Computer Network
Unit-III

Data Link Layer

By:- Dr. D.R.Patil

• Outline

- Design Issues:
  - Framing, Error Control: Error Detection and Correction (Hamming Code, CRC, Checksum).
  - Flow Control: Stop and Wait, Sliding Window (Go-Back- N, Selective Repeat),
- Elementary Data Link Protocols, HDLC, PPP.
- Medium Access Control Sublayer:
  - Channel Allocation Problem,
  - Multiple Access Protocols (ALOHA,
  - Carrier Sense Multiple Access (CSMA/CA, CSMA/CD)),
- Wired LANS: Ethernet, Ethernet Standards, Virtual LANS.

- **Data Link Layer**
  - The data link layer uses the services of the physical layer to send and receive bits over communication channels.
  - It has a number of functions, including:
    - Providing a well-defined service interface to the network layer.
    - Dealing with transmission errors.
    - Regulating the flow of data so that slow receivers are not swamped by fast senders.
  - To accomplish these goals, the data link layer takes the packets it gets from the network layer and encapsulates them into frames for transmission.
  - Each frame contains a frame header, a payload field for holding the packet, and a frame trailer, as illustrated in Fig.

Figure 3-1. Relationship between packets and frames.

- **Services Provided to the Network Layer**
  - The function of the data link layer is to provide services to the network layer.
  - The principal service is transferring data from the network layer on the source machine to the network layer on the destination machine.
  - On the source machine is an entity, call it a process, in the network layer that hands some bits to the data link layer for transmission to the destination.
  - The job of the data link layer is to transmit the bits to the destination machine so they can be handed over to the net- work layer there, as shown in Fig. 3-2(a).
  - The actual transmission follows the path of Fig. 3-2(b), but it is easier to think in terms of two data link layer processes communicating using a data link protocol.

Figure 3-2. (a) Virtual communication. (b) Actual communication.

- The data link layer can be designed to offer various services.
- The actual services that are offered vary from protocol to protocol.
- Three reasonable possibilities that we will consider in turn are:
  - Unacknowledged connectionless service.
  - Acknowledged connectionless service.
  - Acknowledged connection-oriented service.

- **Unacknowledged connectionless service.**
  - It consists of having the source machine send independent frames to the destination machine without having the destination machine acknowledge them.
  - Ethernet is a good example of a data link layer that provides this class of service.
  - No logical connection is established beforehand or released afterward.
  - If a frame is lost due to noise on the line, no attempt is made to detect the loss or recover from it in the data link layer.

- **Acknowledged connectionless service.**
  - The next step up in terms of reliability is acknowledged connectionless service.
  - When this service is offered, there are still no logical connections used, but each frame sent is individually acknowledged.
  - In this way, the sender knows whether a frame has arrived correctly or been lost.
  - If it has not arrived within a specified time interval, it can be sent again.
  - This service is useful over unreliable channels, such as wireless systems.
  - 802.11 (WiFi) is a good example of this class of service.

- **Acknowledged connection-oriented service**
  - The most sophisticated service the data link layer can provide to the network layer is connection-oriented service.
  - With this service, the source and destination machines establish a connection before any data are transferred.
  - Each frame sent over the connection is numbered, and the data link layer guarantees that each frame sent is indeed received.
  - Furthermore, it guarantees that each frame is received exactly once and that all frames are received in the right order.
  - Connection-oriented service thus provides the network layer processes with the equivalent of a reliable bit stream.
  - It is appropriate over long, unreliable links such as a satellite channel or a long-distance telephone circuit.

- **Data Link Layer**
  - **Objectives**
  - The data link layer transforms the physical layer, a raw transmission facility, to a link responsible for node-to-node (hop-to-hop) communication.
  - Specific responsibilities of the data link layer include
    - framing,
    - addressing,
    - flow control,
    - error control, and
    - media access control.
  - The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
  - The data link layer adds a header to the frame to define the addresses of the sender and receiver of the frame.

- If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.
- The data link layer also adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged, duplicate, or lost frames.
- When two or more devices are connected to the same link, data link layer protocols arenecessary to determine which device has control over the link at any given time.

- **FRAMING**
  - Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination.
  - The physical layer provides synchronization to ensure that the sender and receiver use the same bit durations and timing.
  - The data link layer, on the other hand, needs to pack bits into frames, so that each frame is distinguishable from another.
  - Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, 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.

- **Fixed-Size Framing**
  - 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.
  - An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells.
- **Variable-Size Framing**
  - In variable-size framing, we need a way to define the end of the frame and the beginning of the next.
  - Historically, two approaches were used for this purpose: a character-oriented approach and a bit-oriented approach.

- **There are four methods of Framing**
  - Character count
  - Flag bytes with byte stuffing.
  - Starting and ending flags, with bit stuffing.
  - Physical layer coding violations
- **Character count**
  - The first framing method uses a field in the header to specify the number of characters in the frame.
  - When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is.

A character stream. (a) Without errors. (b) With one error.

- The trouble with this algorithm is that the count can be garbled by a transmission error.
- For example, if the character count of 5 in the second frame of Fig. (b) becomes a 7, the destination will get out of synchronization and will be unable to locate the start of the next frame.
- Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts.
- Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission.
- For this reason, the character count method is rarely used anymore.

- **Flag bytes with byte stuffing**
- The second framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes.
- In the past, the starting and ending bytes were different, but in recent years most protocols have used the same byte, called a flag byte, as both the starting and ending delimiter, as shown in Fig. 3-5(a) as FLAG.
- In this way, if the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame. Two consecutive flag bytes indicate the end of one frame and start of the next one.

Figure 3-5. (a) A frame delimited by flag bytes.
(b) Four examples of byte sequences before and after byte stuffing.

(a) A frame delimited by flag bytes.
(b) Four examples of byte sequences before and after stuffing.

- A serious problem occurs with this method when binary data, such as object programs or floating-point numbers, are being transmitted.
- It may easily happen that the flag byte's bit pattern occurs in the data.
- This situation will usually interfere with the framing.
- One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each "accidental" flag byte in the data.
- The data link layer on the receiving end removes the escape byte before the data are given to the network layer.
- This technique is called **byte stuffing or character stuffing**.
- Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.

- **A major disadvantage** of using this framing method is that it is closely tied to the use of 8-bit characters. Not all character codes use 8-bit characters.
- For example. UNICODE uses 16-bit characters, As networks developed, the disadvantages of embedding the character code length in the framing mechanism became more and more obvious, so a new technique had to be developed to allow arbitrary sized characters.

- **Starting and ending flags, with bit stuffing**
  - The new technique allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character.
  - It works like this.
  - Each frame begins and ends with a special bit pattern, 01111110 (in fact, a flag byte).
  - Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream.
  - This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.
  - When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit.
  - Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing.
  - If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110. Figure 3-6 gives an example of bit stuffing.

Figure 3-6. Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

Bit stuffing

(a) The original data.

(b) The data as they appear on the line.

(c) The data as they are stored in receiver's memory after destuffing.

- With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern.
- Thus, if the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data.

- **Physical layer coding violations**
  - **The last method o**f framing is only applicable to networks in which the encoding on the physical medium contains some redundancy.
  - For example, some LANs encode 1 bit of data by using 2 physical bits.
  - Normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair.
  - The scheme means that every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries.
  - The combinations high-high and low-low are not used for data but are used for delimiting frames in some protocols.

- **Numerical Examples**
- **1.The following data fragment occurs in the middle of a data stream for which the byte-stuffing algorithm described in the text is used: A B ESC C ESC FLAG FLAG D. What is the output after stuffing?**
  - **Solution**
  - After stuffing, we get A B ESC ESC C ESC ESC ESC FLAG ESC FLAG D.
- **2. A bit string, 011110111111011111110, needs to be transmitted at the data link layer. What is the string actually transmitted after bit stuffing?**
  - **Solution**
  - **The output is 011110111110011111010**

- **Flow Control:**
- Flow control coordinates that amount of data that can be sent before receiving an acknowledgement.
  - It is one of the most important duties of the data link layer.
  - Flow control tells the sender how much data to send.
  - It makes the sender wait for some sort of an acknowledgement (ACK) before continuing to send more data.
  - Flow Control Techniques:
    - Stop-and-wait, and
    - Sliding Window
- **Error Control**
- Error control in the data link layer is based on ARQ (automatic repeat request), which is the retransmission of data.
  - The term error control refers to methods of error detection and retransmission.
  - Everytime an error is detected in an exchange, specied frames are retransmitted. This process is called ARQ.

Categories of flow control
Stop and wait
Sliding window
Send one frame at a time
Send several frame at a time

- To ensure reliable communication, there needs to exist
  - flow control (managing the amount of data the sender sends), and
  - error control (that data arrives at the destination error free).
  - Flow and error control needs to be done at several layers.
  - For node-to-node links, flow and error control is carried out in the data-link layer.
  - For end-point to end-point, flow and error control is carried out in the transport layer.
  - Flow & Error control techniques:
    - Stop-and-Wait ARQ,
    - Go-Back-N ARQ, and
    - Selective Repeat ARQ

- **Stop and Wait Flow control:**
  - The sender has to wait for an acknowledgment of every frame that it sends.
  - Only when a acknowledgment has been received is the next frame sent.
  - This process continues until the sender transmits an End of Transmission (EOT) frame.
  - In Stop-and-Wait flow control, the receiver indicates its readiness to receive data for each frame.
  - For every frame that is sent, there needs to be an acknowledgment, which takes a similar amount of propagation time to get back to the sender.
  - Only one frame can be in transmission at a time.
  - This leads to ineffciency if propagation delay is much longer than the transmission delay.

Last bit received
End of Transmit of ACK
Receiver
propagation delay
Frame
ACK
Frame
ACK
Sender
transmission delay
Start Transmit
End Transmit
Last bit of ACK received

- **Advantages of Stop and Wait:**
  - It is simple and each frame checked and acknowledged.
- **Disadvantages**
  - Only one frame can be in transmission at a time.
  - It is ineffcient, if the distance between devices is long.
  - Reason is propagation delay is much longer than the transmission delay.
  - The time spent for waiting acknowledgements between each frame can add significant amount to the total transmission time.

- **Advantages of Stop and Wait:**
  - It is simple and each frame checked and acknowledged.
- **Disadvantages**
  - Only one frame can be in transmission at a time.
  - It is ineffcient, if the distance between devices is long.
  - Reason is propagation delay is much longer than the transmission delay.
  - The time spent for waiting acknowledgements between each frame can add significant amount to the total transmission time.

- **Sliding Window Flow Control:**
  - It works by having the sender and receiver have a “window” of frames.
  - Each frame has to be numbered in relation to the sliding window.
  - For a window of size n, frames get a number from 0 to n - 1.
  - Subsequent frames get a number mod n.
  - The sender can send as many frames as would fit into a window.
  - The receiver, upon receiving enough frames, will respond with an acknowledgment of all frames up to a certain point in the window.
  - It is called slide.
  - This window can hold frames at either end and provides the upper limit on the number of frames that can be transmitted before requiring an acknowledgement.

- For example, if n = 8, the frames are numbered 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1...so on.
- The size of the window is (n -1) = 7.
- When the receiver sends an ACK, it includes the number of the next frame it expects to receive.
- When the receiver sends an ACK containing the number 5, it means all frames upto number 4 have been received.
- If the window size is sufficiently large the sender can continuously transmit packets:

- **Stop and Wait ARQ:**
- Include re-transmission of data in case of lost or damaged frames.
- It is addition to the basic flow control mechanism with re-transmissions.
  - Sender sends an information frame to receiver.
  - Sender waits for an ACK before sending the next frame.
  - Receiver sends an ACK if frame is correctly received.
  - If no ACK arrives within time-out, sender will resend the frame.
  - **Time-out period > Rround trip time**
  - If an error is discovered in a data frame, indicating that it has been corrupted in transit, a NAK frame is returned.
  - NAK frames, which are numbered, tell the sender to retransmit the last frame sent.

- **Piggybacking:**
  - In bidirectional communications, both parties send & acknowledge data, i.e. both parties implement flow control.
  - Outstanding ACKs are placed in the header of information frames, piggybacking can save bandwidth since the overhead from a data frame and an ACK frame (addresses, CRC, etc) can be combined into just one frame.

- **Sliding Window ARQ:**
  - To cover retransmission of lost or damaged frames, some features are added to the basic flow control mechanism of sliding window.
  - A Sender may send multiple frames as allowed by the window size.
  - The sending device keeps copies of all transmitted frames, until they have been acknowledged. .
  - In addition to ACK frames, the receiver has the option of returning a NAK frame, if the data have been received damaged.
  - NAK frame tells the sender to retransmit a damaged frame.
  - Here, both ACK and NAK frames must be numbered for identification.
  - ACK frames carry the number of next frame expected.
  - NAK frames on the other hand, carry the number of the damaged frame itself.

- If the last ACK was numbered 3, an ACK 6 acknowledges the receipt of frames 3, 4 and 5 as well.
- If data frames 4 and 5 are received damaged, both NAK 4 and NAK 5 must be returned.
- Like stop and wait ARQ, the sending device in sliding window ARQ is equipped with a timer to enable it to handle lost acknowledgements.
- Sliding window ARQ is two types: **Go-back-n ARQ, and Selective Reject ARQ.**
- There are two ACK processing methods in sliding windows:
  - **Selective ACK:** The ACK N message acknowledges only the frame with sequence number N.
  - **Cumulative ACK :** The ACK N message acknowledges all frames with sequence number <= N.

- **Go-back-n ARQ:**
  - **The sliding window method using cumulative ACK is known as the Go-Back-N ARQ protocol.**
  - Receiver window size is 1.
  - In this method, if one frame is lost or damaged all frames sent, since the last frame acknowledged are retransmitted.
  - For example, sender may send frames 1,2,3,4 and get an NAK with a value of 2.
  - The NAK acknowledges everything that came before it, and asks for frame 2 (and subsequent frames) to be resent.
  - NAK number refer to the next expected frame number.
  - Example: In the following figure, frame 2 has an error, then all subsequent frames are discarded.
  - After timeout sender sends all frames from frame 2.

Timeout interval
0 1 2 3 4 5 6 7 8 2 3 4 5 6 7 8 9 10
Ack 0
Ack 1
Ack 2
Ack 3
Ack 4
Ack 5
Ack 6
Ack 7
0 1 E D D D D D D 2 3 4 5 6 7 8
Error
Msg discarded by data link layer
Time

- **Damaged/Error Frame :**
  - In go-back-n ARQ, The receiver sends the NAK for this frame along with that frame number, that it expects to be retransmitted.
  - After sending NAK, the receiver discards all the frames that it receives, after a damaged frame.
  - The receiver does not send any ACK (acknowledgement) for the discarded frames.
  - After the sender receives the NAK for the damaged frame, it retransmits all the frames onwards the frame number referred by NAK.

- **Lost frame:**
  - In go-back-n ARQ, Receiver easily detects the loss of a frame as the newly received frame is received out of sequence.
  - The receiver sends the NAK for the lost frame and then the receiver discards all the frames received after a lost frame.
  - The receiver does not send any ACK for that discarded frames.
  - After the sender receives the NAK for the lost frame, it retransmits the lost frame referred by NAK and also retransmits all the frames which it has sent after the lost frame.

- **Lost Acknowledgement :**
  - In go-back-n ARQ, If the sender does not receive any ACK or if the ACK is lost or damaged in between the transmission.
  - The sender waits for the time to run out and as the time run outs, the sender retransmits all the frames for which it has not received the ACK.
  - The sender identifies the loss of ACK with the help of a timer.
  - The ACK number, like NAK number, shows the number of the frame, that receiver expects to be the next in sequence.
  - The window size of the receiver is 1 as the data link layer only require the frame which it has to send next to the network layer.
  - The sender window size is equal to ‘w’.
  - If the error rate is high, a lot of bandwidth is lost wasted.

- **Selective Repeat ARQ:**
  - Selective Repeat ARQ overcomes the limitations of Go-Back-N by adding two new features:
    - Receiver window > 1 frame: Out-of-order but error-free frames can be accepted
    - Retransmission mechanism is modified: Only individual frames are retransmitted
  - In this method, only specific damaged or lost frame is retransmitted
  - Sender only retransmits frames for which a NAK is received.
  - NAK number refer to the frame lost.
  - If a frame is corrupted in transmit, a NAK is returned and the frame is resent out of sequence.
  - The sender needs to maintain all data that hasn't been aknowledged yet.

- The receiving device must be able to sort the frames it has and insert the retransmitted frame into its proper place in the sequence.
- It has advantage that few re-transmissions than go-back-n.
- But complexity at sender and receiver is involved.
- Example: Frame 2 has an error, so receiver maintains buffer to store the next frames.

Sender
Timeout interval
0 1 2 3 4 5 6 7 8 2 9 10 11 12 13 14 15 0
Ack 0
Ack 1
Ack 3
Ack 2
0 1 E 3 4 5 6 7 8 2 9 10 11 12 13 14
Receiver
Error
Buffered by data link layer
Messages 2-8 passed to network layer
Time

- **Damaged frames :**
  - In Selective repeat, If a receiver receives a damaged frame, it sends the NAK for the frame in which error or damage is detected.
  - The NAK number, like in go-back-n also indicate the acknowledgement of the previously received frames and error in the current frame.
  - The receiver keeps receiving the new frames while waiting for the damaged frame to be replaced.
  - The frames that are received after the damaged frame are not be acknowledged until the damaged frame has been replaced.

- **Lost Frame :**
  - As in a selective repeat protocol, a frame can be received out of order and further they are sorted to maintain a proper sequence of the frames.
  - While sorting, if a frame number is skipped, the receiver recognise that a frame is lost and it sends NAK for that frame to the sender.
  - After receiving NAK for the lost frame the sender searches that frame in its window and retransmits that frame.
  - If the last transmitted frame is lost then receiver does not respond and this silence is a negative acknowledgement for the sender.

- **Lost Acknowledgement :**
  - In Selective repeat, If the sender does not receive any ACK or the ACK is lost or damaged in between the transmission.
  - The sender waits for the time to run out and as the time run outs, the sender retransmit all the frames for which it has not received the ACK.
  - The sender identifies the loss of ACK with the help of a timer.

- **Elementary Data Link Protocols**
- Elementary Data Link protocols are classified into three categories, as given below,
  - Unrestricted simplex protocol
  - Simplex stop and wait protocol
  - Simplex protocol for noisy channels.
- **Unrestricted Simplex Protocol**
  - Data transmitting is carried out in one direction only.
  - The transmission (Tx) and receiving (Rx) are always ready and the processing time can be ignored.
  - In this protocol, infinite buffer space is available, and no errors are occurring that is no damage frames and no lost frames.
  - The Unrestricted Simplex Protocol is diagrammatically represented as follows –

Sender
Receiver
Sender
Receiver
A
B
Request
Frame
Arrival
Request
Frame
Arrival
Request
Frame
Arrival
Time
Time
Time
Time

- **Simplex Stop and Wait protocol**
- In this protocol we assume that data is transmitted in one direction only.
- No error occurs; the receiver can only process the received information at finite rate.
- These assumptions imply that the transmitter cannot send frames at rate faster than the receiver can process them.
- The main problem here is how to prevent the sender from flooding the receiver.
- The general solution for this problem is to have the receiver send some sort of feedback to sender, the process is as follows –

- **Simplex Stop and Wait protocol**
- **Step1 –** The receiver send the acknowledgement frame back to the sender telling the sender that the last received frame has been processed and passed to the host.
- **Step 2 –** Permission to send the next frame is granted.
- **Step 3 –** The sender after sending the sent frame has to wait for an acknowledge frame from the receiver before sending another frame.
- This protocol is called Simplex Stop and wait protocol, the sender sends one frame and waits for feedback from the receiver.
- When the ACK arrives, the sender sends the next frame.
- The Simplex Stop and Wait Protocol is diagrammatically represented as follows –

Sender
Receiver
A
B
Request
Frame
Arrival
ACK
Arrival
Request
Frame
ACK
Arrival
Arrival
Time
Time

- **Simplex Protocol for Noisy Channel**
- Data transfer is only in one direction, consider separate sender and receiver, finite processing capacity and speed at the receiver, since it is a noisy channel, errors in data frames or acknowledgement frames are expected.
- Every frame has a unique sequence number.
- After a frame has been transmitted, the timer is started for a finite time.
- Before the timer expires, if the acknowledgement is not received , the frame gets retransmitted, when the acknowledgement gets corrupted or sent data frames gets damaged, how long the sender should wait to transmit the next frame is infinite.
- The Simplex Protocol for Noisy Channel is diagrammatically represented as follows –

Sender
Receiver
A
B
Request
Frame 1
Arrival
ACK 1
Arrival
Frame 2
Frame 2
Lost
Request
Frame 2
ACK 2
Arrival
Arrival
Time
Time

- **HDLC**
- High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links.
- It implements the Stop-and-Wait protocol.
- Although this protocol is more a theoretical issue than practical, most of the concept defined in this protocol is the basis for other practical protocols such as PPP.

- **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 mul- tipoint 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.
- I-frames are used to data-link user data and control information relating to user data (piggy- backing).
- 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 ink itself.

- **Framing**
- Each frame in HDLC may contain up to six fields, as shown in Figure 11.16: 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*

- **Framing**
- **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.

- **Framing**
- **Control field.** The control field is one or two bytes used for flow and error control.
- **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.
- The format is specific for the type of frame, as shown in Figure 11.17.

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.

- **Control Field for I-Frames**
- 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.
- This kind of frame acknowledges the receipt of a safe and sound frame or group of frames.
- In this case, the value of the N(R) field defines the acknowledgment number.
- **Receive not ready (RNR).** If the value of the code subfield is 10, it is an RNR S-frame.
- This kind of frame is an RR frame with additional functions. It acknowledges the receipt of a frame or group of frames, and it announces that the receiver is busy and cannot receive more frames.

- **Reject (REJ).** If the value of the code subfield is 01, it is an REJ S-frame.
- This is a NAK frame, but not like the one used for Selective Repeat ARQ.
- It is a NAK that can be used in Go-Back-N ARQ to improve the efficiency of the process by informing the sender, before the sender timer expires, that the last frame is lost or damaged.
- The value of N(R) is the negative acknowledgment number.
- **Selective reject (SREJ).** If the value of the code subfield is 11, it is an SREJ S-frame.
- This is a NAK frame used in Selective Repeat ARQ. Note that the HDLC Protocol uses the term selective reject instead of selective repeat.

- **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).
- Today, millions of Internet users who need to connect their home computers to the server of an Internet service provider use PPP.
- The majority of these users have a traditional modem; they are connected to the Internet through a telephone line, which provides the services of the physical layer.
- But to control and manage the transfer of data, there is a need for a point-to-point protocol at the data-link layer.
- PPP is by far the most common.

- **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.

- **Point-to-Point Protocol (PPP)**
- **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 shows the format of a PPP frame. The description of each field follows:

Figure 11.20 *PPP frame format*

- **Point-to-Point Protocol (PPP)**
- **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).
- **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.

- **Point-to-Point Protocol (PPP)**
- **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.

- **Point-to-Point Protocol (PPP)**
- **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 flaglike 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.

- **Point-to-Point Protocol (PPP)**
- **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.

- **Point-to-Point Protocol (PPP)**
- 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.

- **Point-to-Point Protocol (PPP)**
- 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, as shown in Figure 11.22.
- Note that there are one LCP, two APs, and several NCPs.
- Data may also come from several different network layers.

Figure 11.22 *Multiplexing in PPP*

Legend

LCP : Link control protocol
AP: Authentication protocol
NCP: Network control protocol

Protocol values:

LCP: 0xC021
AP: 0xC023 and 0xC223
NCP: 0x8021 and ....
Data: 0x0021 and ....

- **Link Control Protocol**
- The Link Control Protocol (LCP) is responsible for establishing, maintaining, configuring, and terminating links.
- It also provides negotiation mechanisms to set options between the two endpoints.
- Both endpoints of the link must reach an agreement about the options before the link can be established. See Figure 11.21.
- All LCP packets are carried in the payload field of the PPP frame with the protocol field set to C021 in hexadecimal (see Figure 11.23).

Figure 11.23 *LCP packet encapsulated in a frame*

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

- There are three categories of packets.
- The first category, comprising the first four **packet types**, is used for link configuration during the establish phase.
- The second category, comprising packet types 5 and 6, is used for link termination during the termination phase.
- The last five packets are used for link monitoring and debugging.
- 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.

- There are many options that can be negotiated between the two endpoints.
- Options are inserted in the information field of the configuration packets.
- In this case, the information field is divided into three fields: option type, option length, and option data.
- We list some of the most common options in Table 11.2.

Table 11.2 *Common options*

Option	*Default*
Maximum receive unit (payload field size)	1500
Authentication protocol	None
Protocol field compression	Off
Address and control field compression	Off

- **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.**
- Note that these protocols are used during the authentication phase.

- **PAP**
- The Password Authentication Protocol (PAP) is a simple authentication procedure with a two-step process:
  - a. The user who wants to access a system sends an authentication identification (usually the user name) and a password.
  - b. The system checks the validity of the identification and password and either accepts or denies connection.

- Figure 11.24 shows the three types of packets used by PAP and how they are actually exchanged.
- When a PPP frame is carrying any PAP packets, the value of the protocol field is 0xC023.
- The three PAP packets are authenticate-request, authenticate-ack, and authenticate-nak.
- The **first packet** is used by the user to send the user name and password.
- The **second** is used by the system to allow access.
- The **third** is used by the system to deny access.

Figure 11.24 *PAP packets encapsulated in a PPP frame*

- **CHAP**
- The **Challenge Handshake Authentication Protocol (CHAP)** is a three-way hand-shaking authentication protocol that provides greater security than PAP.
- In this method, the password is kept secret; it is never sent online.
  - a. The system sends the user a challenge packet containing a challenge value, usually a few bytes.
  - b. 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 shows the packets and how they are used.

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.

- **Network Control Protocols**
- PPP is a multiple-network-layer protocol.
- It can carry a network-layer data packet from protocols defined by the Internet, OSI, Xerox, DECnet, AppleTalk, Novel, and so on.
- To do this, PPP has defined a specific Network Control Protocol for each network protocol.
- For example, IPCP (Internet Protocol Control Protocol) configures the link for carrying IP data packets.
- Xerox CP does the same for the Xerox protocol data packets, and so on.
- Note that none of the NCP packets carry network-layer data; they just configure the link at the network layer for the incoming data.

- **Network Control Protocols**
- **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.
- The format of an IPCP packet is shown in Figure 11.26. Note that the value of the protocol field in hexadecimal is 8021.
- IPCP defines seven packets, distinguished by their code values, as shown in Table 11.3.

Figure 11.26 *IPCP packet encapsulated in PPP frame*

IPCP defines seven packets, distinguished by their code values, as shown in Table 11.3.

Table 11.3 *Code value for IPCP packets*

Code	*IPCP Packet*
0x01	Configure-request
0x02	Configure-ack
0x03	Configure-nak
0x04	Configure-reject
0x05	Terminate-request
0x06	Terminate-ack
0x07	Code-reject

- **Other Protocols**
- There are other NCP protocols for other network-layer protocols.
- The OSI Network Layer Control Protocol has a protocol field value of 8023; the Xerox NS IDP Control Protocol has a protocol field value of 8025; and so on.
- **Data from the Network Layer**
- After the network-layer configuration is completed by one of the NCP protocols, the users can exchange data packets from the network layer.
- Here again, there are different protocol fields for different network layers. Figure 11.27 shows the frame for IP.

Figure 11.27 *IP datagram encapsulated in a PPP frame*

- **Multilink PPP**
- PPP was originally designed for a single-channel point-to-point physical link.
- The availability of multiple channels in a single point-to-point link motivated the development of Multilink PPP.
- In this case, a logical PPP frame is divided into several actual PPP frames.
- A segment of the logical frame is carried in the payload of an actual PPP frame, as shown in Figure 11.28.
- To show that the actual PPP frame is carrying a frag- ment of a logical PPP frame, the protocol field is set to $(003d)_{16}$.
- This new development adds complexity.

Figure 11.28 *Multilink PPP*

- **Error Control Techniques:**
  - Many factors including line noise can alter or wipe out one or more bits of a given data unit.
  - Reliable systems must have mechanism for detecting and correcting such errors.
  - Error detection and correction are implemented either at the data link layer or the transport layer of the OSI model.

Types of Errors

- Single bit (only one bit in the data unit has changed)
- Burst (2 or more bit in the data unit has changed)

Errors classification

Error correction

- **Error Detection**
  - Error detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination.
  - Checking function performs the action that the received bit stream passes the checking criteria, the data portion of the data unit is accepted else rejected.
- **Vertical Redundancy Check (VRC)**
  - In this technique, a redundant bit, called parity bit, is appended to every data unit, so that the total number of 1's in the unit (including the parity bit) becomes even.
  - If number of 1's are already even in data, then parity bit will be 0.
  - Some systems may use odd parity checking, where the number of 1's should be odd.
  - The principle is the same, the calculation is different.

How error detection works with redundancy concept

Detection of error using vertical redundancy check method

- **ERROR DETECTION TECHNIQUES**
  - Basic approach used for error detection is the use of redundancy, where additional bits are added to facilitate detection and correction of errors.
  - Popular techniques are
    - Simple Parity check
    - Two-dimensional Parity check
    - Checksum
    - Cyclic redundancy check
  - **Redundancy** is the method in which some extra bits are added to the data so as to check whether the data contain error or not.
    - m - data bits (i.e., message bits)
    - r - redundant bits (or check bits).
    - n - total number of bits
    - n= (m + r).
    - An n-bit unit containing data and check-bits is often referred to as an n-bit codeword.

- **SIMPLE PARITY CHECK**
  - The simplest and most popular error detection scheme.
  - Appends a Parity bit to the end of the data.
  - **Even Parity:** 01000001 Number of ones in the group of bits is even.
  - **Odd Parity:** 11000001 - Number of ones in the group of bits is odd.
  - A parity of 1 is added to the block if it contains an odd number of 1's (ON bits) and 0 is added if it contains an even number of 1's.
  - At the receiving end the parity bit is computed from the received data bits and compared with the received parity bit.
  - This scheme makes the total number of 1's even, that is why it is called even parity checking.
  - Considering a 4-bit word, different combinations of the data words and the corresponding code words are given in Table 3.2.1.

Sender
Data
1011011
Compute parity bit
1011011
1
Transmission Media
Receiver
Accept Data
Y
Even
N
Reject Data
Compute parity bit
1011011
1

Decimal value	Data Block	Parity bit	Code word
0	0000	0	00000
1	0001	1	00011
2	0010	1	00101
3	0011	0	00110
4	0100	1	01001
5	0101	0	01010
6	0110	0	01100
7	0111	1	01111
8	1000	1	10001
9	1001	0	10010
10	1010	0	10100
11	1011	1	10111
12	1100	0	11000
13	1101	1	11011
14	1110	1	11101
15	1111	0	11110

Example:

Sender
Receiver
1 1 0 0 1 1 1
Accept data
Reject data
Even parity
Odd parity
Even Parity Generator
(counts 1 in data)
1
Parity checker
Counts 1 in data + VRC)
1 1 0 0 1 1 1 1
1 1 0 0 1 1 1 1

- **PERFORMANCE OF SIMPLE PARITY CHECK**
  - Simple parity check can detect all single-bit errors.
  - It can also detect burst error, if the number of bits in even or odd.
  - The technique is not foolproof against burst errors that invert more than one bit.
  - If an even number of bits is inverted due to error, the error is not detected.

- **Checksum**
  - There are two algorithms involved in this process, checksum generator at sender end and checksum checker at receiver end.
  - The sender follows these steps
    - The data unit is divided into k sections each of n bits.
    - All sections are added together using 1's complement to get the sum.
    - The sum is complemented and becomes the checksum.
    - The checksum is sent with the data.
  - The receiver follows these steps
    - The received unit is divided into k sections each of n bits.
    - All sections are added together using 1's complement to get the sum.
    - The sum is complemented.
    - If the result is zero, the data are accepted, otherwise they are rejected.

- **Limitation of checksum:**
  - It is not possible to detect the vertical error from the data which is received at receivers end.
  - If noise modify the data in such a way that vertically placed bits can cancel the change made to them then calculated checksum will always be same as received checksum.
  - Such errors cannot be detected and they are known as vertical errors.

Example 1:

Sender

10101001	subunit1
00111001	subunit 2
11100010	sum
00011101	Complement of sum

10101001	00111001	00011101
Data		checksum

Reciever:

10101001	subunit1
00111001	subunit2
00011101	Checksum
11111111	sum
00000000	complement

Conclusion = Accept data.

- **Cyclic Redundancy Check (CRC):**
  - One of the most powerful and commonly used error detecting codes.
  - **Basic approach:**
  - Given a m-bit block of bit sequence, the sender generates an n-bit sequence known as frame sequence check(FCS), so that the resulting frame, consisting of m+n bits exactly divisible by same predetermined number.
  - The receiver divides the incoming frame by that number and, if there is no reminder, assumes there was no error.

- **CRC Generator-**
  - CRC generator is an algebraic polynomial represented as a bit pattern.
  - Bit pattern is obtained from the CRC generator using the following rule-
    - The power of each term gives the position of the bit and the coefficient gives the value of the bit.
    - Example-
    - Consider the CRC generator is x7 + x6 + x4 + x3 + x + 1.
    - The corresponding binary pattern is obtained as-

$$1x^7 + 1x^6 + 0x^5 + 1x^4 + 1x^3 + 0x^2 + 1x^1 + 1x^0$$

$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$

$$1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1$$

Thus, for the given CRC generator, the corresponding binary pattern is 11011011.

- **Properties Of CRC Generator-**
  - The algebraic polynomial chosen as a CRC generator should have at least the following properties-
  - **Rule-01:**
  - It should not be divisible by x.
  - This condition guarantees that all the burst errors of length equal to the length of polynomial are detected.
  - **Rule-02:**
  - It should be divisible by x+1.
  - This condition guarantees that all the burst errors affecting an odd number of bits are detected.

- **Important Notes-**
- If the CRC generator is chosen according to the above rules, then-
  - CRC can detect all single-bit errors
  - CRC can detect all double-bit errors provided the divisor contains at least three logic 1's.
  - CRC can detect any odd number of errors provided the divisor is a factor of x+1.
  - CRC can detect all burst error of length less than the degree of the polynomial.
  - CRC can detect most of the larger burst errors with a high probability.
  - **Steps Involved-**

- **Step-01: Calculation Of CRC At Sender Side-**
- At sender side,
  - A string of n 0's is appended to the data unit to be transmitted.
  - Here, n is one less than the number of bits in CRC generator.
  - Binary division is performed of the resultant string with the CRC generator.
  - After division, the remainder so obtained is called as CRC.
  - It may be noted that CRC also consists of n bits.
- **Step-02: Appending CRC To Data Unit-**
- At sender side,
  - The CRC is obtained after the binary division.
  - The string of n 0's appended to the data unit earlier is replaced by the CRC remainder.

- **Step-03: Transmission To Receiver-**
  - The newly formed code word (Original data + CRC) is transmitted to the receiver.
- **Step-04: Checking at Receiver Side-**
- At receiver side,
  - The transmitted code word is received.
  - The received code word is divided with the same CRC generator.
  - On division, the remainder so obtained is checked.
- **The following two cases are possible-**
- **Case-01: Remainder = 0**
  - If the remainder is zero,
  - Receiver assumes that no error occurred in the data during the transmission.
  - Receiver accepts the data.

- Case-02: Remainder $\neq 0$
  - If the remainder is non-zero,
  - Receiver assumes that some error occurred in the data during the transmission.
  - Receiver rejects the data and asks the sender for retransmission.

Sender
m
n
Data
00....0
(n+1)bits
Divisor
n bits
CRC
Data
CRC
Receiver
m
n
Data
CRC
(n+1)bits
Divisor
Reminder
N
Y
Reject
Zero
Accept

- Fig. 3.2.7 by dividing a sample 4- bit number by the coefficient of the generator polynomial x3+x+1, which is 1011, using the modulo-2 arithmetic.
- Modulo-2 arithmetic is a binary addition process without any carry over, which is just the Exclusive-OR operation.
- Consider the case where k=1101. Hence we have to divide 1101000 (i.e. k appended by 3 zeros) by 1011, which produces the remainder r=001, so that the bit frame (k+r) =1101001 is actually being transmitted through the communication channel.
- At the receiving end, if the received number, i.e.,1101001 is divided by the same generator polynomial 1011 to get the remainder as 000, it can be assumed that the data is free of errors.

Sender: Sender transmit the data along with remainder (**CRC**)

```
        1111   Quotient                    Data: 1101
1011 | 1101000   <------ K                 Divisor: 1011
       ---------
       1011
        1100
        1011
         1110
         1011
          1010
          1011
            001   Reminder (r)
```

Data to be sent: **1 1 0 1 0 0 1**

Data | CRC

Receiver:

```
        1111   Quotient              Data: 1101001
1011 | 1101001                       Divisor: 1011
       |
       1011
       ----
        1100
        1011
        ----
         1110
         1011
         ----
          1011
          1011
          ----
           000   Reminder
```

Note: Remainder is zero, no error. Receiver can accept the data.

Performance of CRC

- CRC can detect all single-bit errors.
- CRC can detect all double-bit errors(three1's)
- CRC can detect any odd number of errors of less than the degree of the polynomial.
- CRC detects most of the larger burst errors with a high probability.

- **Problem-01:**
  - A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x4+x+1. What is the actual bit string transmitted?
  - **Solution-**
  - The generator polynomial G(x) = x4 + x + 1 is encoded as 10011.
  - Clearly, the generator polynomial consists of 5 bits.
  - So, a string of 4 zeroes is appended to the bit stream to be transmitted.
  - The resulting bit stream is 11010110110000.
  - Now, the binary division is performed as-

```
                 1 1 0 0 0 0 1 0 1 0
10011 | 1 1 0 1 0 1 1 0 1 1 0 0 0 0
        1 0 0 1 1
          1 0 0 1 1
          1 0 0 1 1
            0 0 0 0 1
            0 0 0 0 0
              0 0 0 1 0
              0 0 0 0 0
                0 0 1 0 1
                0 0 0 0 0
                  0 1 0 1 1
                  0 0 0 0 0
                    1 0 1 1 0
                    1 0 0 1 1
                      0 1 0 1 0
                      0 0 0 0 0
                        1 0 1 0 0
                        1 0 0 1 1
                          0 1 1 1 0
                          0 0 0 0 0
                            1 1 1 0   <- Remainder
```

- From here, CRC = 1110.
- Now,
- The code word to be transmitted is obtained by replacing the last 4 zeroes of 11010110110000 with the CRC.
- Thus, the code word transmitted to the receiver = 11010110111110.

- **Problem-02:**
  - A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is x3+1.
  - What is the actual bit string transmitted?
  - Suppose the third bit from the left is inverted during transmission. How will receiver detect this error?

- **Solution-**
- **Part-01:**
  - The generator polynomial G(x) = x3 + 1 is encoded as 1001.
  - Clearly, the generator polynomial consists of 4 bits.
  - So, a string of 3 zeroes is appended to the bit stream to be transmitted.
  - The resulting bit stream is 10011101000.
  - Now, the binary division is performed as-

```
                 10001100
1001 | 10011101000
       1001
       ----
       00001
        0000
        ----
        00011
         0000
         ----
         00110
          0000
          -----
          01101
           1001
           -----
           01000
            1001
            -----
            00010
             0000
             -----
             00100
              0000
              -----
              0100
```

CRC

- From here, CRC = 100.
- Now,
- The code word to be transmitted is obtained by replacing the last 3 zeroes of 10011101000 with the CRC.
- Thus, the code word transmitted to the receiver = 10011101100.
- **Part-02:**
  - According to the question,
  - Third bit from the left gets inverted during transmission.
  - So, the bit stream received by the receiver = 10111101100.

  - Now,
  - Receiver receives the bit stream = 10111101100.
  - Receiver performs the binary division with the same generator polynomial as-

```
                 1 0 1 0 1 0 0 0
1 0 0 1 | 1 0 1 1 1 1 0 1 1 0 0
          1 0 0 1
          ---------
          0 0 1 0 1
            0 0 0 0
            ---------
            0 1 0 1 1
              1 0 0 1
              ---------
              0 0 1 0 0
                0 0 0 0
                ---------
                0 1 0 0 1
                  1 0 0 1
                  ---------
                  0 0 0 0 1
                    0 0 0 0
                    ---------
                    0 0 0 1 0
                      0 0 0 0
                      ---------
                      0 0 1 0 0
                        0 0 0 0
                        ---------
                        0 1 0 0   <---  Remainder
                        ---------
```

- From here,
  - The remainder obtained on division is a non-zero value.
  - This indicates to the receiver that an error occurred in the data during the transmission.
  - Therefore, receiver rejects the data and asks the sender for retransmission.

- From here,
  - The remainder obtained on division is a non-zero value.
  - This indicates to the receiver that an error occurred in the data during the transmission.
  - Therefore, receiver rejects the data and asks the sender for retransmission.

- **Error-Correcting Codes**
- We will examine four different error-correcting codes:
  - 1. Hamming codes.
  - 2. Binary convolutional codes.
  - 3. Reed-Solomon codes.
  - 4. Low-Density Parity Check codes.
  - All of these codes add redundancy to the information that is sent.
  - A frame consists of m data (i.e., message) bits and r redundant (i.e. check) bits.
  - In a **block code**, the r check bits are computed solely as a function of the m data bits with which they are associated, as though the m bits were looked up in a large table to find their corresponding r check bits.
  - In a **systematic code,** the m data bits are sent directly, along with the check bits, rather than being encoded themselves before they are sent.

- In a **linear code**, the r check bits are computed as a linear function of the m data bits.
- Exclusive OR (XOR) or modulo 2 addition is a popular choice.
- This means that encoding can be done with operations such as matrix multiplications or simple logic circuits.
- The codes we will look at in this section are linear, systematic block codes unless otherwise noted.

- Let the total length of a block be n (i.e., $n = m + r$).
- We will describe this as an (n,m) code.
- An n-bit unit containing data and check bits is referred to as an n-bit codeword.
- The **code rate, or simply rate,** is the fraction of the codeword that carries information that is not redundant, or m/n.
- The rates used in practice vary widely.

- They might be 1/2 for a noisy channel, in which case half of the received information is redundant, or close to 1 for a high-quality channel, with only a small number of check bits added to a large message.

- To understand how errors can be handled, it is necessary to first look closely at what an error really is.
- Given any two codewords that may be transmitted or received—say, 10001001 and 10110001—it is possible to determine how many corresponding bits differ.
- In this case, 3 bits differ.
- To determine how many bits differ, just XOR the two codewords and count the number of 1 bits in the result.
- 10001001
- 10110001
- 00111000

- The number of bit positions in which two codewords differ is called the **Hamming distance (Hamming, 1950)**.
- Its significance is that if two codewords are a Hamming distance d apart, it will require d single-bit errors to convert one into the other.

- **Hamming Codes**
  - Hamming code is a set of error-correction codes that can be used to detect and correct the errors that can occur when the data is moved or stored from the sender to the receiver.
  - It is technique developed by R.W. Hamming for error correction.
  - **Redundant bits** –
  - Redundant bits are extra binary bits that are generated and added to the information-carrying bits of data transfer to ensure that no bits were lost during the data transfer.
  - The number of redundant bits can be calculated using the following formula:
  - $2^r \geq m + r + 1$, where, r = redundant bit, m = data bit
  - Suppose the number of data bits is 7, then the number of redundant bits can be calculated using:
  - $= 2^4 \geq 7 + 4 + 1$
  - Thus, the number of redundant bits= 4

- **Parity bits –**
  - A parity bit is a bit appended to a data of binary bits to ensure that the total number of 1's in the data is even or odd. Parity bits are used for error detection. There are two types of parity bits:
- **Even parity bit:**
  - In the case of even parity, for a given set of bits, the number of 1's are counted. If that count is odd, the parity bit value is set to 1, making the total count of occurrences of 1's an even number. If the total number of 1's in a given set of bits is already even, the parity bit's value is 0.
- **Odd Parity bit –**
  - In the case of odd parity, for a given set of bits, the number of 1's are counted. If that count is even, the parity bit value is set to 1, making the total count of occurrences of 1's an odd number. If the total number of 1's in a given set of bits is already odd, the parity bit's value is 0.

- **General Algorithm of Hamming code –**
  - The Hamming Code is simply the use of extra parity bits to allow the identification of an error.
  - Write the bit positions starting from 1 in binary form (1, 10, 11, 100, etc).
  - All the bit positions that are a power of 2 are marked as parity bits (1, 2, 4, 8, etc).
  - All the other bit positions are marked as data bits.
  - Each data bit is included in a unique set of parity bits, as determined its bit position in binary form.
  - a. Parity bit 1 covers all the bits positions whose binary representation includes a 1 in the least significant
  - position (1, 3, 5, 7, 9, 11, etc).
  - b. Parity bit 2 covers all the bits positions whose binary representation includes a 1 in the second position from the least significant bit (2, 3, 6, 7, 10, 11, etc).

- c. Parity bit 4 covers all the bits positions whose binary representation includes a 1 in the third position from the least significant bit (4–7, 12–15, 20–23, etc).
- d. Parity bit 8 covers all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit bits (8–15, 24–31, 40–47, etc).
- e. In general, each parity bit covers all bits where the bitwise AND of the parity position and the bit position is non-zero.
- Since we check for even parity set a parity bit to 1 if the total number of ones in the positions it checks is odd.
- Set a parity bit to 0 if the total number of ones in the positions it checks is even.

- **Determining the position of redundant bits –**
  - These redundancy bits are placed at the positions which correspond to the power of 2.
  - As in the above example:
  - The number of data bits = 7
  - The number of redundant bits = 4
  - The total number of bits = 11
  - The redundant bits are placed at positions corresponding to power of 2- 1, 2, 4, and 8

- Suppose the data to be transmitted is 1011001, the bits will be placed as follows:

11	10	9	8	7	6	5	4	3	2	1
1	0	1	R8	1	0	0	R4	1	R2	R1

- **Determining the Parity bits –**
  - R1 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the least significant position.
  - R1: bits 1, 3, 5, 7, 9, 11

- To find the redundant bit R1, we check for even parity.
- Since the total number of 1's in all the bit positions corresponding to R1 is an even number the value of R1 (parity bit's value) = 0

- R2 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the second position from the least significant bit.
- R2: bits 2,3,6,7,10,11

- To find the redundant bit R2, we check for even parity.
- Since the total number of 1's in all the bit positions corresponding to R2 is odd the value of R2(parity bit's value)=1

- R4 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the third position from the least significant bit.
- R4: bits 4, 5, 6, 7

- To find the redundant bit R4, we check for even parity.
- Since the total number of 1's in all the bit positions corresponding to R4 is odd the value of R4(parity bit's value) = 1

- R8 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit.
- R8: bit 8,9,10,11

- To find the redundant bit R8, we check for even parity.
- Since the total number of 1's in all the bit positions corresponding to R8 is an even number the value of R8(parity bit's value)=0.

- Thus, the data transferred is:

11	10	9	8	7	6	5	4	3	2	1
1	0	1	0	1	0	0	1	1	1	0

- **Error detection and correction –**
  - Suppose in the above example the 6th bit is changed from 0 to 1 during data transmission, then it gives new parity values in the binary number:

- The bits give the binary number as 0110 whose decimal representation is 6.
- Thus, the bit 6 contains an error. To correct the error the 6th bit is changed from 1 to 0.

- **Medium Access Control Sublayer**

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.
  - This decision depends on the state of the medium (idle or busy).
  - In other words, each station can transmit when it desires on the condition that it follows the predefined procedure, including testing the state of the medium.
  - 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.

- **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.
  - In a random-access method, each station has the right to the medium without being controlled by any other station.
  - However, if more than one station tries to send, there is an access conflict—collision—and the frames will be either destroyed or modified.

- **ALOHA**
  - ALOHA, the earliest random access method, was developed at the University of Hawaii in early 1970.
  - It was designed for a radio (wireless) LAN, but it can be used on any shared medium.
  - It is obvious that there are potential collisions in this arrangement.
  - The medium is shared between the stations.
  - When a station sends data, another station may attempt to do so at the same time.
  - The data from the two stations collide and become garbled.

- **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 shows an example of frame collisions in pure ALOHA.

Figure 12.2 *Frames in a pure ALOHA network*

- There are four stations (unrealistic assumption) that contend with one another for access to the shared channel.
- The figure shows that each station sends two frames; there are a total of eight frames on the shared medium.
- Some of these frames collide because multiple frames are in contention for the shared channel.
- Figure 12.2 shows that only two frames survive: one frame from station 1 and one frame from station 3.
- We need to mention that even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed.
- It is obvious that we need to resend the frames that have been destroyed during transmission.

- The pure ALOHA protocol relies on acknowledgments from the receiver.
- When a station sends a frame, it expects the receiver to send an acknowledgment.
- If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.
- 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.

- We call this time the backoff time TB .
- Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames.
- After a maximum number of retransmission attempts K max , a station must give up and try later.
- Figure 12.3 shows the procedure for pure ALOHA based on the above strategy

Figure 12.3 *Procedure for pure ALOHA protocol*

- The time-out period is equal to the maximum possible round-trip propagation delay, which is twice the amount of time required to send a frame between the two most widely separated stations ($2 \times T_p$).
- The backoff time T_B is a random value that normally depends on K (the number of attempted unsuccessful transmissions).
- The formula for T_B depends on the implementation.
- One common formula is the binary exponential backoff. In this method, for each retransmission, a multiplier R = 0 to $2^K - 1$ is randomly chosen and multiplied by Tp (maximum propagation time) or Tfr (the average time required to send out a frame) to find T_B.

- Note that in this procedure, the range of the random numbers increases
- after each collision. The value of K max is usually chosen as 15.

- **Vulnerable time**
- Let us find the vulnerable time, the length of time in which there is a possibility of collision.
- We assume that the stations send fixed-length frames with each frame taking Tfr seconds to send.
- Figure 12.4 shows the vulnerable time for station B.

Figure 12.4 *Vulnerable time for pure ALOHA protocol*

- **Vulnerable time**
- Station B starts to send a frame at time t.
- Now imagine station A has started to send its frame after t – T fr .
- This leads to a collision between the frames from station B and station A.
- On the other hand, suppose that station C starts to send a frame before time t + Tfr .
- Here, there is also a collision between frames from station B and station C.
- Looking at Figure 12.4, we see that the vulnerable time during which a collision may occur in pure ALOHA is 2 times the frame transmission time.
- Pure ALOHA vulnerable time=2 *Tfr

- **Example 12.2**
- 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 × 1 ms =2 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.

- **Throughput**
- Let us call G the average number of frames generated by the system during one frame transmission time.
- Then it can be proven that the average number of successfully transmitted frames for pure ALOHA is $S = G \times e^{-2G}$.
- The maximum throughput S_{max} is 0.184, for $G = 1/2$.
- In other words, if one-half a frame is generated during one frame transmission time (one frame during two frame transmission times), then 18.4 percent of these frames reach their destination successfully.
- We expect $G = 1/2$ to produce the maximum throughput because the vulnerable time is 2 times the frame transmission time.
- Therefore, if a station generates only one frame in this vulnerable time (and no other stations generate a frame during this time), the frame will reach its destination successfully.

- **Throughput**

The throughput for pure ALOHA is $S = G \times e^{-2G}$.

The maximum throughput $S_{max} = 1/(2e) = 0.184$ when $G = (1/2)$.

- **Example 12.3**
- 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.
- a. 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.

- b. 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.
- c. 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.

- **Slotted ALOHA**
- Pure ALOHA has a vulnerable time of 2 × Tfr.
- 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 Tfr seconds and force the station to send only at the beginning of the time slot.
- Figure 12.5 shows an example of frame collisions in slotted ALOHA.

- **Slotted ALOHA**

Figure 12.5 *Frames in a slotted ALOHA network*

- **Slotted ALOHA**
- 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 shows the situation.

- **Slotted ALOHA**

Figure 12.6 *Vulnerable time for slotted ALOHA protocol*

Slotted ALOHA vulnerable time = T_{fr}

- **Throughput**
- It can be proven that the average number of successful transmissions for slotted ALOHA is $S = G \times e^{-G}$.
- The maximum throughput S max is 0.368, when G = 1. In other words, if one frame is generated during one frame transmission time, then 36.8 percent of these frames reach their destination successfully.
- We expect G = 1 to produce maximum throughput because the vulnerable time is equal to the frame transmission time.
- Therefore, if a station generates only one frame in this vulnerable time (and no other station generates a frame during this time), the frame will reach its destination successfully.

The throughput for slotted ALOHA is $S = G \times e^{-G}$.
The maximum throughput $S_{max} = 0.368$ when $G = 1$.

- **Example 12.4**
- 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
- a. 1000 frames per second.
- b. 500 frames per second.
- c. 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.
- **a.** In this case G is 1. So $S = G \times e^{-G} = 0.368$ (36.8 percent).
- This means that the throughput is $1000 \times 0.0368 = 368$ frames.

- Only 368 out of 1000 frames will probably survive.
- Note that this is the maximum throughput case, percentagewise.
- **b**. Here G is 1/2. In this case $S = G \times e^{-G} = 0.303$ (30.3 percent).
- This means that the throughput is $500 \times 0.0303 = 151$. Only 151 frames out of 500 will probably survive.

- **c.** 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.

- **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."
- CSMA can reduce the possibility of collision, but it cannot eliminate it. The reason for this is shown in Figure 12.7, a space and time model of a CSMA network. Stationsare connected to a shared channel (usually a dedicated medium).

- **CSMA**
- The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it.
- In other words, a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

- **Slotted ALOHA**

Figure 12.7 *Space/time model of a collision in CSMA*

At time t_1, station B senses the medium and finds it idle, so it sends a frame. At time t_2 ($t_2 > t_1$), station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C. Station C also sends a frame. The two signals collide and both frames are destroyed.

- **CSMA**
- The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it.
- In other words, a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

- **Vulnerable Time**
- The vulnerable time for CSMA is the propagation time Tp .
- This is the time needed for a signal to propagate from one end of the medium to the other.
- When a station sends a frame and any other station tries to send a frame during this time, a collision will result.
- But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending.
- Figure 12.8 shows the worst case.
- The leftmost station, A, sends a frame at time t 1 , which reaches the rightmost station, D, at time t 1 + Tp .
- The gray area shows the vulnerable area in time and space.

- **Slotted ALOHA**

Figure 12.8 *Vulnerable time in CSMA*

- **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.
- To better understand CSMA/CD, let us look at the first bits transmitted by the two stations involved in the collision.
- Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide.
- In Figure 12.11, stations A and C are involved in the collision.

- **Slotted ALOHA**

Figure 12.11 *Collision of the first bits in CSMA/CD*

- **CSMA/CD**
- At time t 1 , station A has executed its persistence procedure and starts sending the bits of its frame. At time t 2 , station C has not yet sensed the first bit sent by A.
- Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right.
- 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 (or after a short time, but we assume immediately) aborts transmission.
- Station A detects collision at time t 4 when it receives the first bit of C's frame; it also immediately aborts transmission.

- **CSMA/CD**
- Looking at the figure, we see that A transmits for the duration t 4 – t 1 ; C transmits for the duration t 3 – t 2 .
- Now that we know the time durations for the two transmissions, we can show a more complete graph in Figure 12.12.

Figure 12.12 *Collision and abortion in CSMA/CD*

- **Minimum Frame Size**
- For CSMA/CD to work, we need a restriction on the frame size.
- Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission.
- This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection.
- Therefore, the frame transmission time Tfr must be at least two times the maximum propagation time Tp .
- To understand the reason, let us think about the worst-case scenario.

- **Minimum Frame Size**
- If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time Tp to reach the second, and the effect of the collision takes another time T P to reach the first.
- So the requirement is that the first station must still be transmitting after 2T p .

- **Example 12.5**
- 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 μ s, what is the minimum size of the frame?
- **Solution**
- The minimum frame transmission time is $T_{fr} = 2 \times T_p = 51.2$ μ s.
- This means, in the worst case, a station needs to transmit for a period of 51.2 μ s to detect the collision.
- The minimum size of the frame is 10 Mbps × 51.2 μ s = 512 bits or 64 bytes.
- This is actually the minimum size of the frame for Standard Ethernet, as we will see later in the chapter.

- **Procedure**
- Now let us look at the flow diagram for CSMA/CD in Figure 12.13.
- It is similar to the one for the ALOHA protocol, but there are differences.

Figure 12.13 *Flow diagram for the CSMA/CD*

- **Throughput**
- The throughput of CSMA/CD is greater than that of pure or slotted ALOHA.
- The maximum throughput occurs at a different value of G and is based on the persistence method and the value of p in the p-persistent approach.
- For the 1-persistent method, the maximum throughput is around 50 percent when G = 1.
- For the nonpersistent method, the maximum throughput can go up to 90 percent when G is between 3 and 8.

- **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, as shown in Figure 12.15.

Figure 12.15 *Flow diagram of CSMA/CA*

- **CSMA/CA**
- **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.
- Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting.
- The distant station's signal has not yet reached this station.
- The IFS time allows the front of the transmitted signal by the distant station to reach this station.
- After waiting an IFS time, if the channel is still idle, the station can send, but it still needs to wait a time equal to the contention window

- **CSMA/CA**
- **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.
- This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time.
- This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting station.

- **CSMA/CA**
- **Contention Window.**
- One interesting point about the contention window is that the station needs to sense the channel after each time slot.
- However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle.
- This gives priority to the station with the longest waiting time.
- See Figure 12.16.

- However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle.

- This gives priority to the station with the longest waiting time. See Figure 12.16.

Figure 12.16 *Contention window*

- **CSMA/CA**
- **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.

- **Wired LANS: Ethernet**
- **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.
- Project 802 does not seek to replace any part of the OSI model or TCP/IP protocol suite.
- Instead, it is a way of specifying functions of the physical layer and the data-link layer of major LAN protocols.
- The relationship of the 802 Standard to the TCP/IP protocol suite is shown in Figure 13.1.
- The IEEE has subdivided the data-link layer into two sublayers: logical link control (LLC) and media access control (MAC). IEEE has also created severalphysical-layer standards for different LAN protocols.

Figure 13.1 *IEEE standard for LANs*

- **Wired LANS: Ethernet**
- **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.
- The LLC provides a single link-layer control protocol for all IEEE LANs.
- This means LLC protocol can provide interconnectivity between different LANs because it makes the MAC sublayer transparent.

- **Wired LANS: Ethernet**
- **Media Access Control (MAC)**
- IEEE Project 802 has created a sublayer called media access control that defines the specific access method for each LAN.
- For example, it defines CSMA/CD as the media access method for Ethernet LANs and defines the token-passing method for Token Ring and Token Bus LANs.
- As we mentioned in the previous section, part of the framing function is also handled by the MAC layer.

- **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), as shown in Figure 13.2.
- We briefly discuss all these generations.

Figure 13.2 *Ethernet evolution through four generations*

- **STANDARD ETHERNET**
- We refer to the original Ethernet technology with the data rate of 10 Mbps as the Standard Ethernet.
- Although most implementations have moved to other technologies in the Ethernet evolution, there are some features of the Standard Ethernet that have not changed during the evolution.
- We discuss this standard version to pave the way for understanding the other three technologies.

- **Characteristics**
- **Connectionless and Unreliable Service**
- Ethernet provides a connectionless service, which means each frame sent is independent of the previous or next frame.
- Ethernet has no connection establishment or connection termination phases.
- The sender sends a frame whenever it has it; the receiver may or may not be ready for it.
- The sender may overwhelm the receiver with frames, which may result in dropping frames.
- If a frame drops, the sender will not know about it. Since IP, which is using the service of Ethernet, is also connectionless, it will not know about it either.

- **Characteristics**
- If the transport layer is also a connectionless protocol, such as UDP, the frame is lost and salvation may only come from the application layer.
- However, if the transport layer is TCP, the sender TCP does not receive acknowledgment for its segment and sends it again.
- Ethernet is also unreliable like IP and UDP.
- If a frame is corrupted during transmission and the receiver finds out about the corruption, which has a high level of probability of happening because of the CRC-32, the receiver drops the frame silently.
- It is the duty of high-level protocols to find out about it.

- **Frame Format**
- The Ethernet frame contains seven fields, as shown in Figure 13.3.

Figure 13.3 *Ethernet frame*

- **Characteristics**
- If the transport layer is also a connectionless protocol, such as UDP, the frame is lost and salvation may only come from the application layer.
- However, if the transport layer is TCP, the sender TCP does not receive acknowledgment for its segment and sends it again.
- Ethernet is also unreliable like IP and UDP.
- If a frame is corrupted during transmission and the receiver finds out about the corruption, which has a high level of probability of happening because of the CRC-32, the receiver drops the frame silently.
- It is the duty of high-level protocols to find out about it.

- **MAC Sublayer**
  - **In Standard Ethernet, the MAC sublayer governs the operation of the access method.**
  - It also frames data received from the upper layer and passes them to the physical layer.
- **Frame Format**
  - The Ethernet frame contains seven fields:
  - preamble,
  - SFD,
  - DA,
  - SA,
  - length or type of protocol data unit (PDU),
  - upper-layer data, and
  - the CRC.
  - **Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium.**
  - Acknowledgments must be implemented at the higher layers. The format oft he MAC frame is shown in Figure 13.4.

Figure 802.3 MAC frame

Preamble: 56 bits of alternating 1s and 0s.

SFD: Start frame delimiter, flag (10101011)

- **Preamble.**
  - **The first field of the 802.3 frame contains 7 bytes (56 bits) of alternating 0s and 1s that alerts the receiving system to the coming frame and enables it to synchronize its input timing.**
  - The pattern provides only an alert and a timing pulse.
  - The 56-bit pattern allows the stations to miss some bits at the beginning of the frame.
- **Start frame delimiter (SFD).**
  - **The second field (l byte: 10101011) signals the beginning of the frame.**
  - The SFD warns the station or stations that this is the last chance for synchronization. .
  - **The last 2 bits is 11** and alerts the receiver that the next field is the destination address.

- **Destination address (DA).**
  - The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet.
- **Source address (SA).**
  - The SA field is also 6 bytes and contains the physical address of the sender of the packet.
- **Length or type.**
  - **This field is defined as a type field or length field.**
  - **The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame.**
  - The IEEE standard used it as the length field to define the number of bytes in the data field.
  - Both uses are common today.
- **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,

- **Frame Length**
  - **Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame.**

Figure Minimum and maximum lengths

- **An Ethernet frame needs to have a minimum length of 512** bits or 64 bytes.
- Part of this length is the header and the trailer. If we count 18 bytes of header and trailer (6 bytes of source address, 6 bytes of destination address, 2 bytes of length or type, and 4 bytes of CRC), then the minimum length of data from the upper layer is 64 - 18 = 46 bytes.
- **The standard defines the maximum length of a frame** (without preamble and SFD field) as 1518 bytes.
- If we subtract the 18 bytes of header and trailer, the maximum length of the payload is 1500 bytes.

Frame length:

Minimum: 64 bytes (512 bits)

Maximum: 1518 bytes (12,144 bits)

- **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 6-byte physical address.
  - As shown in Figure 13.6, the Ethernet address is 6 bytes (48 bits), nonnally written in hexadecimal notation, with a colon between the bytes.

Figure *Example of an Ethernet address in hexadecimal notation*

- **Unicast, Multicast, and Broadcast Addresses**
  - **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.
  - Figure 13.7 shows how to distinguish a unicast address from a multicast address.
  - **If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast.**

Figure *Unicast and multicast addresses*

- **The broadcast address is a special case of the multicast address; the recipients are all the stations on the LAN.**
- A broadcast destination address is forty-eight 1s.
- The broadcast destination address is a special case of the multicast address in which all bits are 1s.

- **Numerical Examples**
- **1.Define the type of the following destination addresses:**
  - **a. 4A:30:10:21:10:1A**
  - **b. 47:20:1B:2E:08:EE**
  - **c. FF:FF:FF:FF:FF:FF**
- **Solution:-**
  - To find the type of the address, we need to look at the second hexadecimal digit from the left.
  - If it is even, the address is unicast.
  - If it is odd, the address is multicast.
  - If all digits are F's, the address is broadcast.
  - Therefore, we have the following:
  - a. This is a **unicast address** because A in binary is 1010 (even).
  - b. This is a **multicast address** because 7 in binary is 0111 (odd).
  - c. This is a **broadcast address** because all digits are F's.

- **Note:**
  - The way the addresses are sent out on line is different from the way they are written in hexadecimal notation.
  - The transmission is left-to-right, byte by byte; however, for each byte, the least significant bit is sent first and the most significant bit is sent last.
  - This means that the bit that defines an address as unicast or multicast arrives first at the receiver.
- **2. Show how the address 47:20:1B:2E:08:EE is sent out on line**.
- **Solution:-**
  - 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:
  - 11100010 00000100 11011000 01110100 00010000 01110111

- **Physical Layer**
  - **The Standard Ethernet defines several physical layer implementations; four of the most common.**

Figure *Categories of Standard Ethernet*

- **Encoding and Decoding**
- **All standard implementations use digital signaling (baseband) at 10 Mbps.**
- **At the sender, data are converted to a digital signal using the Manchester scheme; at the receiver, the received signal is interpreted as Manchester and decoded into data.**
- Manchester encoding is self-synchronous, providing a transition at each bit interval.
- Figure shows the encoding scheme for Standard Ethernet.

Figure *Encoding in a Standard Ethernet implementation*

- **10 Base5: Thick Ethernet**
- The first implementation is called 10Base5, thick Ethernet, or Thicknet.
- The nickname derives from the size of the cable, which is roughly the size of a garden hose and too stiff to bend with your hands.
- **10Base5 was the first Ethernet specification to use a bus topology with an external transceiver (transmitter/receiver) connected via a tap to a thick coaxial cable.**
- Figure shows a schematic diagram of a l0Base5 implementation.

Figure *10Base5 implementation*

- **The transceiver is responsible for transmitting, receiving, and detecting collisions.**
- The transceiver is connected to the station via a transceiver cable that provides separate paths for sending and receiving.
- This means that collision can only happen in the coaxial cable.
- **The maximum length of the coaxial cable must not exceed 500 m, otherwise, there is excessive degradation of the signal.**

- **10Base2: Thin Ethernet**
- The second implementation is called 10Base2, thin Ethernet, or Cheapernet.
- **10Base2 also uses a bus topology, but the cable is much thinner and more flexible.**
- The cable can be bent to pass very close to the stations. In this case, the transceiver is normally part of the network interface card (NIC), which is installed inside the station.
- Figure shows the schematic diagram of a 10Base2 implementation.

- **Note that the collision here occurs in the thin coaxial cable.**
- **This implementation is more cost effective than 10Base5 because thin coaxial cable is less expensive than thick coaxial and the tee connections are much cheaper than taps.**
- **Installation is simpler because the thin coaxial cable is very flexible.**
- However, the length of each segment cannot exceed 185 m (close to 200 m) due to the high level of attenuation in thin coaxial cable.
- **10Base-T: Twisted-Pair Ethernet**
- **The third implementation is called 10Base-T or twisted-pair Ethernet.**
- **10Base-T uses a physical star topology.**
- The stations are connected to a hub via two pairs of twisted cable, as shown in Figure 13.12.
- Note that two pairs of twisted cable create two paths (one for sending and one for receiving) between the station and the hub.
- Any collision here happens in the hub.

- Compared to 10Base5 or 10Base2, we can see that the hub actually replaces the coaxialcable as far as a collision is concerned.
- **The maximum length of the twisted cable here is defined as 100 m, to minimize the effect of attenuation in the twisted cable.**

Figure *10Base-T implementation*

- **10Base-F: Fiber Ethernet**
- **Although there are several types of optical fiber 10-Mbps Ethernet, the most common is called 10Base-F.**
- **10Base-F uses a star topology to connect stations to a hub.**
- The stations are connected to the hub using two fiber-optic cables, as shown in Figure.

Figure *10Base-F implementation*

Table *Summary of Standard Ethernet implementations*

Characteristics	*10Base5*	*10Base2*	*10Base-T*	*10Base-F*
Media	Thick coaxial cable	Thin coaxial cable	2 UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m
Line encoding	Manchester	Manchester	Manchester	Manchester

- **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.

- **FAST ETHERNET (100 MBPS)**
- 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:
  - Upgrade the data rate to 100 Mbps.
  - Make it compatible with Standard Ethernet.
  - Keep the same 48-bit address.
  - Keep the same frame format.

- **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.
- For example, a device with a maximum data rate of 10 Mbps can communicate with a device with a 100 Mbps data rate (but which can work at a lower rate). We can summarize the goal of autonegotiation as follows.

- **Autonegotiation**
- 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 (see Figure 13.16).

Figure 13.16 *Encoding for Fast Ethernet implementation*

- **100Base-TX** uses two pairs of twisted-pair cable (either category 5 UTP or STP).
- For this implementation, the MLT-3 scheme was selected since it has good bandwidth performance.
- However, since MLT-3 is not a self-synchronous line coding scheme, 4B/5B block coding is used to provide bit synchronization by preventing the occurrence of a long sequence of 0s and 1s.
- This creates a data rate of 125 Mbps, which is fed into MLT-3 for encoding.

- **100Base-FX** uses two pairs of fiber-optic cables.
- Optical fiber can easily handle high bandwidth requirements by using simple encoding schemes.
- The designers of 100Base-FX selected the NRZ-I encoding scheme for this implementation.
- However, NRZ-I has a bit synchronization problem for long sequences of 0s (or 1s, based on the encoding).
- To overcome this problem, the designers used 4B/5B block encoding, as we described for 100Base-TX.
- The block encoding increases the bit rate from 100 to 125 Mbps, which can easily be handled by fiber-optic cable.

- A new standard, called **100Base-T4**, was designed to use category 3 or higher UTP.
- The implementation uses four pairs of UTP for transmitting 100 Mbps.
- Encoding/decoding in 100Base-T4 is more complicated.
- As this implementation uses category 3 UTP, each twisted-pair cannot easily handle more than 25 Mbaud.
- In this design, one pair switches between sending and receiving.
- Three pairs of UTP category 3, however, can handle only 75 Mbaud (25 Mbaud) each.

Summary

Fast Ethernet implementation at the physical layer can be categorized as either two-wire or four-wire. The two-wire implementation can be either shielded twisted pair (STP), which is called *100Base-TX*, or fiber-optic cable, which is called *100Base-FX*. The four-wire implementation is designed only for unshielded twisted pair (UTP), which is called *100Base-T4*. Table 13.2 is a summary of the Fast Ethernet implementations. We discussed encoding in Chapter 4.

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.
- Use the same 48-bit address.
- Use the same frame format.
- Keep the same minimum and maximum frame lengths.
- 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.
- Almost all implementations of Gigabit Ethernet follow the full-duplex approach, so we mostly ignore the half-duplex mode.

- **Full-Duplex Mode**
- In full-duplex mode, there is a central switch connected to all computers or other switches.
- In this mode, for each input port, each switch has buffers in which data are stored until they are transmitted.
- Since the switch uses the destination address of the frame and sends a frame out of the port connected to that particular destination, there is no collision.
- This means that CSMA/CD is not used.
- Lack of collision implies that the maximum length of the cable is determined by the signal attenuation in the cable, not by the collision detection process.

- **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. However, as we saw before, the maximum length of the network in this approach is totally dependent on the minimum frame size.
- Three methods have been defined:
  - traditional,
  - carrier extension, and
  - frame bursting.

- **Traditional**
- In the traditional approach, we keep the minimum length of the frame as in traditional Ethernet (512 bits).
- However, because the length of a bit is 1/100 shorter in Gigabit Ethernet than in 10-Mbps Ethernet, the slot time for Gigabit Ethernet is 512 bits $\times$ 1/1000 μs, which is equal to 0.512 μs.
- The reduced slot time means that collision is detected 100 times earlier.
- This means that the maximum length of the network is 25 m.
- This length may be suitable if all the stations are in one room, but it may not even be long enough to connect the computers in one single office.

- **Carrier Extension**
- To allow for a longer network, we increase the minimum frame length.
- The carrier extension approach defines the minimum length of a frame as 512 bytes (4096 bits).
- This means that the minimum length is 8 times longer.
- This method forces a station to add extension bits (padding) to any frame that is less than 4096 bits.
- In this way, the maximum length of the network can be increased 8 times to a length of 200 m.
- This allows a length of 100 m from the hub to the station.

- **Frame Bursting**
- Carrier extension is very inefficient if we have a series of short frames to send; each frame carries redundant data.
- To improve efficiency, frame bursting was proposed.
- Instead of adding an extension to each frame, multiple frames are sent.
- However, to make these multiple frames look like one frame, padding is added between the frames (the same as that used for the carrier extension method) so that the channel is not idle.
- In other words, the method deceives other stations into thinking that a very large frame has been transmitted.

- **Physical Layer**
- The physical layer in Gigabit Ethernet is more complicated than that in Standard or Fast Ethernet. We briefly discuss some features of this layer.
- **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).
- In other words, we have four implementations.
- 1000Base-T was designed in response to those users who had already installed this wiring for other purposes such as Fast Ethernet or telephone services.

- **Encoding**
- Figure 13.17 shows the encoding/decoding schemes for the four implementations.
- Gigabit Ethernet cannot use the Manchester encoding scheme because it involves a very high bandwidth (2 GBaud).
- The two-wire implementations use an NRZ scheme, but NRZ does not self-synchronize properly.
- To synchronize bits, particularly at this high data rate, 8B/10B block encoding is used.
- This block encoding prevents long sequences of 0s or 1s in the stream, but the resulting stream is 1.25 Gbps.
- Note that in this implementation, one wire (fiber or STP) is used for sending and one for receiving.

- **Encoding**
- In the four-wire implementation it is not possible to have 2 wires for input and 2 for output, because each wire would need to carry 500 Mbps, which exceeds the capacity for category 5 UTP.
- As a solution, 4D-PAM5 encoding, is used to reduce the bandwidth.
- Thus, all four wires are involved in both input and output; each wire carries 250 Mbps, which is in the range for category 5 UTP cable.
- need to carry 500 Mbps, which exceeds the capacity for category 5 UTP. As a solution, 4D-PAM5 encoding, is used to reduce the bandwidth.
- Thus, all four wires are involved in both input and output; each wire carries 250 Mbps, which is in the range for category 5 UTP cable.

Figure 13.17 *Encoding in Gigabit Ethernet implementations*

Implementation Summary

Table 13.3 is a summary of the Gigabit Ethernet implementations. S-W and L-W mean short-wave and long-wave respectively.

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.

- **10 GIGABIT ETHERNET**
- 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.
- 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 shows a summary of the 10 Gigabit Ethernet implementations.

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

- **Virtual LANS**
- A station is considered part of a LAN if it physically belongs to that LAN.
- The criterion of membership is geographic.
- What happens if we need a virtual connection between two stations belonging to two different physical LANs? We can roughly define a virtual local area network (VLAN) as a local area network configured by software, not by physical wiring.
- Let us use an example to elaborate on this definition.
- Figure 17.10 shows a switched LAN in an engineering firm in which nine stations are grouped into three LANs that are connected by a switch.

Figure 17.10 *A switch connecting three LANs*

- **Virtual LANS**
- The first three engineers work together as the first group, the next two engineers work together as the second group, and the last four engineers work together as the third group.
- The LAN is configured to allow this arrangement.
- But what would happen if the administrators needed to move two engineers from the first group to the third group, to speed up the project being done by the third group?
- The LAN configuration would need to be changed.
- The network technician must rewire.

- **Virtual LANS**
- Figure 17.11 shows the same switched LAN divided into VLANs.
- The whole idea of VLAN technology is to divide a LAN into logical, instead of physical, segments.
- A LAN can be divided into several logical LANs, called VLANs.
- Each VLAN is a work group in the organization. If a person moves from one group to another, there is no need to change the physical configuration.
- The group membership in VLANs is defined by software, not hardware.
- Any station can be logically moved to another VLAN.
- All members belonging to a VLAN can receive broadcast messages sent to that particular VLAN. This means that if a station moves from VLAN 1 to VLAN 2, it receives broadcast messages sent to VLAN 2, but no longer receives broadcast messages sent to VLAN 1.

Figure 17.11 *A switch using VLAN software*

- **Virtual LANS**
- It is obvious that the problem in our previous example can easily be solved by using VLANs.
- Moving engineers from one group to another through software is easier than changing the configuration of the physical network.
- VLAN technology even allows the grouping of stations connected to different switches in a VLAN.
- Figure 17.12 shows a backbone local area network with two switches and three VLANs.
- Stations from switches A and B belong to each VLAN.

Figure 17.12 *Two switches in a backbone using VLAN software*

- **Virtual LANS**
- This is a good configuration for a company with two separate buildings.
- Each building can have its own switched LAN connected by a backbone.
- People in the first building and people in the second building can be in the same work group even though they are connected to different physical LANs.
- From these three examples, we can see that a VLAN defines broadcast domains.
- VLANs group stations belonging to one or more physical LANs into broadcast domains.
- The stations in a VLAN communicate with one another as though they belonged to a physical segment.

- **Membership**
- What characteristic can be used to group stations in a VLAN?
- Vendors use different characteristics such as interface numbers, port numbers, MAC addresses, IP addresses, IP multicast addresses, or a combination of two or more of these.
- **Interface Numbers**
- Some VLAN vendors use switch interface numbers as a membership characteristic.
- For example, the administrator can define that stations connecting to ports 1, 2, 3, and 7 belong to VLAN 1, stations connecting to ports 4, 10, and 12 belong to VLAN 2, and so on.

- **MAC Addresses**
- Some VLAN vendors use the 48-bit MAC address as a membership characteristic.
- For example, the administrator can stipulate that stations having MAC addresses E2:13:42:A1:23:34 and F2:A1:23:BC:D3:41 belong to VLAN 1.
- **IP Addresses**
- Some VLAN vendors use the 32-bit IP address (see Chapter 18) as a membership characteristic.
- For example, the administrator can stipulate that stations having IP addresses 181.34.23.67, 181.34.23.72, 181.34.23.98, and 181.34.23.112 belong to VLAN 1.

- **Multicast IP Addresses**
- Some VLAN vendors use the multicast IP address as a membership characteristic.
- Multicasting at the IP layer is now translated to multicasting at the data-link layer.
- **Combination**
- Recently, the software available from some vendors allows all these characteristics to be combined.
- The administrator can choose one or more characteristics when installing the software.
- In addition, the software can be reconfigured to change the settings.

- **Configuration**
- How are the stations grouped into different VLANs?
- Stations are configured in one of three ways: manually, semiautomatically, and automatically.
- **Manual Configuration**
- In a manual configuration, the network administrator uses the VLAN software to manually assign the stations into different VLANs at setup.
- Later migration from one VLAN to another is also done manually.
- Note that this is not a physical configuration; it is a logical configuration.
- The term manually here means that the administrator types the port numbers, the IP addresses, or other characteristics, using the VLAN software.

- **Automatic Configuration**
- In an automatic configuration, the stations are automatically connected or disconnected from a VLAN using criteria defined by the administrator.
- For example, the administrator can define the project number as the criterion for being a member of a group.
- When a user changes projects, he or she automatically migrates to a new VLAN.
- **Semiautomatic Configuration**
- A semiautomatic configuration is somewhere between a manual configuration and an automatic configuration.
- Usually, the initializing is done manually, with migrations done automatically.

- **Communication between Switches**
- In a multi-switched backbone, each switch must know not only which station belongs to which VLAN, but also the membership of stations connected to other switches.
- For example, in Figure 17.12, switch A must know the membership status of stations connected to switch B, and switch B must know the same about switch A.
- Three methods have been devised for this purpose: **table maintenance, frame tagging, and time-division multiplexing.**
- **Table Maintenance**
- In this method, when a station sends a broadcast frame to its group members, the switch creates an entry in a table and records station membership.
- The switches send their tables to one another periodically for updating.

- **Frame Tagging**
- In this method, when a frame is traveling between switches, an extra header is added to the MAC frame to define the destination VLAN.
- The frame tag is used by the receiving switches to determine the VLANs to be receiving the broadcast message.
- **Time-Division Multiplexing (TDM)**
- In this method, the connection (trunk) between switches is divided into time-shared channels. For example, if the total number of VLANs in a backbone is five, each trunk is divided into five channels.
- The traffic destined for VLAN 1 travels in channel 1, the traffic destined for VLAN 2 travels in channel 2, and so on. The receiving switch determines the destination VLAN by checking the channel
- from which the frame arrived.

- **IEEE Standard**
- In 1996, the IEEE 802.1 subcommittee passed a standard called 802.1Q that defines the format for frame tagging.
- The standard also defines the format to be used in multi-switched backbones and enables the use of multivendor equipment in VLANs.
- IEEE 802.1Q has opened the way for further standardization in other issues related to VLANs.
- Most vendors have already accepted the standard.

- **Advantages**
- **Cost and Time Reduction**
- VLANs can reduce the migration cost of stations going from one group to another.
- Physical reconfiguration takes time and is costly.
- Instead of physically moving one station to another segment or even to another switch, it is much easier and quicker to move it by using software.
- **Creating Virtual Work Groups**
- VLANs can be used to create virtual work groups.
- For example, in a campus environment, professors working on the same project can send broadcast messages to one another without the necessity of belonging to the same department. This can reduce traffic if the multicasting capability of IP was previously used.

- **Advantages**
- **Security**
- VLANs provide an extra measure of security.
- People belonging to the same group can send broadcast messages with the guaranteed assurance that users in other groups will not receive these messages.