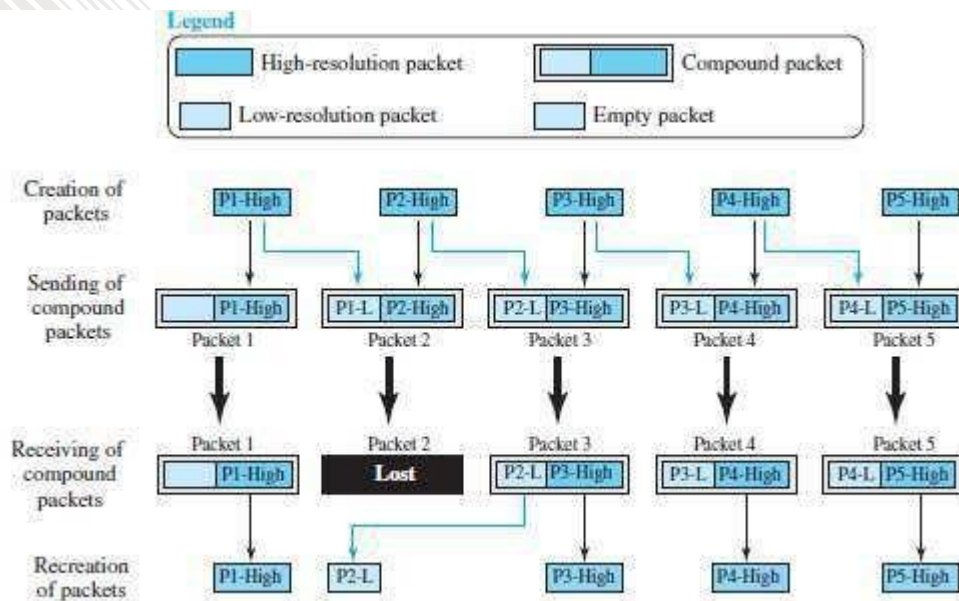


Figure 10.22 Compounding high- and low-resolution packets

DATA LINK CONTROL

3.6 DLC SERVICES

- The data link control (DLC) deals with procedures for communication between two adjacent nodes i.e. node-to-node communication.
- Data link control functions include 1) Framing and 2) Flow control and 3) Error control.

3.6.1 Framing

- A frame is a group of bits.
- Framing means organizing the bits into a frame that are carried by the physical layer.
- The data-link-layer needs to form frames, so that each frame is distinguishable from another.
- Framing separates a message from other messages by adding sender-address & destination-address.
- The destination-address defines where the packet is to go.

The sender-address helps the recipient acknowledge the receipt.

- Q: Why the whole message is not packed in one frame?

Ans: Large frame makes flow and error-control very inefficient.

Even a single-bit error requires the re-transmission of the whole message.

When a message is divided into smaller frames, a single-bit error affects only that small frame. (Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter. In addition, each envelope defines the sender and receiver addresses since the postal system is a many-to-many carrier facility).

3.6.1.1 Frame Size • Two types

of frames: 1) **Fixed Size**

Framing

- There is no need for defining boundaries of frames; the size itself can be used as a delimiter. □ For example: ATM WAN uses frames of fixed size called cells.

2) **Variable Size Framing**

- We need to define the end of the frame and the beginning of the next frame. □ Two approaches are used: 1) Character-oriented approach 2) Bit-oriented approach.

3.6.1.2 Character Oriented Framing

- Data to be carried are 8-bit characters from a coding system such as ASCII (Figure 11.1).
- The header and the trailer are also multiples of 8 bits.
 - 1) Header carries the source and destination-addresses and other control information. 2) Trailer carries error-detection or error-correction redundant bits.
- To separate one frame from the next frame, an 8-bit (1-byte) flag is added at the beginning and the end of a frame.
- The flag is composed of protocol-dependent special characters.
- The flag signals the start or end of a frame.

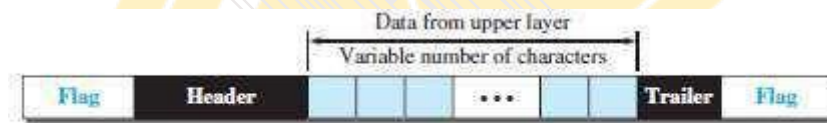


Figure 11.1 A frame in a character-oriented protocol

• Problem:

- Character-oriented framing is suitable when only text is exchanged by the data-link-layers. □ However, if we send other type of information (say audio/video), then any pattern used for the flag can also be part of the information.
- If the flag-pattern appears in the data-section, the receiver might think that it has reached the end of the frame.

Solution: A byte-stuffing is used.

(Byte stuffing □ character stuffing)

- In byte 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 called the escape character (ESC), which has a predefined bit pattern.
- When a receiver encounters the ESC character, the receiver → removes ESC character from the data-section and → treats the next character as data, not a delimiting flag.

• Problem:

- What happens if the text contains one or more escape characters followed by a flag?

- ❑ The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame. Solution:
- ❑ Escape characters part of the text must also be marked by another escape character (Fig 11.2).

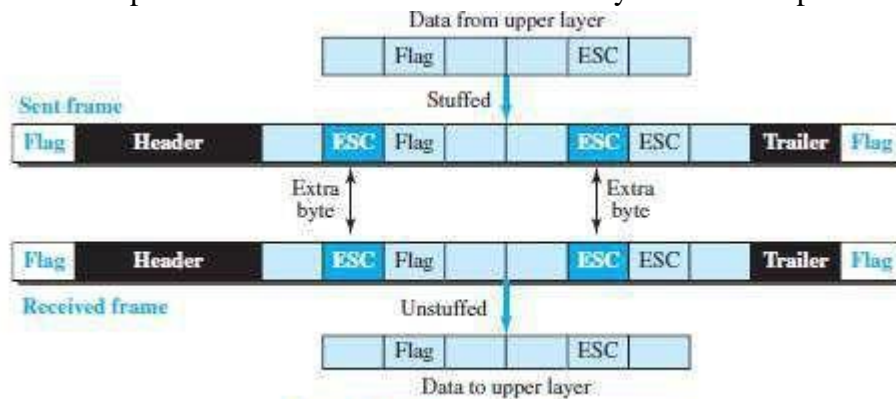


Figure 11.2 Byte stuffing and unstuffing

- In short, byte stuffing is the process of adding one extra byte whenever there is a flag or escape character in the text.

3.6.1.3 Bit Oriented Framing

- The data-section of a frame is a sequence of bits to be interpreted by the upper layer as text, audio, video, and so on.
- However, in addition to headers and trailers, we 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).

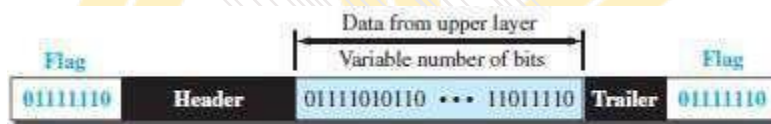


Figure 11.3 A frame in a bit-oriented protocol

- Problem:
 - ❑ If the flag-pattern appears in the data-section, the receiver might think that it has reached the end of the frame.

Solution: A bit-stuffing is used.

- 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. (Figure 11.4).
- This guarantees that the flag field sequence does not inadvertently appear in the frame.

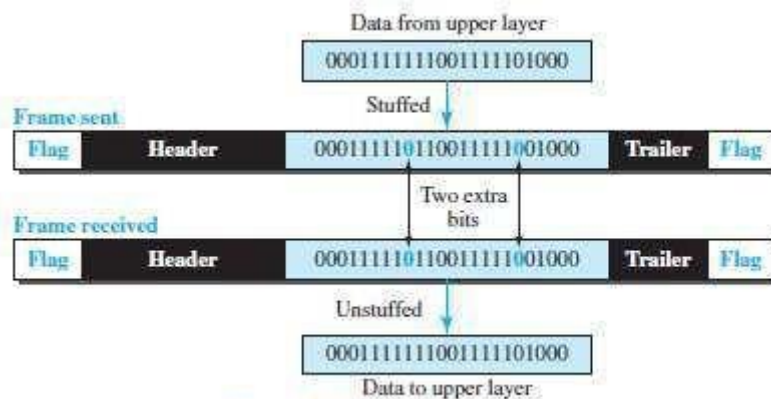


Figure 11.4 Bit stuffing and unstuffing

- In short, bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

3.6.2 Flow Control and Error Control

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

3.6.2.1 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.
- We need to prevent losing the data items at the consumer site.

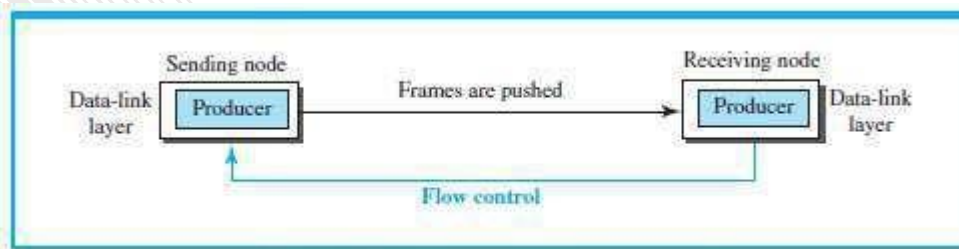


Figure 11.5 Flow control at the data-link layer

- At the sending node, the data-link layer tries to push frames toward the data-link layer at the receiving node (Figure 11.5).
- 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.
- Here, flow control can be feedback from the receiving node to the sending node to stop or slow down pushing frames.

3.6.2.1.1 Buffers

- Flow control can be implemented by using buffer.
- A buffer is a set of memory locations that can hold packets at the sender and receiver.
- Normally, two buffers can be used. 1) First buffer at the sender.
2) Second buffer at the receiver.
- The flow control communication can occur by sending signals from the consumer to the producer.
- When the buffer of the receiver is full, it informs the sender to stop pushing frames.

3.6.2.2 Error Control

- Error-control includes both error-detection and error-correction.
- Error-control allows the receiver to inform the sender of any frames lost/damaged in transmission.
- A CRC is
 - added to the frame header by the sender and → checked by the receiver.
- At the data-link layer, error control is normally implemented using one of the following two methods.
 - 1) First method: If the frame is corrupted, it is discarded;
If the frame is not corrupted, the packet is delivered to the network layer.
This method is used mostly in wired LANs such as Ethernet.

2) Second method: If the frame is corrupted, it is discarded;

If the frame is not corrupted, an acknowledgment is sent to the sender.

Acknowledgment is used for the purpose of both flow and error control.

3.6.2.2.1 Combination of Flow and Error Control

- Flow and error control can be combined.
- 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.
- A frame that carries an acknowledgment is normally called an ACK to distinguish it from the dataframe.

3.6.3 Connectionless and Connection-Oriented

- A DLC protocol can be either connectionless or connection-oriented.

1) Connectionless Protocol

- ☐ Frames are sent from one node to the next without any relationship between the frames; each frame is independent.
- ☐ The term connectionless does not mean that there is no physical connection (transmission medium) between the nodes; it means that there is no connection between frames.
- ☐ The frames are not numbered and there is no sense of ordering.
- ☐ Most of the data-link protocols for LANs are connectionless protocols.

2) Connection Oriented Protocol

- ☐ A logical connection should first be established between the two nodes (setup phase).
- ☐ After all frames that are somehow related to each other are transmitted (transfer phase), the logical connection is terminated (teardown phase).
- ☐ The frames are numbered and sent in order.
- ☐ If the frames are not received in order, the receiver needs to wait until all frames belonging to the same set are received and then deliver them in order to the network layer.
- ☐ Connection oriented protocols are rare in wired LANs, but we can see them in some point-to-point protocols, some wireless LANs, and some WANs.

3.7 DATA LINK LAYER PROTOCOLS

- Traditionally 2 protocols have been defined for the data-link layer to deal with flow and error control: 1) Simple Protocol and 2) Stop-and-Wait Protocol.
- The behavior of a data-link-layer protocol can be better shown as a finite state machine (FSM).
- An FSM is a machine with a finite number of states (Figure 11.6).
- The machine is always in one of the states until an event occurs.
- Each event is associated with 2 reactions:
 - 1) Defining the list (possibly empty) of actions to be performed.
 - 2) 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.

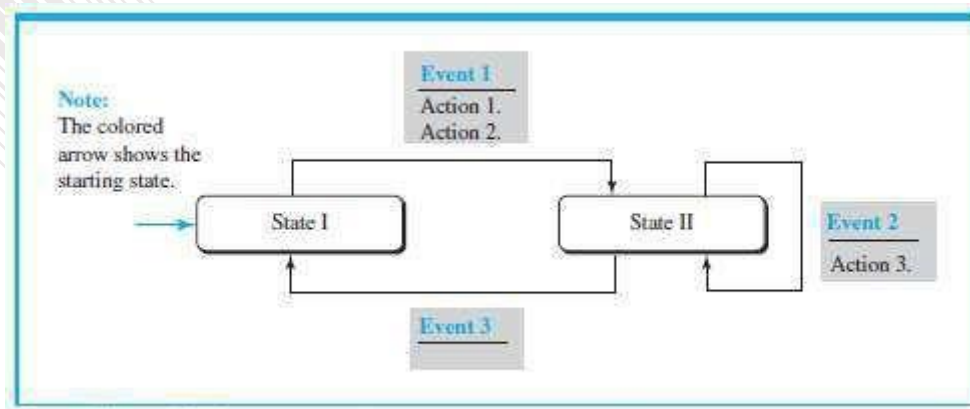


Figure 11.6 Connectionless and connection-oriented service represented as FSMs

3.7.1 Simplest Protocol

- Assumptions:
 - ☐ The protocol has no flow-control or error-control.
 - ☐ The protocol is a unidirectional protocol (in which frames are traveling in only one direction). ☐ The receiver can immediately handle any frame it receives.

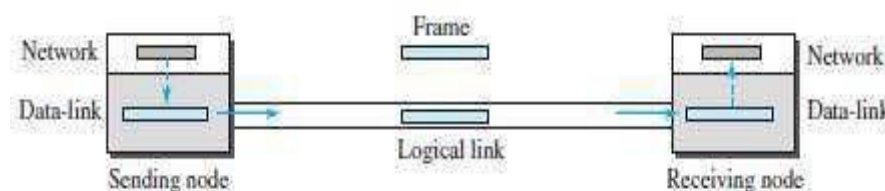


Figure 11.7 Simple protocol

3.7.1.1 Design

- Here is how it works (Figure 11.7):

1) At Sender

- ☒ The data-link-layer
 - gets data from its network-layer →
 - makes a frame out of the data and →
 - sends the frame.

2) At Receiver

✧ The data-link-layer

→ receives a frame from its physical layer →
extracts data from the frame and
→ delivers the data to its network-layer.

□ Data-link-layers of sender & receiver provide transmission services for their network-layers. □
Data-link-layers use the services provided by their physical layers for the physical transmission of bits.

3.7.1.2 FSMs

- Two main requirements:
 - 1) The sender-site cannot send a frame until its network-layer has a data packet to send.
 - 2) The receiver-site cannot deliver a data packet to its network-layer until a frame arrives.
- These 2 requirements are shown using two FSMs.
- Each FSM has only one state, the ready state.

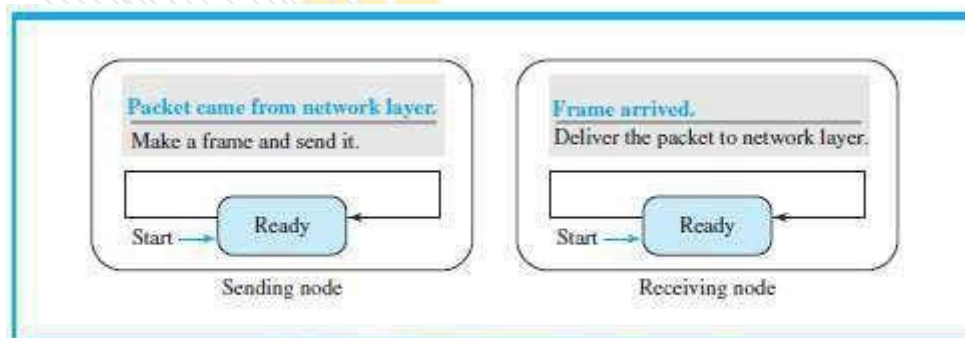


Figure 11.8 FSMs for the simple protocol

- Here is how it works (Figure 11.8):

1) At Sending Machine

- ✧ 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.

2) At 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.

Example 3.6

Figure 11.9 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.

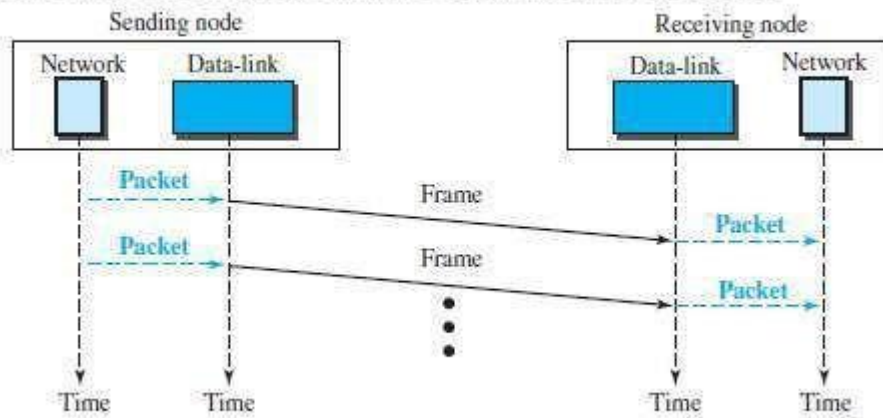


Figure 11.9 Flow diagram

3.7.2 Stop & Wait Protocol

- This uses both flow and error control.
- Normally, the receiver has limited storage-space.
- If the receiver is receiving data from many sources, the receiver may
 - be overloaded with frames &
 - discard the frames.
- To prevent the receiver from being overloaded with frames, we need to tell the sender to slow down.

3.7.2.1 Design

1) At Sender

☐ The sender

- sends one frame & starts a timer
- keeps a copy of the sent-frame and
- waits for ACK-frame from the receiver (okay to go ahead).

☐ Then,

- 1) If an ACK-frame arrives before the timer expires, the timer is stopped and the sender sends the next frame.
Also, the sender discards the copy of the previous frame.
- 2) If the timer expires before ACK-frame arrives, the sender resends the previous frame and restarts the timer

2) At Receiver

- ☐ To detect corrupted frames, a CRC is added to each data frame. ☐ When a frame arrives at the receiver-site, the frame is checked.
- ☐ If frame's CRC is incorrect, the frame is corrupted and discarded.
- ☐ The silence of the receiver is a signal for the sender that a frame was either corrupted or lost.

3.7.2.2 FSMs

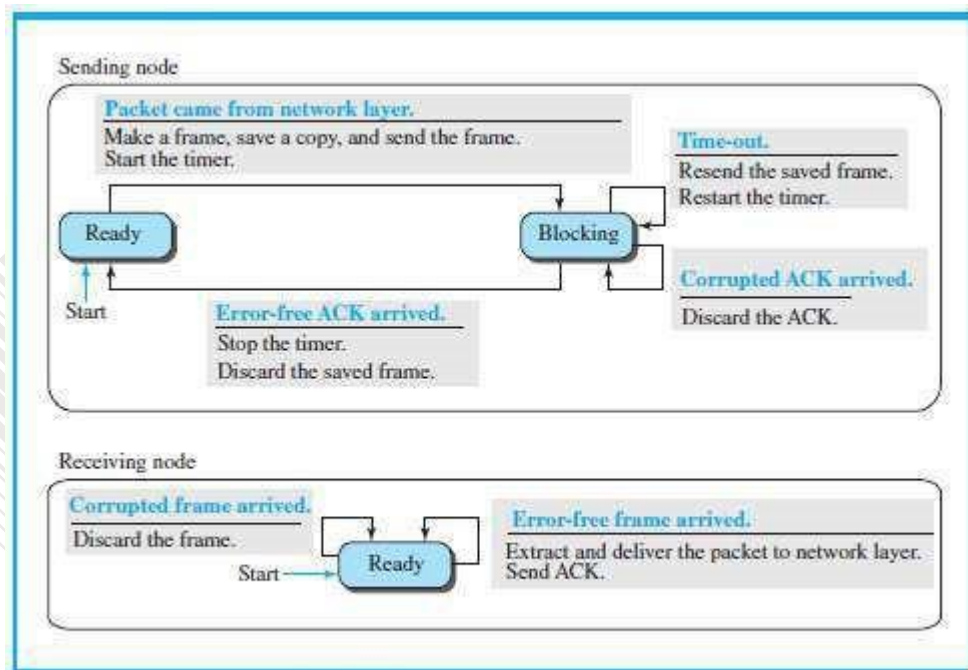


Figure 11.11 FSM for the Stop-and-Wait protocol

- Here is how it works (Figure 11.11):

1) Sender States

- Sender is initially in the ready state, but it can move between the ready and blocking state.

i) 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.

ii) 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.

2) 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.

Example 3.7

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.

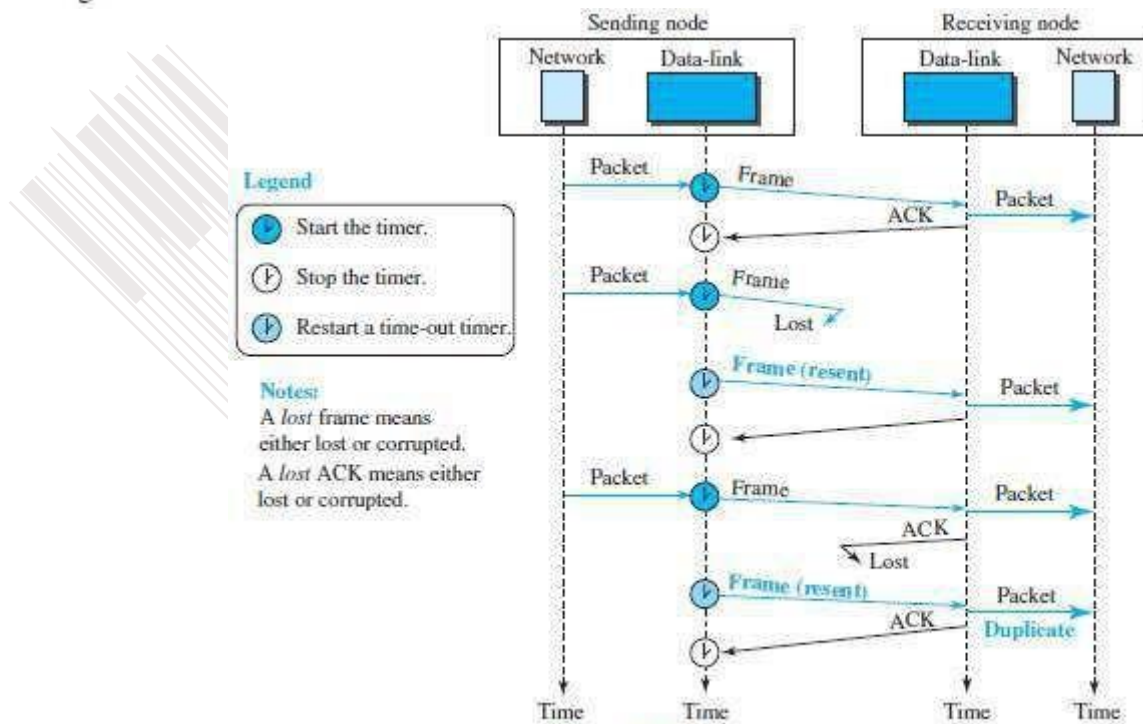


Figure 11.12 Flow diagram

3.7.2.3 Sequence and Acknowledgment Numbers • Q:

How to deal with corrupted-frame?

Ans: If the corrupted-frame arrives at the receiver-site, then the frame is simply discarded.

• Q: How to deal with lost-frames?

Ans: If the receiver receives out-of-order data-frame, then it means that frames were lost. ∴ The lost-frames need to be resent.

• Problem in Stop and Wait protocols:

- 1) There is no way to identify a frame.
- 2) The received-frame could be the correct one, or a duplicate, or a frame out of order.

Solution: 1) Use sequence-number for each data frame.

2) Use Acknowledgment-number for each ACK frame.

Sequence Numbers

□ Frames need to be numbered. This is done by using sequence-numbers. □

A sequence-number field is added to the data-frame.

Acknowledgment Numbers

▣ An acknowledgment-number field is added to the ACK-frame. □ Sequence numbers are 0, 1, 0, 1, 0, 1, ...

The acknowledgment numbers can also be 1, 0, 1, 0, 1, 0, ...

- ❑ The acknowledgment-numbers always announce the sequence-number of the next frame expected by the receiver.
- ❑ For example,
If frame-0 has arrived safely, the receiver sends an ACK-frame with acknowledgment-1 (meaning frame-1 is expected next).

Example 3.8

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.

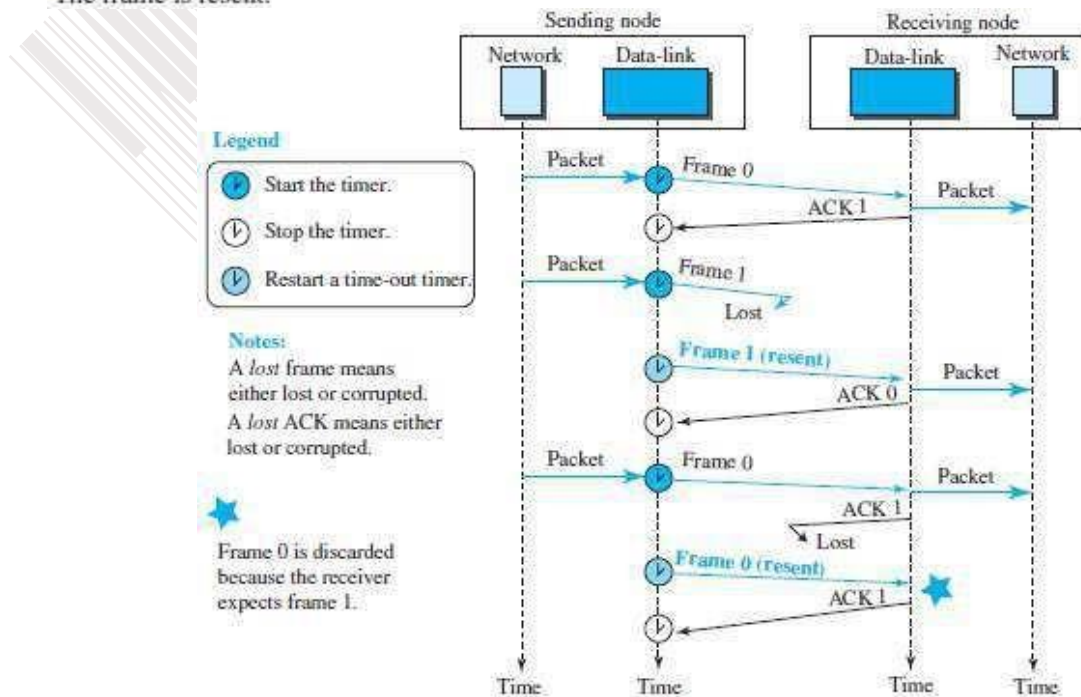


Figure 11.13 Flow diagram

3.7.3 Piggybacking

- A technique called piggybacking is used to improve the efficiency of the bidirectional protocols.
- 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.

3.8 High-Level Data Link Control (HDLC)

- HDLC is a bit-oriented protocol for communication over point-to-point and multipoint links.
- HDLC implements the ARQ mechanisms.

3.8.1 Configurations and Transfer Modes

- HDLC provides 2 common transfer modes that can be used in different configurations: 1) Normal response mode (NRM)
 - 2) Asynchronous balanced mode (ABM).
- The station configuration is unbalanced (Figure 11.14).
- 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 multiple-point links.

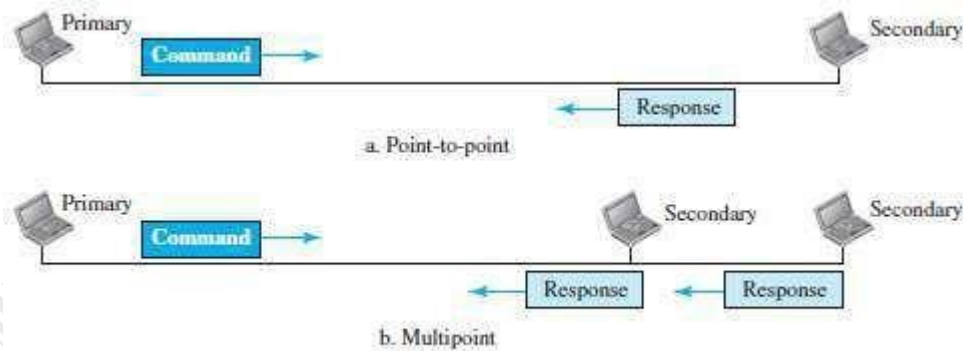


Figure 11.14 Normal response mode

ABM

- The configuration is balanced (Figure 11.15).
- Link is point-to-point, and each station can function as a primary and a secondary (acting as peers).
- This is the common mode today.



Figure 11.15 Asynchronous balanced mode

3.8.2 Framing

- To provide the flexibility necessary to support all the options possible in the modes and configurations, HDLC defines three types of frames:
 - 1) Information frames (I-frames): are used to transport user data and control information relating to user data (piggybacking).
 - 2) Supervisory frames (S-frames): are used only to transport control information.
 - 3) Unnumbered frames (U-frames): are reserved for system management.

Information carried by U-frames is intended for managing the link itself.

- Each type of frame serves as an envelope for the transmission of a different type of message.

3.8.2.1 Frame Format

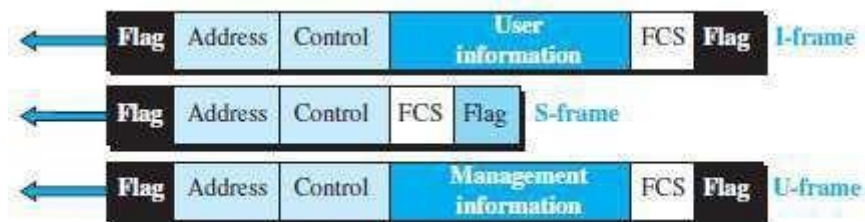


Figure 11.16 HDLC frames

- Various fields of HDLC frame are:

1) Flag Field

- ☐ This field has a synchronization pattern 01111110.
- ☐ This field identifies both the beginning and the end of a frame.

2) 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 creates the frame, it contains a from-address.

☐ This field can be 1 byte or several bytes long, depending on the needs of the network.

3) Control Field

☐ This field is one or two bytes used for flow and error control.

4) Information Field

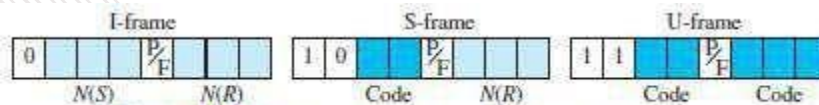
☐ This field contains the user's data from the network-layer or management information. ☐ Its length can vary from one network to another.

5) FCS Field

☐ This field is the error-detection field. (FCS ☐ Frame Check Sequence) ☐ This field can contain either a 2- or 4-byte standard CRC.

3.8.2.1.1 Control Fields of HDLC Frames

• The control field determines the type of frame and defines its functionality (Figure 11.17).



1) 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:

1) The first bit defines the type.

If the first bit of the control field is 0, this means the frame is an I-frame.

2) The next 3 bits N(S) define the sequence-number of the frame.

With 3 bits, we can define a sequence-number between 0 and 7

3) The last 3 bits N(R) correspond to the acknowledgment-number when piggybacking is used.

4) 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 can mean poll or final.

i) 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).

ii) It means final when the frame is sent by a secondary to a primary (when the address field contains the address of the sender).

2) Control Field for S-Frames

- Supervisory frames are used for flow and error-control whenever piggybacking is either impossible or inappropriate (e.g., when the station either has no data of its own to send or needs to send a command or response other than an acknowledgment).
- S-frames do not have information fields.
- The subfields in the control field are:

1) If the first 2 bits of the control field is 10, this means the frame is an S-frame.

2) The last 3 bits N(R) corresponds to the acknowledgment-number (ACK) or negative acknowledgment-number (NAK).

3) The 2 bits called code is used to define the type of S-frame itself.

With 2 bits, we can have four types of S-frames:

1) Receive Ready (RR) = 00

✧ This acknowledges the receipt of frame or group of frames. ✧

The value of N(R) is the acknowledgment-number. 2) Receive

Not Ready (RNR) = 10

✧ This is an RR frame with 1 additional function:

i) It announces that the receiver is busy and cannot receive more frames. ✧ It acts as congestion control mechanism by asking the sender to slow down. ✧ The value of N(R) is the acknowledgment-number.

3) ReJect (REJ) = 01

✧ It is a NAK frame used in Go-Back-N ARQ to improve the efficiency of the process. ✧ It informs the sender, before the sender time expires, that the last frame is lost or damaged.

✧ The value of N(R) is the negative acknowledgment-number.

4) Selective REJect (SREJ) = 11

✧ This is a NAK frame used in Selective Repeat ARQ.

✧ The value of N(R) is the negative acknowledgment-number.

3) Control Field for U-Frames

- Unnumbered frames are used to exchange session management and control information between connected devices.
- U-frames contain an information field used for system management information, but not user data.
- Much of the information carried by U-frames is contained in codes included in the control field.
- U-frame codes are divided into 2 sections:
 - i) A 2-bit prefix before the P/F bit ii)
 - 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.

Example 3.9

Figure 11.18 shows how U-frames can be used for connection establishment and connection release. Node A asks for a connection with a set asynchronous balanced mode (SABM) frame; node B gives a positive response with an unnumbered acknowledgment (UA) frame. After these two exchanges, data can be transferred between the two nodes (not shown in the figure). After data transfer, node A sends a DISC (disconnect) frame to release the connection; it is confirmed by node B responding with a UA (unnumbered acknowledgment).

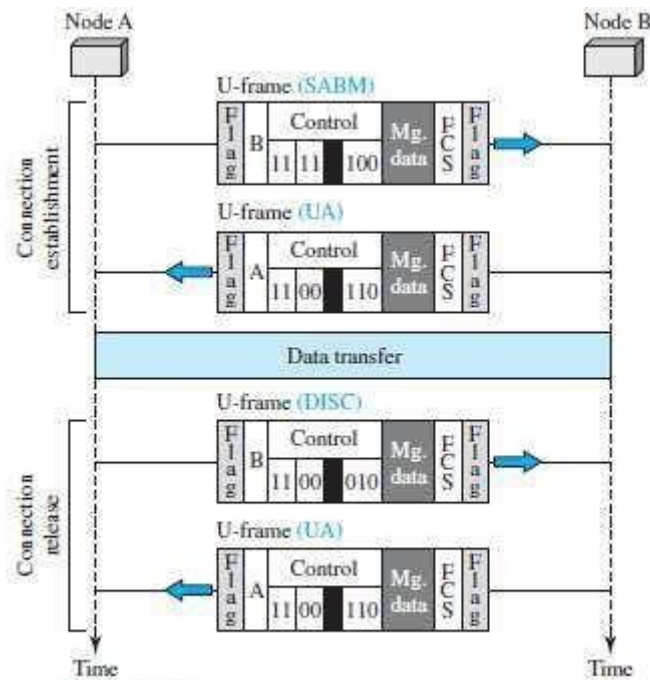


Figure 11.18 Example of connection and disconnection

3.9 POINT-TO-POINT PROTOCOL (PPP)

- PPP is one of the most common protocols for point-to-point access.
- Today, millions of Internet users who connect their home computers to the server of an ISP use PPP.

3.9.1 Framing

- PPP uses a character-oriented (or byte-oriented) frame (Figure 11.20).

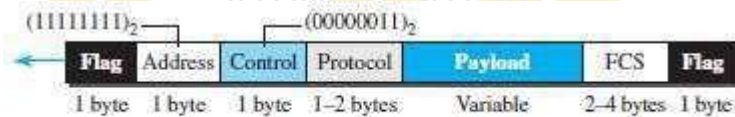


Figure 11.20 PPP frame format

- Various fields of PPP frame are:

1) Flag

- ☐ This field has a synchronization pattern 01111110.
- ☐ This field identifies both the beginning and the end of a frame.

2) Address

- ☐ This field is set to the constant value 11111111 (broadcast address).

3) 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.

4) Protocol

- ☐ This field defines what is being carried in the payload field.
- ☐ Payload field carries either i) user data or ii) other control information.
- ☐ By default, size of this field = 2 bytes.

5) Payload field

- ☐ This field carries either i) user data or ii) other control information.
- ☐ By default, maximum size of this field = 1500 bytes.

- ☐ This field is byte-stuffed if the flag-byte pattern appears in this field. ☐ Padding is needed if the payload-size is less than the maximum size. **6) FCS**
- ☐ This field is the PPP error-detection field.
- ☐ This field can contain either a 2- or 4-byte standard CRC.

3.9.1.1 Byte Stuffing

- Since PPP is a byte-oriented protocol, the flag in PPP is a byte that needs to be escaped whenever it appears in the data section of the frame.
- The escape byte is 01111101, which means that every time the flag like pattern appears in the data, this extra byte is stuffed to tell the receiver that the next byte is not a flag.
- Obviously, the escape byte itself should be stuffed with another escape byte.

3.9.2 Transition Phases

- The transition diagram starts with the dead state (Figure 11.21). **1) Dead State**

- ☐ In dead state, there is no active carrier and the line is quiet.

2) Establish State

- ☐ When 1 of the 2 nodes starts communication, the connection goes into the establish state. ☐ In establish state, options are negotiated between the two parties.

3) Authenticate State

- ☐ If the 2 parties agree that they need authentication,
Then the system needs to do authentication;
Otherwise, the parties can simply start communication.

4) Open State

- ☐ Data transfer takes place in the open state.

5) Terminate State

- ☐ When 1 of the endpoints wants to terminate connection, the system goes to terminate state.

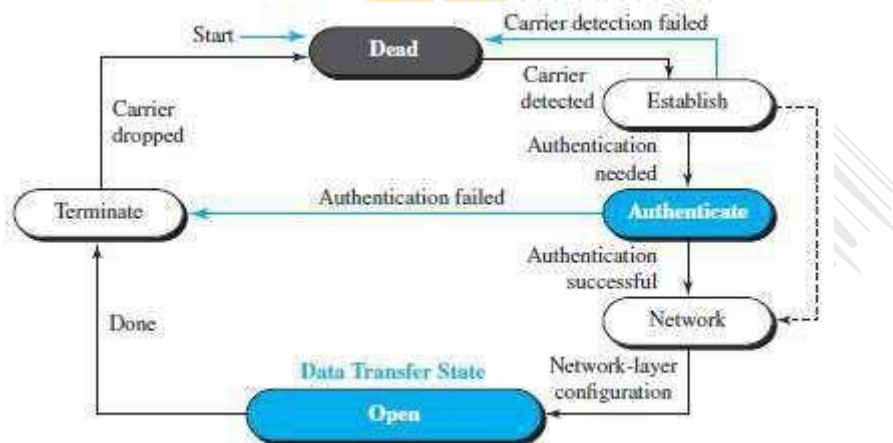


Figure 11.21 Transition phases

MODULE-WISE QUESTIONS

MODULE 3: ERROR-DETECTION AND CORRECTION

- 1) Explain two types of errors (4*)
- 2) Compare error detection vs. error correction (2)
- 3) Explain error detection using block coding technique. (10*)
- 4) Explain hamming distance for error detection (6*)
- 5) Explain parity-check code with block diagram. (6*)
- 6) Explain CRC with block diagram & an example. (10*)
- 7) Write short notes on polynomial codes. (5*)
- 8) Explain internet checksum algorithm along with an example. (6*)
- 9) Explain the following:
 - i) Fletcher checksum and ii) Adler checksum (8)
- 10) Explain various FEC techniques. (6)

DATA LINK CONTROL

- 1) Explain two types of frames. (2)
- 2) Explain character oriented protocol. (6*)
- 3) Explain the concept of byte stuffing and unstuffing with example. (6*)
- 4) Explain bit oriented protocol. (6*)
- 5) Differentiate between character oriented and bit oriented format for Framing. (6*)
- 6) Compare flow control and error control. (4)
- 7) With a neat diagram, explain the design of the simplest protocol with no flow control. (6)
- 8) Write algorithm for sender site and receiver site for the simplest protocol. (6)
- 9) Explain Stop-and-Wait protocol (8*)
- 10) Explain the concept of Piggybacking (2*)
- 11) Explain in detail HDLC frame format. (8*)
- 12) Explain 3 type of frame used in HDLC (8*)
- 13) With a neat schematic, explain the frame structure of PPP protocol. (8*)
- 14) Explain framing and transition phases in Point-to-Point Protocol. (8*)

MODULE 4: MULTIPLE ACCESS

4.1 Introduction

- When nodes use shared-medium, we need multiple-access protocol to coordinate access to medium.
- Analogy:
 - This problem is similar to the rules of speaking in an assembly.
 - We need to ensure
 - Each people has right to speak.
 - Two people do not speak at the same time
 - Two people do not interrupt each other (i.e. Collision Avoidance)
- Many protocols have been designed to handle access to a shared-link (Figure 12.1).
- These protocols belong to a sublayer in the data-link layer called Media Access Control (MAC).
 - 1) Four random-access protocols (or Contention Methods):
 - i) ALOHA
 - ii) CSMA
 - iii) CSMA/CD
 - iv) CSMA/CAThese protocols are mostly used in LANs and WANs.
 - 2) Three controlled-access protocols:
 - i) Reservation
 - ii) Polling
 - iii) Token-passingSome of these protocols are used in LANs.
 - 3) Three channelization protocols:
 - i) FDMA
 - ii) TDMA
 - iii) CDMAThese protocols are used in cellular telephony.

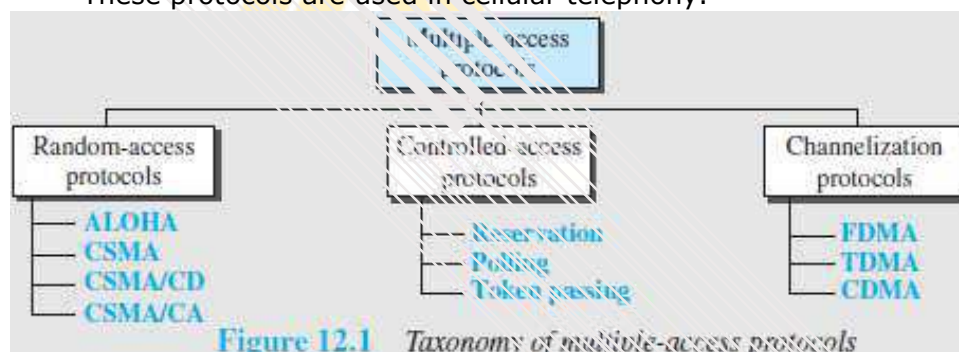


Figure 12.1 Taxonomy of multiple-access protocols

4.2 RANDOM ACCESS PROTOCOL

- No station is superior to another station.
- No station is assigned control over other station.
- To send the data, a station uses a procedure to make a decision on whether or not to send.
- This decision depends on the state of the medium: idle or busy.
- This is called Random Access because
 - Transmission is random among the stations.
 - There is no scheduled-time for a station to transmit.
- This is called Contention Method because
 - Stations compete with one another to access the medium.
- If more than one station tries to send, there is an access-conflict (i.e. collision) and the frames will be destroyed.
- Each station follows a procedure that answers the following questions:
 - 1) When can the station access the medium?
 - 2) What can the station do if the medium is busy?
 - 3) How can the station determine the success or failure of the transmission?
 - 4) What can the station do if there is a collision?
- Four random-access protocols (or Contention methods):
 - 1) ALOHA
 - 2) CSMA (Carrier Sense Multiple Access)
 - 3) CSMA/CD (Carrier Sense Multiple Access with Collision-detection)
 - 4) CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)

DATA COMMUNICATION

4.2.1 ALOHA

- ALOHA was designed for a wireless LAN, but it can be used on any shared medium.
- Since the medium is shared between the stations, there is possibility of collisions.
- When 2 or more stations send the data simultaneously, there is possibility of collision & data loss.

4.2.1.1 Pure ALOHA

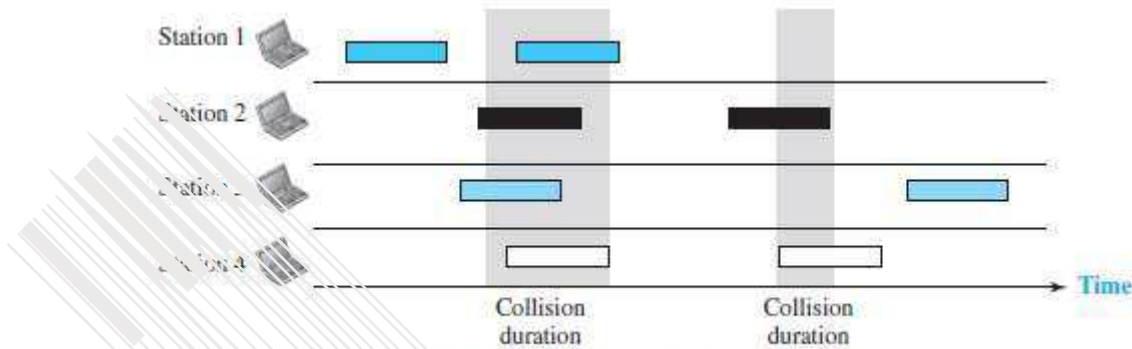


Figure 12.2 Frames in a pure ALOHA network

- Here is how it works (Figure 12.2):
 - 1) The sender sends a frame & starts the timer.
 - 2) The receiver receives the frame and responds with an acknowledgment.
 - 3) If the acknowledgment does not arrive after a time-out period, the sender resends the frame. The sender assumes that the frame (or the acknowledgment) has been destroyed.
 - 4) Since the medium is shared between the stations, there is possibility of collisions.
 - 5) If two stations try to resend the frames after the time-out, the frames will collide again.
 - 6) Two methods to deal with collision:
 - 1) Randomness**
 - ✧ When the time-out period passes, each station waits a random amount of time before resending the frame. This time is called back-off time T_B .
 - ✧ The randomness will help avoid more collisions.
 - 2) Limit Maximum Retransmission**
 - ✧ This method prevents congestion by reducing the number of retransmitted frames.
 - ✧ After a maximum number of retransmission-attempts K_{max} , a station must give up and try later (Figure 12.3).

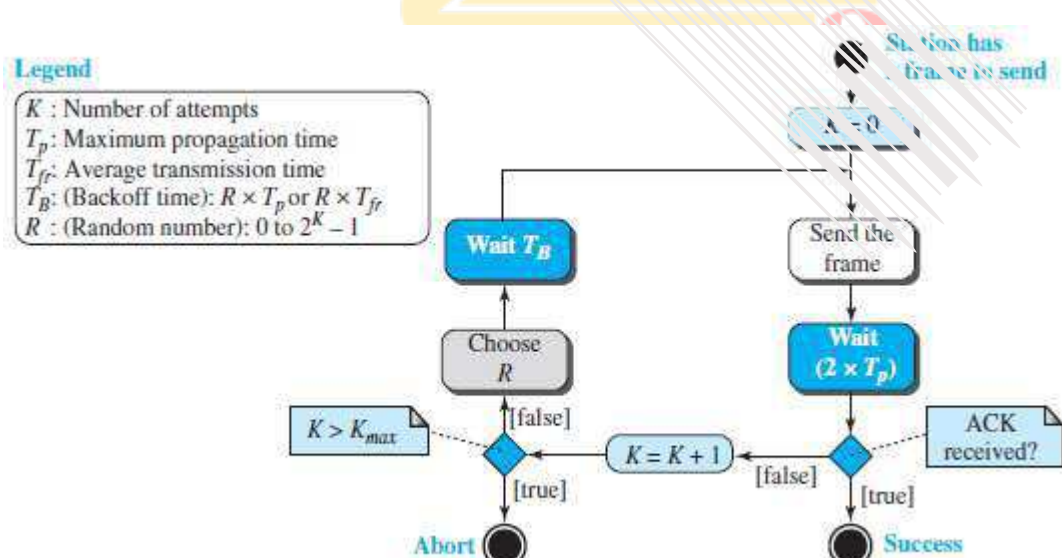


Figure 12.3 Procedure for pure ALOHA protocol

DATA COMMUNICATION

4.2.1.1.1 Vulnerable Time

- The vulnerable-time is defined as a time during which there is a possibility of collision.

$$\text{Pure ALOHA vulnerable time} = 2 \times T_{fr}$$

where T_{fr} = Frame transmission time

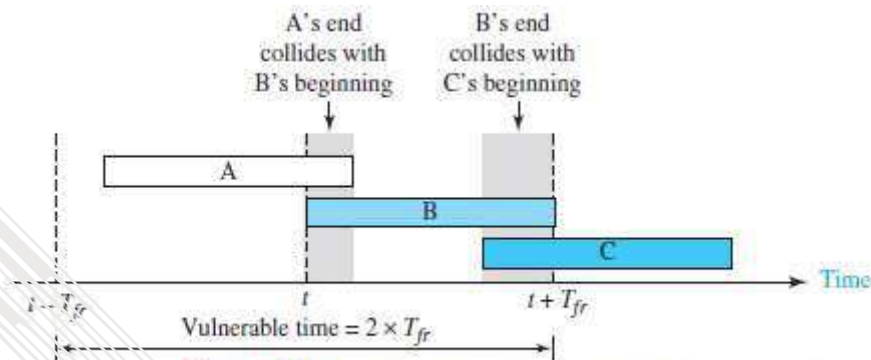


Figure 12.4 Vulnerable time for pure ALOHA protocol

- In Figure 12.4,
 - If station B sends a frame between $t - T_{fr}$ and t , this leads to a collision between the frames from station A and station B.
 - If station C sends a frame between t and $t + T_{fr}$, this leads to a collision between the frames from station A and station C.

Example 4.1

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms. The vulnerable time is $2 \times 1 \text{ ms} = 2 \text{ ms}$. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the period (1 ms) that this station is sending.

4.2.1.1.2 Throughput

- The average number of successful transmissions is given by

$$S = G \times e^{-2G}$$

where G = average no. of frames in one frame transmission time (T_{fr})

- For $G = 1$, the maximum throughput $S_{\max} = 0.184$.
- In other words, out of 100 frames, 18 frames reach their destination successfully.

Example 4.2

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

- a. 1000 frames per second? b. 500 frames per second? c. 250 frames per second?

Solution

The frame transmission time is 200/200 kbps or 1 ms.

- If the system creates 1000 frames per second, or 1 frame per millisecond, then $G = 1$. In this case $S = G \times e^{-2G} = 0.135$ (13.5 percent). This means that the throughput is $1000 \times 0.135 = 135$ frames. Only 135 frames out of 1000 will probably survive.
- If the system creates 500 frames per second, or 1/2 frames per millisecond, then $G = 1/2$. In this case $S = G \times e^{-2G} = 0.184$ (18.4 percent). This means that the throughput is $500 \times 0.184 = 92$ and that only 92 frames out of 500 will probably survive. Note that this is the *maximum* throughput case, percentagewise.
- If the system creates 250 frames per second, or 1/4 frames per millisecond, then $G = 1/4$. In this case $S = G \times e^{-2G} = 0.152$ (15.2 percent). This means that the throughput is $250 \times 0.152 = 38$. Only 38 frames out of 250 will probably survive.

DATA COMMUNICATION

4.2.1.2 Slotted ALOHA

- Slotted ALOHA was invented to improve the efficiency of pure ALOHA.
- The time is divided into time-slots of T_{fr} seconds (Figure 12.5).
- The stations are allowed to send only at the beginning of the time-slot.

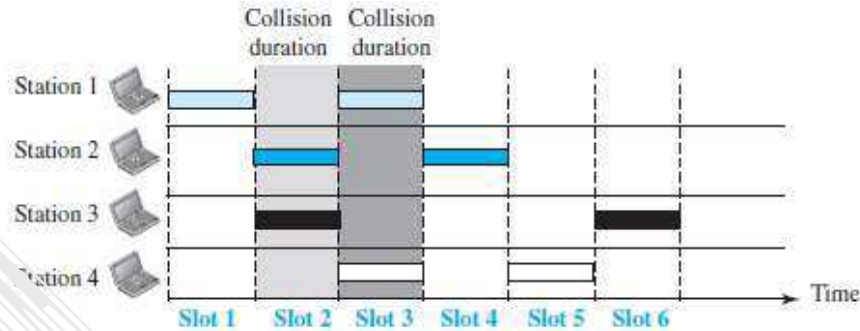


Figure 12.5 Frames in a slotted ALOHA network

- If a station misses the time-slot, the station must wait until the beginning of the next time-slot.
- If 2 stations try to resend at beginning of the same time-slot, the frames will collide again (Fig 12.6).

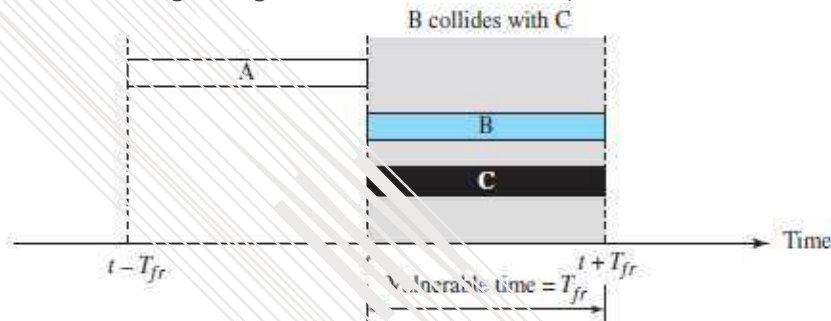


Figure 12.6 Vulnerable time for slotted ALOHA protocol

- The vulnerable time is given by:

$$\text{vulnerable time} = T_{fr}$$

4.2.1.2.1 Throughput

- The average number of successful transmissions is given by

$$S = G \times e^{-G}$$

- For $G = 1$, the maximum throughput $S_{\max} = 0.368$.
- In other words, out of 100 frames, 36 frames reach their destination successfully.

Example 4.3

A slotted ALOHA network transmits 200-bit frames using a shared channel with a 200 kbps bandwidth. Find the throughput if the system (all stations together) produces

- 1000 frames per second.
- 500 frames per second.
- 250 frames per second.

Solution

This situation is similar to the previous exercise except that the network is using slotted ALOHA instead of pure ALOHA. The frame transmission time is $200/200$ kbps or 1 ms.

- In this case G is 1. So $S = G \times e^{-G} = 0.368$ (36.8 percent). This means that the throughput is $1000 \times 0.368 = 368$ frames. Only 368 out of 1000 frames will probably survive. Note that this is the maximum throughput case, percentagewise.
- Here G is 1/2. In this case $S = G \times e^{-G} = 0.303$ (30.3 percent). This means that the throughput is $500 \times 0.303 = 151$. Only 151 frames out of 500 will probably survive.
- Now G is 1/4. In this case $S = G \times e^{-G} = 0.195$ (19.5 percent). This means that the throughput is $250 \times 0.195 = 49$. Only 49 frames out of 250 will probably survive.

DATA COMMUNICATION

4.2.2 CSMA

- CSMA was developed to minimize the chance of collision and, therefore, increase the performance.
- CSMA is based on the principle "sense before transmit" or "listen before talk."
- Here is how it works:
 - 1) Each station checks the state of the medium: idle or busy.
 - 2) i) If the medium is idle, the station sends the data.
ii) If the medium is busy, the station defers sending.
- CSMA can reduce the possibility of collision, but it cannot eliminate it.

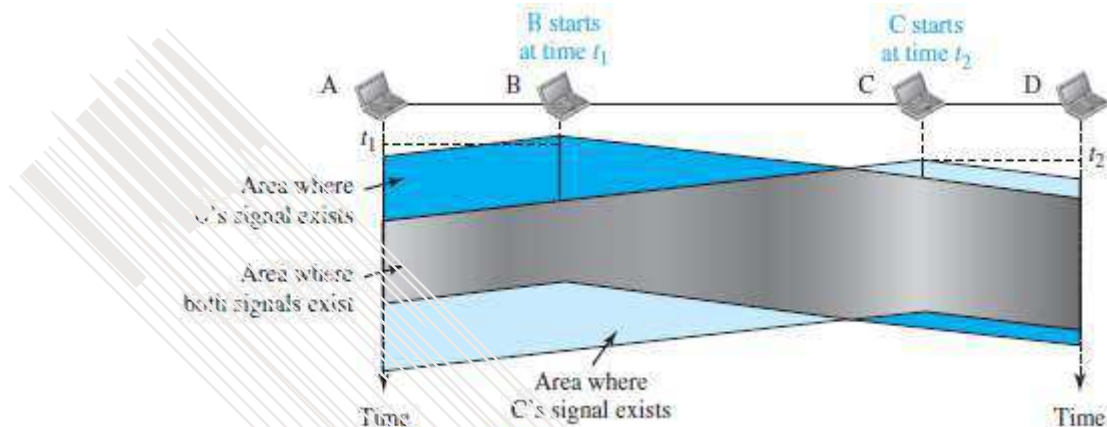


Figure 12.7 Space/time model of a collision in CSMA

- The possibility of collision still exists.
For example:
When a station sends a frame, it still takes time
→ for the first bit to reach every station and →
for every station to sense it.
- For example: In Figure 12.7,
 - At time t_1 , station B senses & finds the medium idle, so sends a frame.
 - At time t_2 , station C senses & finds the medium idle, so sends a frame.
 - The 2 signals from both stations B & C collide and both frames are destroyed.

4.2.2.1 Vulnerable Time

- The vulnerable time is the propagation time T_p (Figure 12.8).
- The propagation time is the time needed for a signal to propagate from one end of the medium to the other.
- Collision occurs when
 - a station sends a frame, and
 - other station also sends a frame during propagation time
- If the first bit of the frame reaches the end of the medium, every station will refrain from sending.

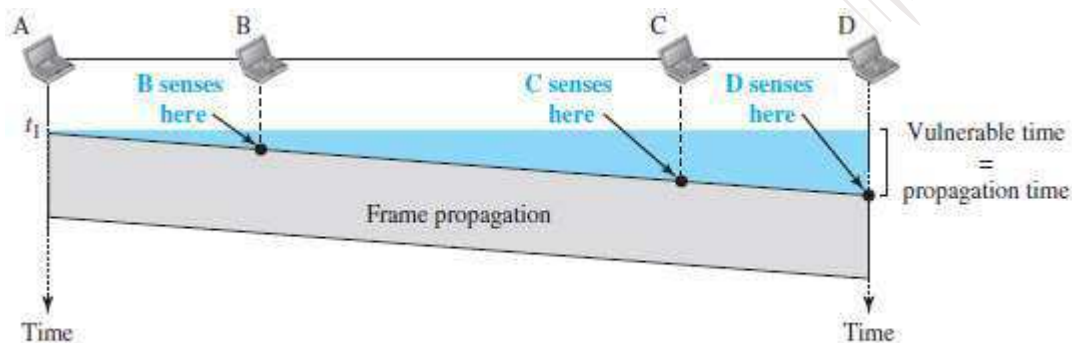


Figure 12.8 Vulnerable time in CSMA

DATA COMMUNICATION

4.2.2.2 Persistence Methods

- Q: What should a station do if the channel is busy or idle?

Three methods can be used to answer this question:

- 1) 1-persistent method
- 2) Non-persistent method
- 3) p-persistent method

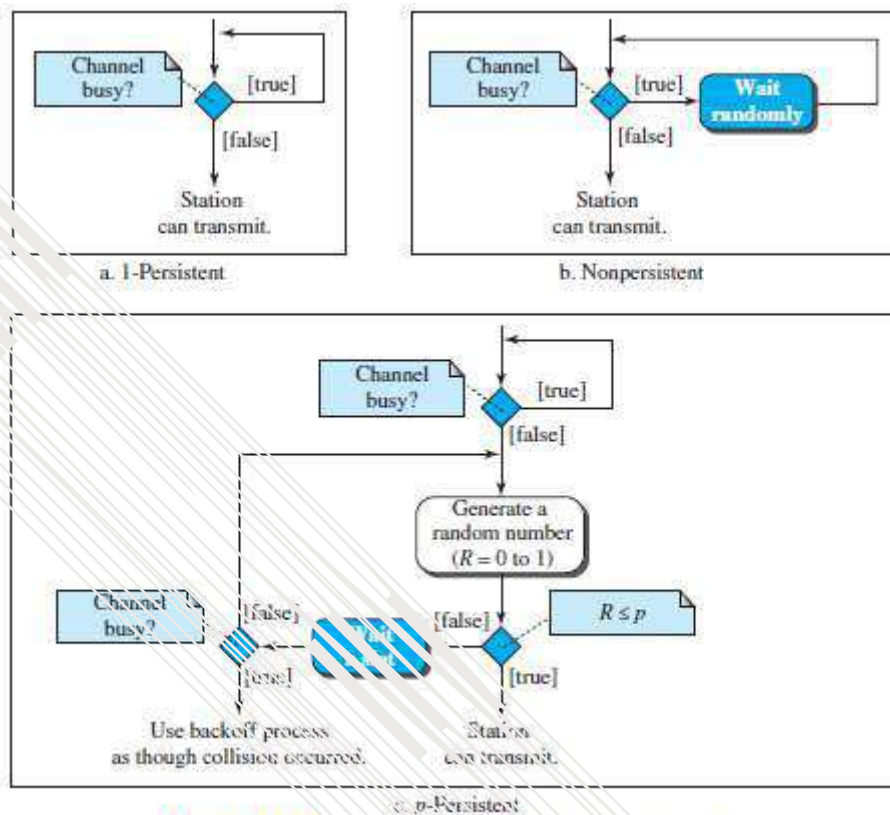


Figure 12.10 Flow diagram for three persistence methods

1) 1-Persistent

- Before sending a frame, a station senses the line (Figure 12.10a).
 - i) If the line is idle, the station sends immediately (with probability = 1).
 - ii) If the line is busy, the station continues sensing the line.
- This method has the highest chance of collision because 2 or more stations:
 - may find the line idle and
 - send the frames immediately.

2) Non-Persistent

- Before sending a frame, a station senses the line (Figure 12.10b).
 - i) If the line is idle, the station sends immediately.
 - ii) If the line is busy, the station waits a random amount of time and then senses the line again.
- This method reduces the chance of collision because 2 or more stations:
 - will not wait for the same amount of time and
 - will not retry to send simultaneously.

3) P-Persistent

- This method is used if the channel has time-slots with a slot-duration equal to or greater than the maximum propagation time (Figure 12.10c).
- Advantages:
 - i) It combines the advantages of the other 2 methods.
 - ii) It reduces the chance of collision and improves efficiency.
- After the station finds the line idle, it follows these steps:
 - 1) With probability p , the station sends the frame.
 - 2) With probability $q=1-p$, the station waits for the beginning of the next time-slot and checks the line again.
 - i) If line is idle, it goes to step 1.
 - ii) If line is busy, it assumes that collision has occurred and uses the back off procedure.

4.2.3 CSMA/CD

- Disadvantage of CSMA: CSMA does not specify the procedure after a collision has occurred.
Solution: CSMA/CD enhances the CSMA to handle the collision.

- Here is how it works (Figure 12.12):

- 1) A station
 - sends the frame &
 - then monitors the medium to see if the transmission was successful or not.
- 2) If the transmission was unsuccessful (i.e. there is a collision), the frame is sent again.

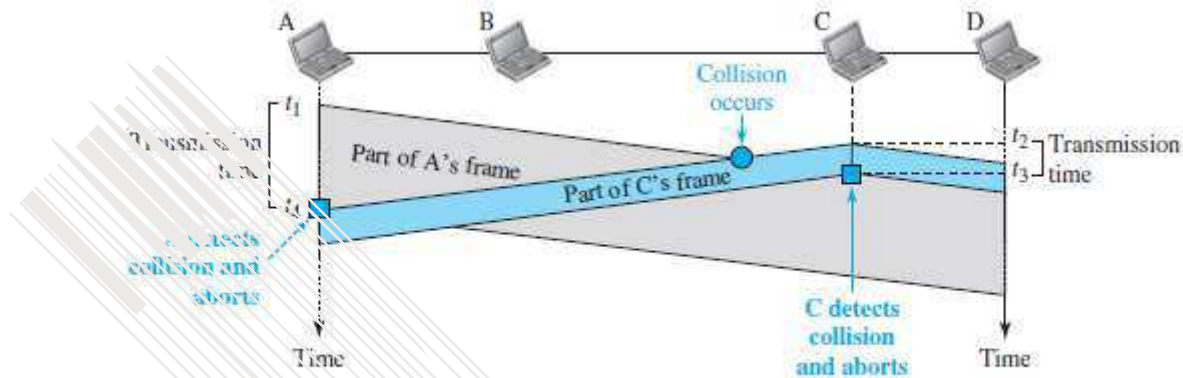


Figure 12.12 Collision and abortion in CSMA/CD

- In the Figure 12.11,
 - At time t_1 , station A has executed its procedure and starts sending the bits of its frame.
 - At time t_2 , station C has executed its procedure and starts sending the bits of its frame.
 - The collision occurs sometime after time t_2 .
 - Station C detects a collision at time t_3 when it receives the first bit of A's frame.
Station C immediately aborts transmission.
 - Station A detects collision at time t_4 when it receives the first bit of C's frame.
Station A also immediately aborts transmission.
- Station A transmits for the duration $t_4 - t_1$.
Station C transmits for the duration $t_3 - t_2$.
- For the protocol to work:
The length of any frame divided by the bit rate must be more than either of these durations.

4.2.3.1 Minimum Frame Size

- For CSMA/CD to work, we need to restrict the frame-size.
- Before sending the last bit of the frame, the sender must
 - detect a collision and
 - abort the transmission.
- This is so because the sender
 - does not keep a copy of the frame and
 - does not monitor the line for collision-detection.
- Frame transmission time T_{fr} is given by
$$T_{fr} = 2T_p \quad \text{where } T_p = \text{maximum propagation time}$$

Example 4.4

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is $25.6 \mu s$, what is the minimum size of the frame?

Solution

The minimum frame transmission time is $T_{fr} = 2 \times T_p = 51.2 \mu s$. This means, in the worst case, a station needs to transmit for a period of $51.2 \mu s$ to detect the collision. The minimum size of the frame is $10 \text{ Mbps} \times 51.2 \mu s = 512 \text{ bits}$ or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet, as we will see later in the chapter.

DATA COMMUNICATION

4.2.3.2 Procedure

- CSMA/CD is similar to ALOHA with 2 differences (Figure 12.13):
 - Addition of the persistence process.
 - We need to sense the channel before sending the frame by using non-persistent, 1-persistent or p-persistent.
 - Frame transmission.
 - In ALOHA, first the entire frame is transmitted and then acknowledgment is waited for.
 - In CSMA/CD, transmission and collision-detection is a continuous process.

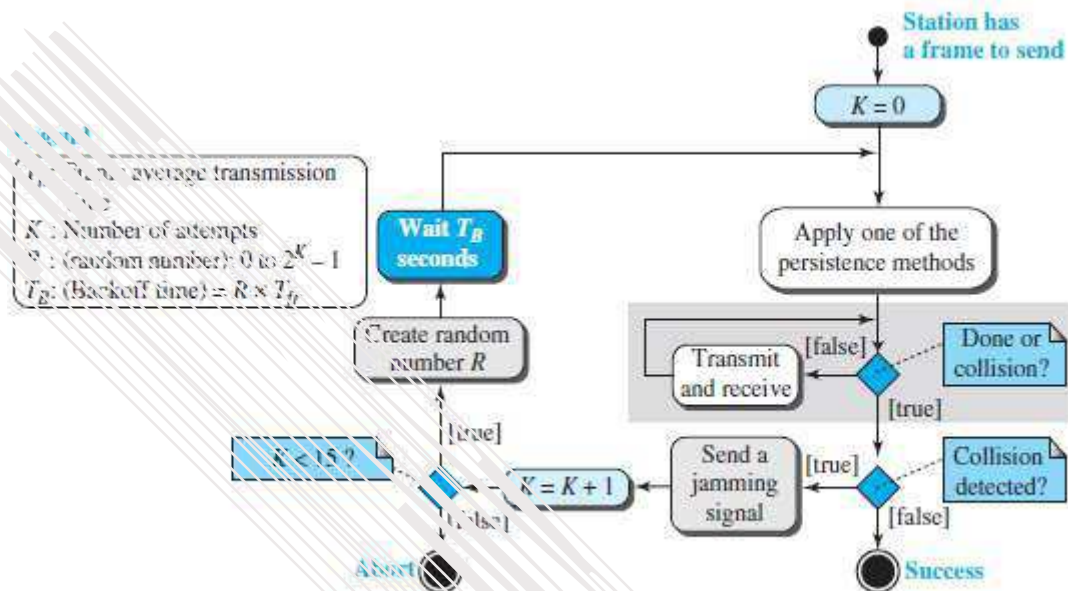


Figure 12.13 Flow diagram for the CSMA/CD

4.2.3.3 Energy Level

- In a channel, the energy-level can have 3 values: 1) Zero 2) Normal and 3) Abnormal.
 - At zero level, the channel is idle (Figure 12.14).
 - At normal level, a station has successfully captured the channel and is sending its frame.
 - At abnormal level, there is a collision and the level of the energy is twice the normal level.
- A sender needs to monitor the energy-level to determine if the channel is
 - Idle
 - Busy or
 - Collision mode

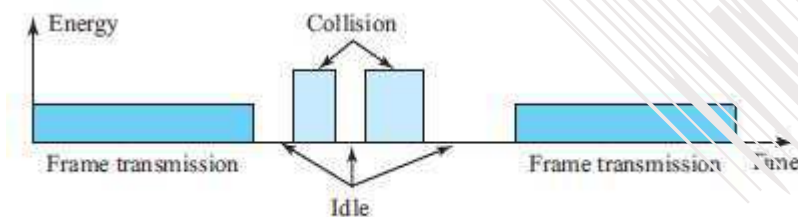


Figure 12.14 Energy level during transmission, idleness, or collision

4.2.3.4 Throughput

- The throughput of CSMA/CD is greater than pure or slotted ALOHA.
- The maximum throughput is based on
 - different value of G
 - persistence method used (non-persistent, 1-persistent, or p-persistent) and → 'p' value in the p-persistent method.
- For 1-persistent method, the maximum throughput is 50% when $G = 1$.
- For non-persistent method, the maximum throughput is 90% when G is between 3 and 8.

DATA COMMUNICATION

4.2.4 CSMA/CA

- Here is how it works (Figure 12.15):

- 1) A station needs to be able to receive while transmitting to detect a collision.
 - i) When there is no collision, the station receives one signal: its own signal.
 - ii) When there is a collision, the station receives 2 signals:
 - a) Its own signal and
 - b) Signal transmitted by a second station.
- 2) To distinguish b/w these 2 cases, the received signals in these 2 cases must be different.

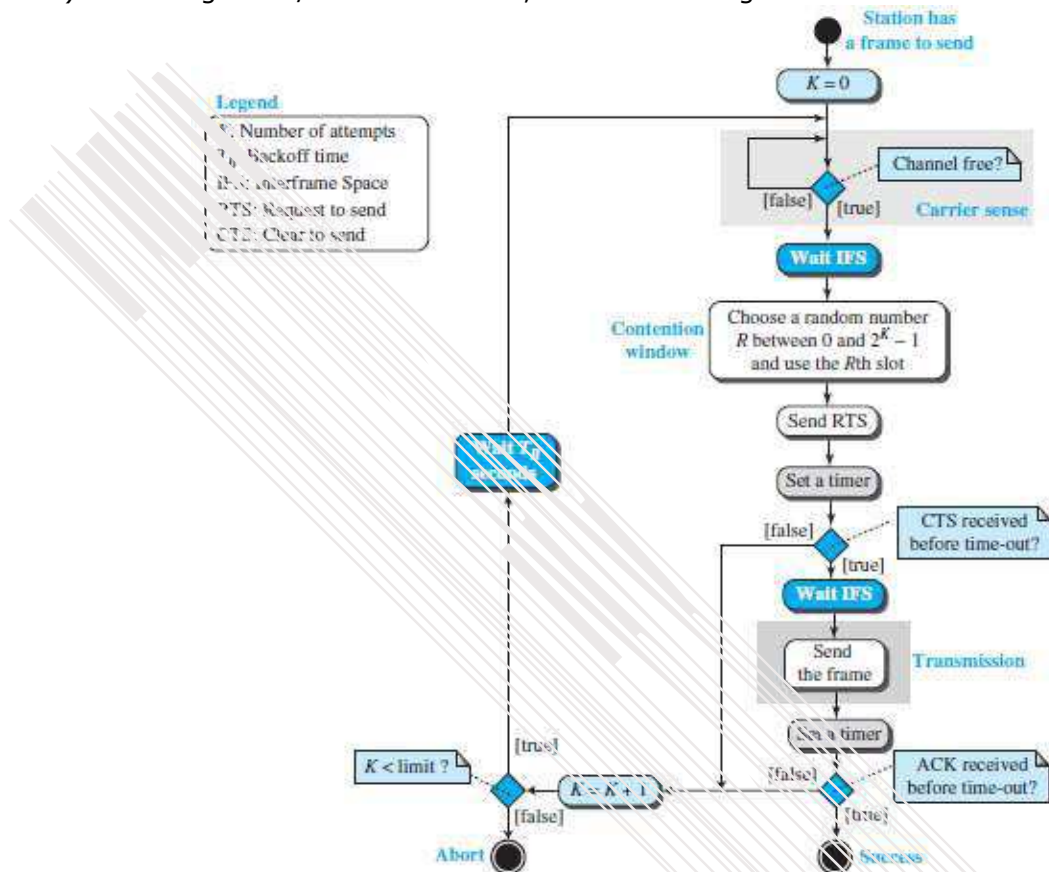


Figure 12.15 Flow diagram of CSMA/CA

- CSMA/CA was invented to avoid collisions on wireless networks.
- Three methods to avoid collisions (Figure 12.16):
 - 1) Interframe space
 - 2) Contention window
 - 3) Acknowledgments

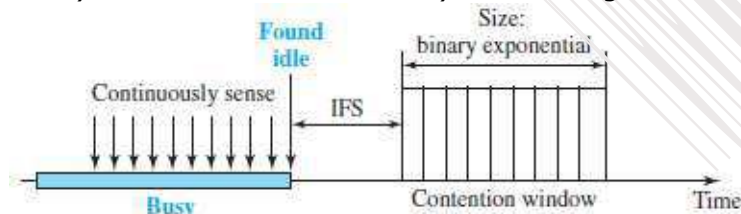


Figure 12.16 Contention window

1) Interframe Space (IFS)

- Collisions are avoided by deferring transmission even if the channel is found idle.
- When the channel is idle, the station does not send immediately. Rather, the station waits for a period of time called the inter-frame space or IFS.
- After the IFS time,
 - if the channel is still idle,
 - then, the station waits for the contention-time & finally, the station sends the frame.
- IFS variable can also be used to prioritize stations or frame types. For example, a station that is assigned a shorter IFS has a higher priority.

DATA COMMUNICATION

2) Contention Window

- The contention-window is an amount of time divided into time-slots.
- A ready-station chooses a random-number of slots as its wait time.
- In the window, the number of slots changes according to the binary exponential back-off strategy.
- For example:
 - At first time, number of slots is set to one slot and
 - Then, number of slots is doubled each time if the station cannot detect an idle channel.

3) Acknowledgment

- There may be a collision resulting in destroyed-data.
- In addition, the data may be corrupted during the transmission.
- To help guarantee that the receiver has received the frame, we can use
 - i) Positive acknowledgment and
 - ii) Time-out timer

4.2.4.1 Frame Exchange Time Line

- Two control frames are used:
 - 1) Request to send (RTS)
 - 2) Clear to send (CTS)
- The procedure for exchange of data and control frames in time (Figure 12.17):
 - 1) The source senses the medium by checking the energy level at the carrier frequency.
 - ii) If the medium is idle, then the source waits for a period of time called the DCF interframe space (DIFS); finally, the source sends a RTS.
 - 2) The destination
 - receives the RTS
 - waits a period of time called the short interframe space (SIFS) →
 - sends a control frame CTS to the source.
 - CTS indicates that the destination station is ready to receive data.
 - 3) The source
 - receives the CTS
 - waits a period of time SIFS
 - sends a data to the destination
 - 4) The destination
 - receives the data
 - waits a period of time SIFS
 - sends a acknowledgment ACK to the source.
 - ACK indicates that the destination has been received the frame.

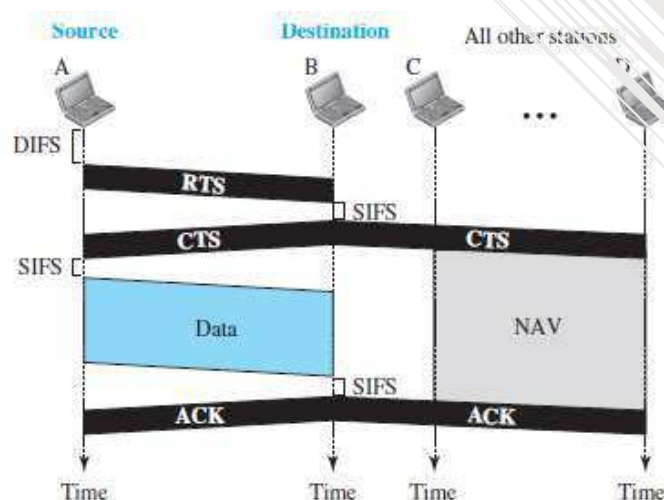


Figure 12.17 CSMA/CA and NAV

DATA COMMUNICATION

4.2.4.2 Network Allocation Vector

- When a source-station sends an RTS, it includes the duration of time that it needs to occupy the channel.
- The remaining stations create a timer called a network allocation vector (NAV).
- NAV indicates waiting time to check the channel for idleness.
- Each time a station accesses the system and sends an RTS frame, other stations start their NAV.

4.2.4.3 Collision during Handshaking

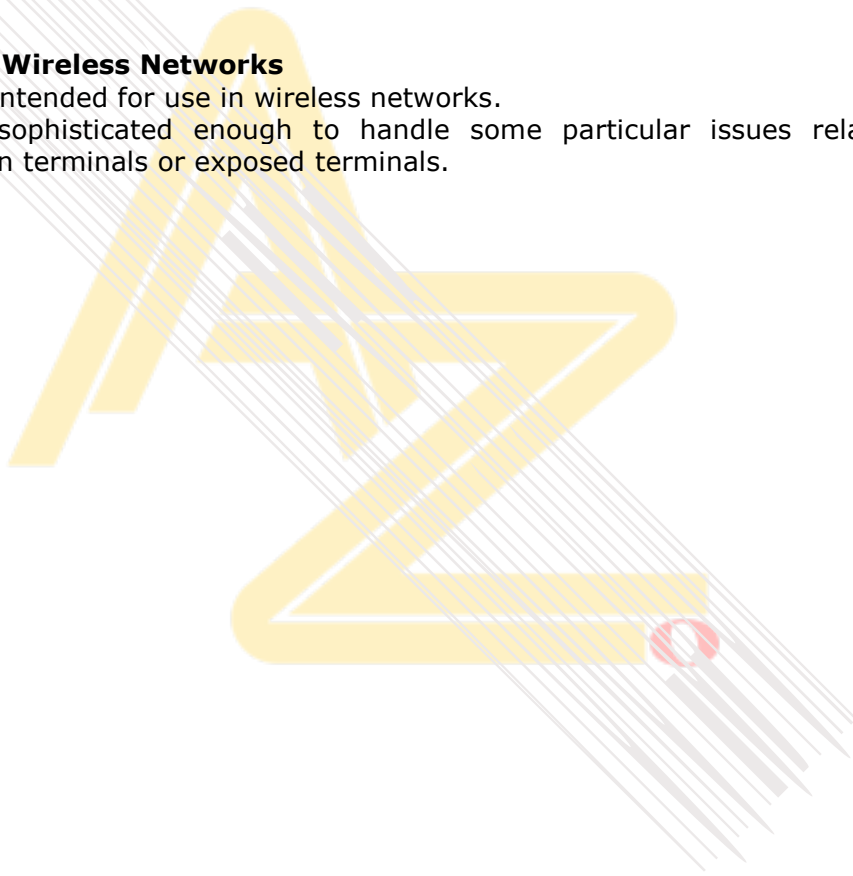
- Two or more stations may try to send RTS at the same time.
- These RTS may collide.
- The source assumes there has been a collision if it has not received CTS from the destination.
- The backoff strategy is employed, and the source tries again.

4.2.4.4 Hidden Station Problem

- Figure 12.17 also shows that the RTS from B reaches A, but not C.
- However, because both B and C are within the range of A, the CTS reaches C.
- Station C knows that some hidden station is using the channel and refrains from transmitting until that duration is over.

4.2.4.5 CSMA/CA and Wireless Networks

- CSMA/CA was mostly intended for use in wireless networks.
- However, it is not sophisticated enough to handle some particular issues related to wireless networks, such as hidden terminals or exposed terminals.



DATA COMMUNICATION

4.3 CONTROLLED ACCESS PROTOCOLS

- Here, 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.
- Three popular controlled-access methods are: 1) Reservation 2) Polling 3) Token Passing

4.3.1 Reservation

- Before sending data, each station needs to make a reservation of the medium.
- Time is divided into intervals.
- In each interval, a reservation-frame precedes the data-frames.
- If no. of stations = N, then there are N reservation mini-slots in the reservation-frame.
- Each mini-slot belongs to a station.
- When a station wants to send a data-frame, it makes a reservation in its own minislot.
- The stations that have made reservations can send their data-frames.

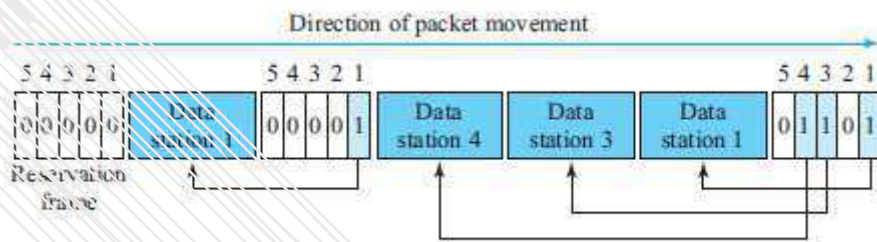


Figure 12.18 Reservation access method

- For example (Figure 12.18):
 - 5 stations have a 5-minislot reservation-frame.
 - In the first interval, only stations 1, 3, and 4 have made reservations.
 - In the second interval, only station-1 has made a reservation.

DATA COMMUNICATION

4.3.2 Polling

- In a network,
 - One device is designated as a primary station and
 - Other devices are designated as secondary stations.
- Functions of primary-device:
 - 1) The primary-device controls the link.
 - 2) The primary-device is always the initiator of a session.
 - 3) The primary-device determines which device is allowed to use the channel at a given time.
 - 4) All data exchanges must be made through the primary-device.
- The secondary devices follow instructions of primary-device.
- Disadvantage: If the primary station fails, the system goes down.
- Poll and select functions are used to prevent collisions (Figure 12.19).

1) Select

- If the primary wants to send data, it tells the secondary to get ready to receive; this is called select function.
- The primary
 - alerts the secondary about upcoming transmission by sending select frame (SEL) →
 - then waits for an acknowledgment (ACK) from secondary
 - then sends the data frame and
 - finally waits for an acknowledgment (ACK) from the secondary.

2) Poll

- If the primary wants to receive data, it asks the secondaries if they have anything to send; this is called poll function.
- When the first secondary is approached, it responds either
 - with a NAK frame if it has no data to send or
 - with data-frame if it has data to send.
- i) If the response is negative (NAK frame), then the primary polls the next secondary in the same manner.
- ii) When the response is positive (a data-frame), the primary
 - reads the frame and
 - returns an acknowledgment (ACK frame).

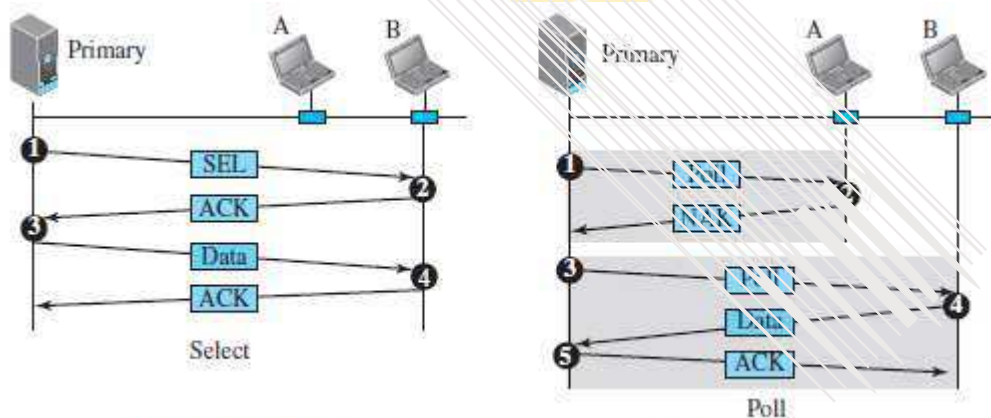


Figure 12.19 Select and poll functions in polling-access method

DATA COMMUNICATION

4.3.3 Token Passing

- In a network, the stations are organized in a ring fashion i.e. for each station; there is a predecessor and a successor.
 - 1) The predecessor is the station which is logically before the station in the ring.
 - 2) 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.
- A token is a special packet that circulates through the ring.
- Here is how it works:
 - A station can send the data only if it has the token.
 - When a station wants to send the data, it waits until it receives the token from its predecessor.
 - Then, the station holds the token and sends its data.
 - When the station finishes sending the data, the station
 - releases the token
 - passes the token to the successor.
- Main functions of token management:
 - 1) Stations must be limited in the time they can hold the token.
 - 2) The token must be monitored to ensure it has not been lost or destroyed.
 - For ex: if a station that is holding the token fails, the token will disappear from the network
 - 3) Assign priorities
 - to the stations and
 - to the types of data being transmitted.
 - 4) Make low-priority stations release the token to high priority stations.

4.3.3.1 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.
- Four physical topologies to create a logical ring (Figure 12.20):
 - 1) Physical ring
 - 2) Dual ring
 - 3) Bus ring
 - 4) Star ring

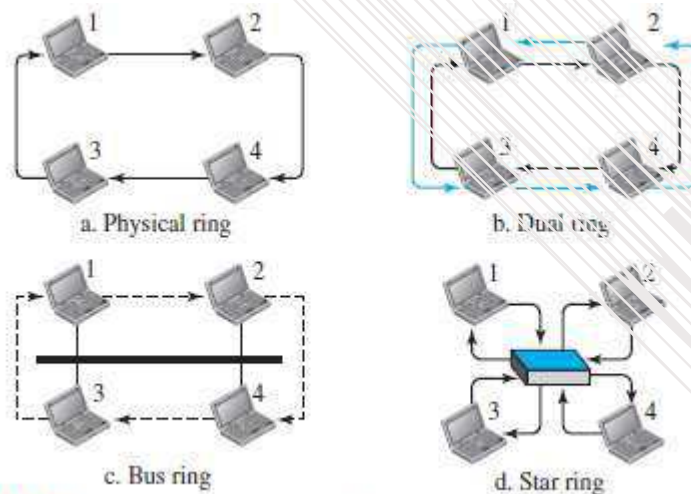


Figure 12.20 Logical ring and physical topology in token-passing access method

1) Physical Ring Topology

- When a station sends token to its successor, token cannot be seen by other stations. (Figure 12.20a)
- This means that the token does not have the address of the next successor.
- Disadvantage: If one of the links fails, the whole system fails.

DATA COMMUNICATION

2) Dual Ring Topology

- A second (auxiliary) ring
 - is used along with the main ring (Figure 12.20b).
 - operates in the reverse direction compared with the main ring.
 - is used for emergencies only (such as a spare tire for a car).
- If the main ring fails, the system automatically combines the 2 rings to form a temporary ring.
- After the failed link is restored, the **second** ring becomes idle again.
- Each station needs to have 2 transmitter-ports and 2 receiver-ports.
- This topology is used in
 - i) FDDI (Fiber Distributed Data Interface) and
 - ii) CDDI (Copper Distributed Data Interface).

3) Bus Ring Topology

- The stations are connected to a single cable called a bus (Figure 12.20c).
- This makes a logical ring, because each station knows the address of its successor and predecessor.
- When a station has finished sending its data, the station
 - releases the token and
 - inserts the address of its successor in the token.
- Only the station gets the token to access the shared media.
- This topology is used in the Token Bus LAN.

4) Star Ring Topology

- The physical topology is a star (Figure 12.20d).
- There is a hub that acts as the connector.
- The wiring inside the hub makes the ring i.e. the stations are connected to the ring through the 2 wire connections.
- Disadvantages:
 - 1) This topology is less prone to failure because
 - If a link goes down,
 - then the link will be bypassed by the hub and
 - the rest of the stations can operate.
 - 2) Also adding and removing stations from the ring is easier.
- This topology is used in the Token Ring LAN.

DATA COMMUNICATION

4.4 CHANNELIZATION PROTOCOLS

- Channelization is a multiple-access method.
- The available bandwidth of a link is shared b/w different stations in time, frequency, or through code.
- Three channelization protocols:
 - 1) FDMA (Frequency Division Multiple Access)
 - 2) TDMA (Time Division Multiple Access) and
 - 3) CDMA (Code Division Multiple Access)

4.4.1 FDMA

- The available bandwidth is divided into frequency-bands (Figure 12.21).
- Each band is reserved for a specific station.
- Each station can send the data in the allocated band.
- Each station also uses a bandpass filter to confine the transmitter frequencies.
- To prevent interferences, small guard bands are used to separate the allocated bands from one another.

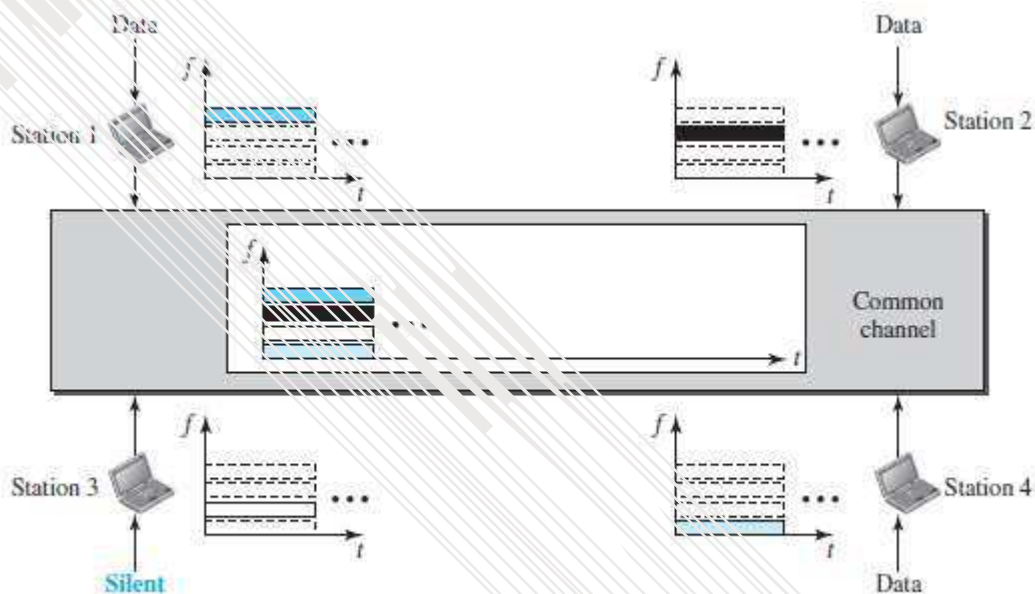


Figure 12.21 Frequency-division multiple access (FDMA)

- FDM vs. FDMA

1) FDM

- FDM is a multiplexing method in the physical layer.
- FDM
 - combines individual-loads from low-bandwidth channels and
 - transmits aggregated-load by using a high-bandwidth channel.
- The channels that are combined are low-pass.
- The multiplexer
 - modulates & combines the signals and
 - creates a bandpass signal.
- The bandwidth of each channel is shifted by the multiplexer.

2) FDMA

- FDMA is an access method in the data link layer.
- In each station, the data link layer tells the physical layer to make a bandpass signal from the data passed to it.
- The signal must be created in the allocated band.
- There is no physical multiplexer at the physical layer.
- The signals created at each station are automatically bandpass-filtered.
- They are mixed when they are sent to the common channel.

DATA COMMUNICATION

4.4.2 TDMA

- The stations share the bandwidth of the channel in time (Figure 12.22).
- Each time-slot is reserved for a specific station.
- Each station can send the data in the allocated time-slot.

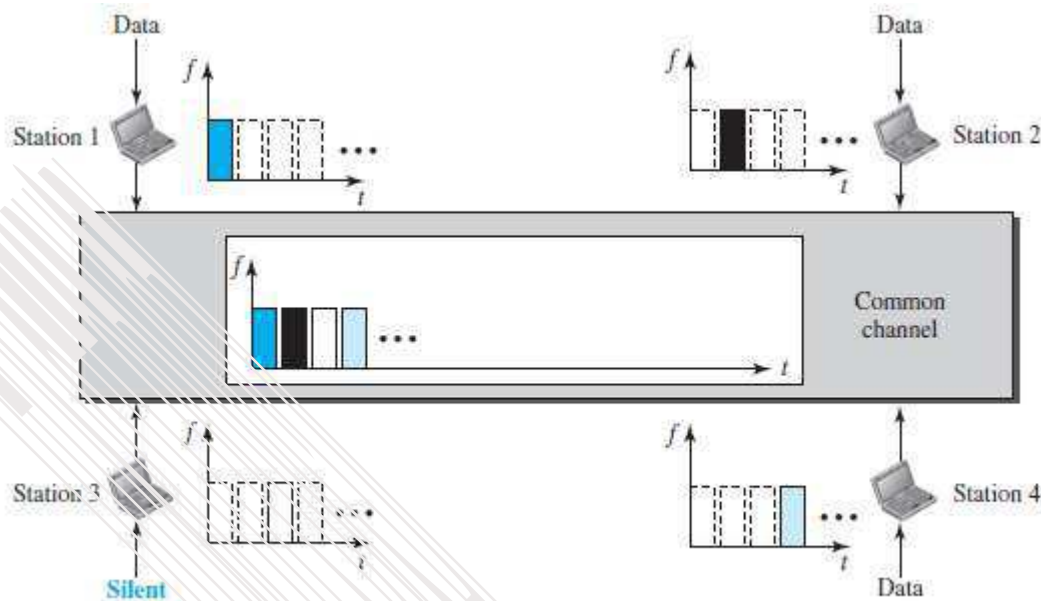


Figure 12.22 Time-division multiple access (TDMA)

- Main problem: Achieving synchronization between the different stations.
i.e. 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.
- To compensate for the delays, we can insert guard-times.
- Normally, synchronization is accomplished by having some synchronization bits at the beginning of each slot.
- TDMA vs. TDM
 - 1) **TDM**
 - TDM is a multiplexing method in the physical layer.
 - TDM
 - combines the individual-data from slower channels and
 - transmits the aggregated- data by using a faster channel.The multiplexer interleaves data units from each channel.
 - 2) **TDMA**
 - TDMA is an access method in the data link layer.
 - In each station, the data link layer tells the physical layer to use the allocated time-slot.There is no physical multiplexer at the physical layer.

DATA COMMUNICATION

4.4.3 CDMA

- CDMA simply means communication with different codes.
- CDMA differs from FDMA because
 - only one channel occupies the entire bandwidth of the link.
- CDMA differs from TDMA because
 - all stations can send data simultaneously; there is no timesharing.

(Analogy: CDMA simply means communication with different codes.

For example, in a large room with many people, 2 people can talk privately in English if nobody else understands English. Another 2 people can talk in Chinese if they are the only ones who understand Chinese, and so on).

4.4.3.1 Implementation

- 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 2 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).

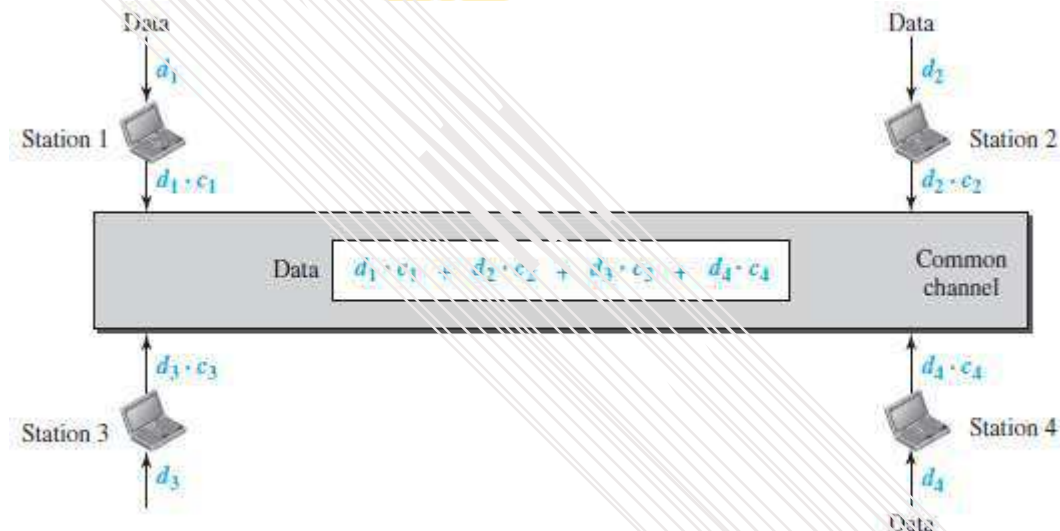


Figure 12.23 Simple idea of communication with codes

- Here is how it works (Figure 12.23):
 - Station-1 multiplies the data by the code to get $d_1 \cdot c_1$.
 - Station-2 multiplies the data by the code to get $d_2 \cdot c_2$. And so on.
 - The data that go on the channel are the sum of all these terms.
$$d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4$$
 - The receiver multiplies the data on the channel by the code of the sender.
- For example, suppose stations 1 and 2 are talking to each other.
- Station-2 wants to hear what station-1 is saying.
 - Station-2 multiplies the data on the channel by c_1 the code of station-1.
$$(c_1 \cdot c_1) = 4, (c_2 \cdot c_1) = 0, (c_3 \cdot c_1) = 0, \text{ and } (c_4 \cdot c_1) = 0,$$

Therefore, station-2 divides the result by 4 to get the data from station-1.

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

DATA COMMUNICATION

4.4.3.2 Chips

- CDMA is based on coding theory.
- Each station is assigned a code, which is a sequence of numbers called chips (Figure 12.24).

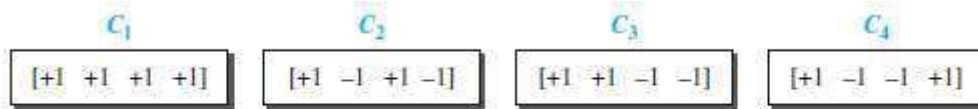


Figure 12.24 Chip sequences

- These sequences were carefully selected & are called orthogonal sequences
- These sequences have the following properties:
 - 1) Each sequence is made of N elements, where N is the number of stations.
 - 2) Multiplication of a sequence by a scalar:
If we multiply a sequence by a number i.e. every element in the sequence is multiplied by that element.
For example,
 $2 \times [+1 +1 -1 -1] = [+2 +2 -2 -2]$
 - 3) Inner product of 2 equal sequences:
If we multiply 2 equal sequences, element by element, and add the results, we get N , where N is the number of elements in the each sequence.
For example,
 $[+1 +1 -1 -1] \cdot [+1 +1 -1 -1] = 1+1+1+1=4$
 - 4) Inner product of 2 different sequences:
If we multiply 2 different sequences, element by element, and add the results, we get 0.
For example,
 $[+1 +1 -1 -1] \cdot [+1 +1 +1 -1] = 1+1-1-1=0$
 - 5) Adding 2 sequences means adding the corresponding elements. The result is another sequence.
For example,
 $[+1 +1 -1 -1] + [+1 +1 +1 -1] = [+2 +2 0 0]$

4.4.3.3 Data Representation

- We follow the following rules for encoding:
 - 1) To send a 0 bit, a station encodes the bit as -1
 - 2) To send a 1 bit, a station encodes the bit as +1
 - 3) When a station is idle, it sends no signal, which is interpreted as a 0.

DATA COMMUNICATION

4.4.3.4 Encoding and Decoding

- We assume that
 - Stations 1 and 2 are sending a 0 bit.
 - Station-4 is sending a 1 bit.
 - Station-3 is silent.
- Here is how it works (Figure 12.26):
 - At the sender-site, the data are translated to -1, -1, 0, and +1.
 - Each station multiplies the corresponding number by its chip (its orthogonal sequence).
 - The result is a new sequence which is sent to the channel.
 - The sequence on the channel is the sum of all 4 sequences.
 - Now imagine station-3, which is silent, is listening to station-2.
 - Station-3 multiplies the total data on the channel by the code for station-2, which is [+1 -1 +1 -1], to get

$$[-1 \ -1 \ 0 \ +1] \times [+1 \ -1 \ +1 \ -1] = -4/4 = -1 \rightarrow \text{bit 1}$$

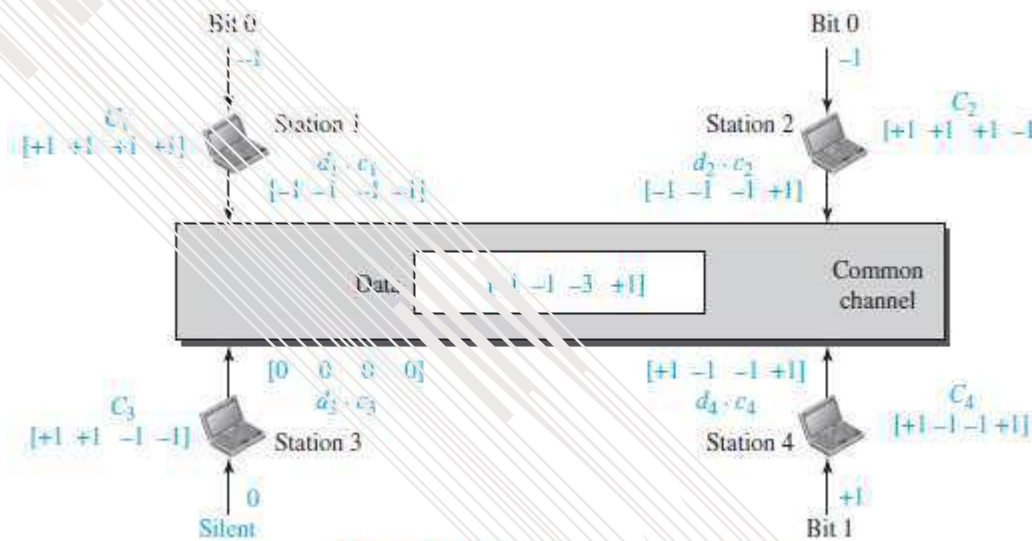


Figure 12.26 Sharing channel in CDMA

4.4.3.5 Sequence Generation

- To generate chip sequences, we use a Walsh table (Figure 12.29).
- Walsh table is a 2-dimensional table with an equal number of rows and columns.

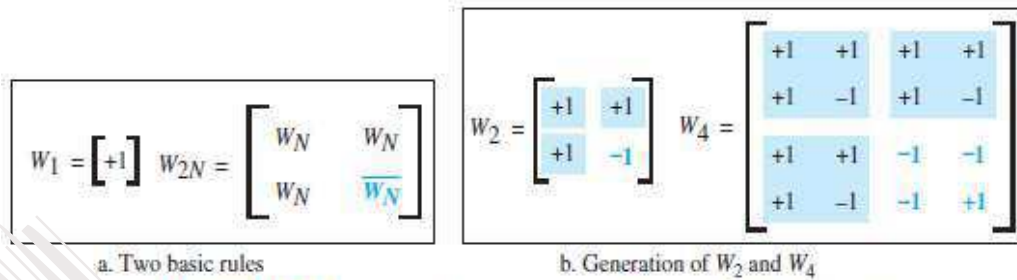


Figure 12.29 General rule and examples of creating Walsh tables

- In the Walsh table, each row is a sequence of chips.
- W_1 for a one-chip sequence has one row and one column. We can choose -1 or +1 for the chip for this trivial table (we chose +1).
- According to Walsh, if we know the table for N sequences W_N , we can create the table for $2N$ sequences W_{2N} (Figure 12.29).
- The W_N with the overbar $\overline{W_N}$ stands for the complement of W_N where each +1 is changed to -1 and vice versa.
- After we select W_1 , W_2 can be made from four W_1 's, with the last one the complement of W_1 .
- After W_2 is generated, W_4 can be made of four W_2 's, with the last one the complement of W_2 .
- The number of sequences in a Walsh table needs to be $N = 2^m$.

Example 4.5

Find the chips for a network with

- Two stations
- Four stations

Solution

We can use the rows of W_2 and W_4 in Figure 12.29:

- For a two-station network, we have $[+1 \ +1]$ and $[+1 \ -1]$.
- For a four-station network we have $[+1 \ +1 \ +1 \ +1]$, $[+1 \ -1 \ +1 \ -1]$, $[+1 \ +1 \ -1 \ -1]$, and $[+1 \ -1 \ -1 \ +1]$.

Example 4.6

What is the number of sequences if we have 90 stations in our network?

Solution

The number of sequences needs to be 2^m . We need to choose $m = 7$ and $N = 2^7$ or 128. We can then use 90 of the sequences as the chips.

Example 4.7

Prove that a receiving station can get the data sent by a specific sender if it multiplies the entire data on the channel by the sender's chip code and then divides it by the number of stations.

Solution

Let us prove this for the first station, using our previous four-station example. We can say that the data on the channel $D = (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4)$. The receiver that wants to get the data sent by station 1 multiplies these data by c_1 .

$$\begin{aligned} D \cdot c_1 &= (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4) \cdot c_1 \\ &= d_1 \cdot c_1 \cdot c_1 + d_2 \cdot c_2 \cdot c_1 + d_3 \cdot c_3 \cdot c_1 + d_4 \cdot c_4 \cdot c_1 \\ &= d_1 \times N + d_2 \times 0 + d_3 \times 0 + d_4 \times 0 \\ &= d_1 \times N \end{aligned}$$

When we divide the result by N , we get d_1 .