Unit -2
Data Link Layer

Introduction

The data link layer takes the packets from the network layer and encapsulates them into frames for transmission.
Each frame contains three parts: frame header, payload field for holding packets, and frame trailer.

Fig: Frame

Header: The header consists of control information whose role is to guide the whole frame to its correct destination.
Frame header includes Source and Destination address field, Physical Link Control field, Flow control field, and Congestion Control field etc.,

Trailer: Data-link Layer adds also a trailer at the end of each frame. The trailer is responsible for ensuring that frames are received intact or undamaged.

Error Detection and Correction

Data can be corrupted during transmission, some applications require that errors can be detected and corrected.

- **Types of Errors:** Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. This interference can change the shape of the signal.

The term ***single-bit error*** means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.
The term ***burst error*** means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.
Figure shows the effect of a single-bit and a burst error on a data unit.

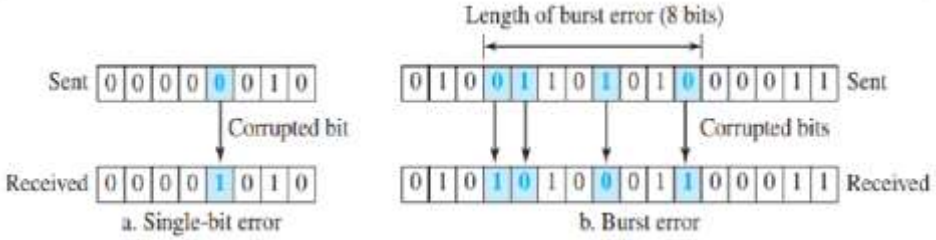


Fig: Single-bit and burst error

- **Redundancy:** The central concept in detecting or correcting errors is **redundancy**. To be able to detect or correct errors, we need to send some extra bits with our data. These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits.

- **Detection versus Correction**

The correction of errors is more difficult than the detection. In **error detection**, we are only looking to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of corrupted bits.
In **error correction**, we need to know the exact number of bits that are corrupted and, more importantly, their location in the message. The number of errors and the size of the message are important factors.

1. Block Coding

In block coding, we divide our message into blocks, each of *k* bits, called ***datawords.*** We add *r* redundant bits to each block to make the length $n = k + r$. The resulting *n*-bit blocks are called ***codewords.*** How the extra *r* bits are chosen or calculated.

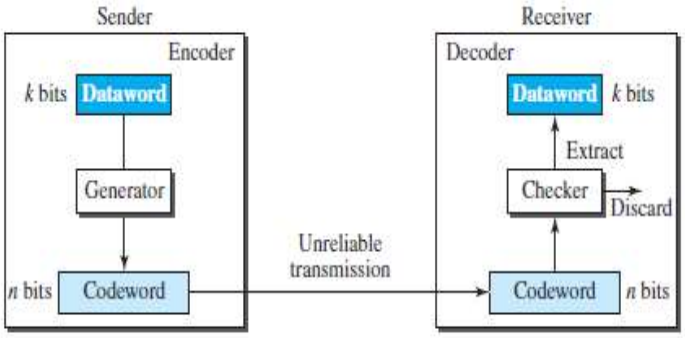


Fig: Process of error detection in block coding

❖ Linear Block Codes

Almost all block codes used today belong to a subset of block codes called ***linear block codes***.

Parity-Check Code

In this parity code, a k-bit dataword is changed to an n-bit codeword where n = k + 1. The extra bit, called the parity bit, is selected to make the total number of 1s in the codeword even.
Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of:

- 1 is added to the block if it contains odd number of 1's, and
- 0 is added if it contains even number of 1's

This scheme makes the total number of 1's even, that is why it is called even parity checking.

Figure shows a possible structure of an encoder (at the sender) and a decoder (at the receiver).

Example:

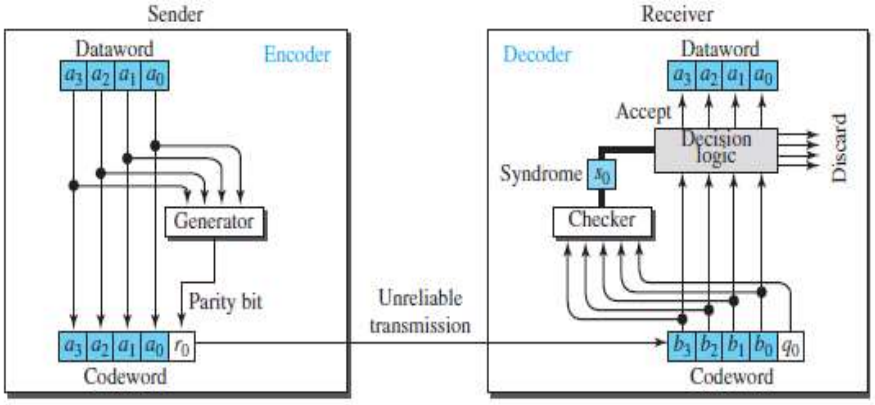


Fig: ***Encoder and decoder for simple parity-check code***

The calculation is done in **modular arithmetic**. The encoder uses a generator that takes a copy of a 4-bit dataword (*a*0, *a*1, *a*2, and *a*3) and generates a parity bit *r*0. The dataword bits and the parity bit create the 5-bit codeword.

$$r_0 = a_3 + a_2 + a_1 + a_0 \quad \text{(modulo-2)}$$

The sender sends the codeword, which may be corrupted during transmission. The receiver receives a 5-bit word. The checker at the receiver does the same thing as the generator in the sender with one exception: The addition is done over all 5 bits. The result, which is called the ***syndrome,*** is just 1 bit.
The syndrome is 0 when the number of 1s in the received codeword is even; otherwise, it is 1.

$$s_0 = b_3 + b_2 + b_1 + b_0 + q_0 \quad \text{(modulo-2)}$$

The syndrome is passed to the decision logic analyzer. If the syndrome is 0, there is no detectable error in the received codeword; the data portion of the received codeword is accepted as the dataword; if the syndrome is 1, the data portion of the received codeword is discarded.

2. Cyclic Codes

Cyclic Redundancy Check

The cyclic redundancy check (CRC), which is used in networks such as LANs and WANs.

CRC encoder and decoder:

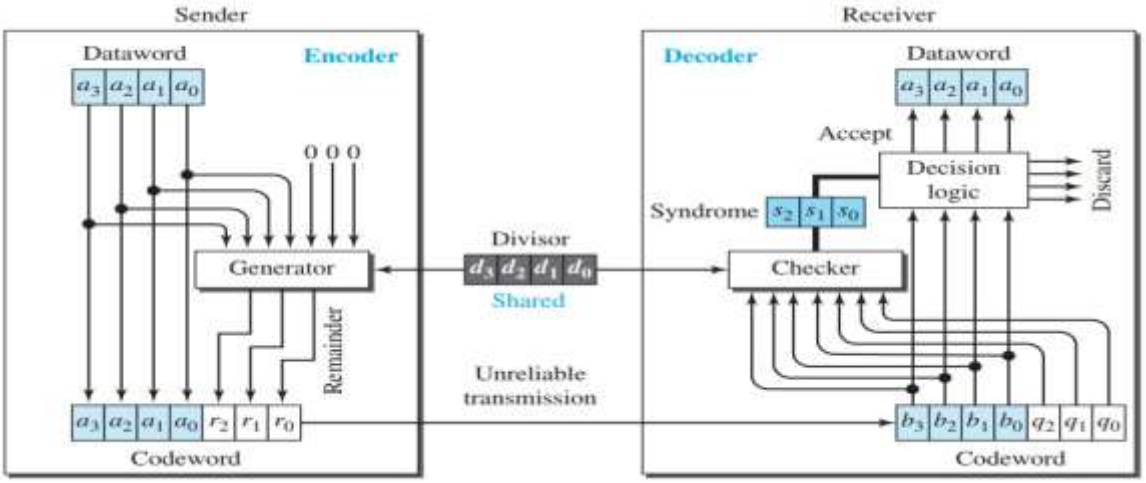


Fig: CRC encoder and decoder

CRC encoder

- In the **encoder**, the dataword has k bits (4 here); the codeword has n bits (7 here). The size of the dataword is increased by adding n − k (3 here) 0s to the right-hand side of the word.
- The n-bit result is providing to the generator. The generator uses a divisor of size n − k + 1 (4 here), predefined.
- The generator divides the increased dataword by the divisor (modulo-2 division). The quotient of the division is discarded; the remainder (r2r1r0) is appended to the dataword to create the codeword.

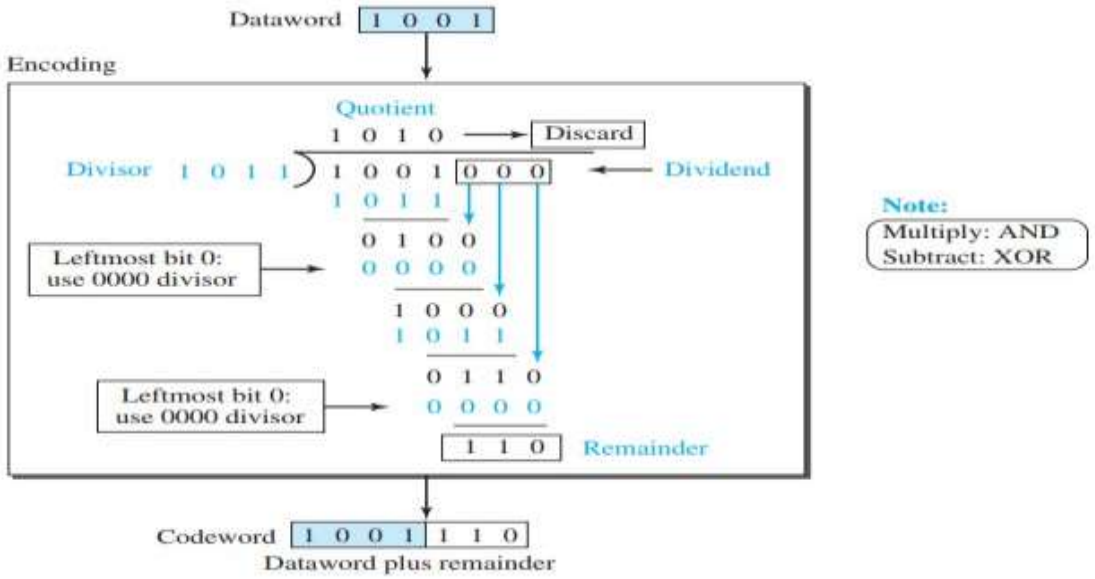
CRC decoder

- The decoder receives the codeword. A copy of all n bits is fed to the checker, which is a copy of the generator.
- The remainder produced by the checker is a syndrome of n − k (3 here) bits, which is fed to the decision logic analyzer.
- The analyzer has a simple function. If the syndrome bits are all 0s, the 4 leftmost bits of the codeword are accepted as the dataword (interpreted as no error); otherwise, the 4 bits are discarded (error).

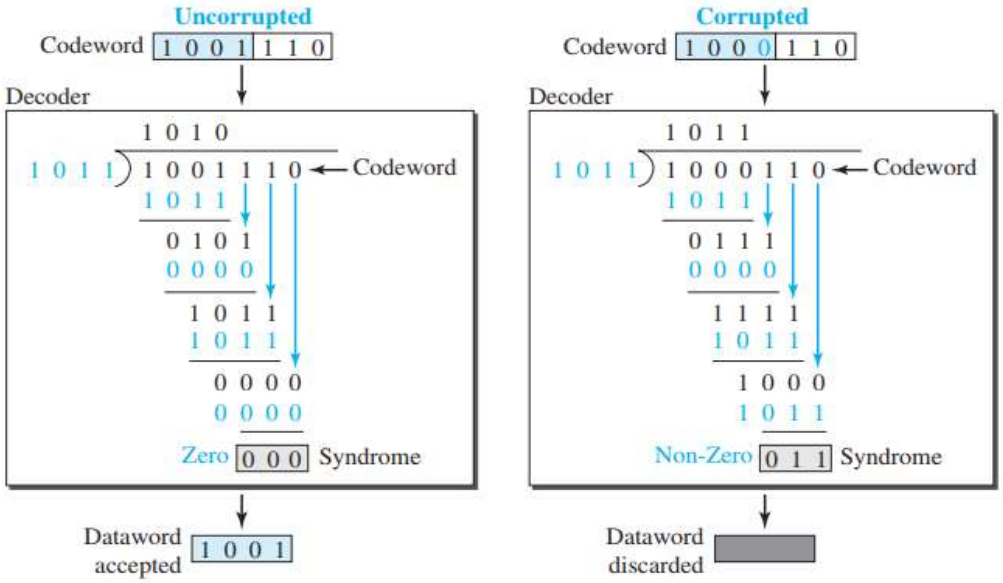
NRCM
your roots to success...

Example:

Encoder



Decoder:



3. Forward Error Correction

Numbers of methods are used for error detection and retransmission. However, retransmission of corrupted and lost packets is not useful for real-time multimedia transmission because it creates an unacceptable delay in reproducing: we need to wait until the lost or corrupted packet is resent. We need to correct the error or reproduce the packet immediately.

Several schemes have been designed and used in these cases that are collectively referred to as forward error correction (FEC) techniques.

Using Hamming Distance: The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits. We show the Hamming distance between two words x and y as d(x, y). We may wonder why Hamming distance is important for error detection.
The reason is that the Hamming distance between the received codeword and the sent codeword is the number of bits that are corrupted during transmission. ***For example***, if the codeword 00000 is sent and 01101 is received, 3 bits are in error and the Hamming distance between the two is d(00000, 01101) = 3.
If the Hamming distance between the sent and the received codeword is not zero, the codeword has been corrupted during transmission. The Hamming distance can easily be found if we apply the XOR operation ($\oplus$) on the two words and count the number of 1s in the result.

Hamming code example: The key to the Hamming Code is the use of extra parity bits to allow the identification of a single error. Create the code word as follows:

a) Mark all bit positions that are powers of two as parity bits. (Positions 1, 2, 4, 8, 16, 32, 64, etc.)
b) All other bit positions are for the data to be encoded. (Positions 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, etc.)
c) Each parity bit calculates the parity for some of the bits in the code word. The position of the parity bit determines the sequence of bits that it alternately checks and skips.

- 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).
- 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).
- 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).
- 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).

d) 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.

Codeword: data word + additional bits (parity)

P4	D4	D3	D2	P3	D1	P2	P1

Example:
Data word: 1001, we check for even parity.

7	6	5	4	3	2	1
1	0	0	P3	1	P2	P1

- P1 bit is calculated using the bits positions: 1, 3, 5, and 7. To find the parity bit P1, we check for even parity. Since the total number of 1's in all the bit positions corresponding to P1 is an even number value (parity bit's value) = 0.
- P2 bit is calculated using the bits positions: 2, 3, 6, and 7. Since the total number of 1's in all the bit positions corresponding to P2 is an even number value (parity bit's value) = 0.
- P3 bit is calculated using the bits positions: 4 , 5, 6 and 7. Since the total number of 1's in all the bit positions corresponding to P2 is an even number value (parity bit's value) = 1.

Thus, the data transferred is:

7	6	5	4	3	2	1
1	0	0	1	1	0	0

If the data transfer perfectly, there is no error.

Error detection and correction: Suppose in the above example the 3rd bit is changed from 1 to 0 during data transmission, then it gives new parity values in the binary number:

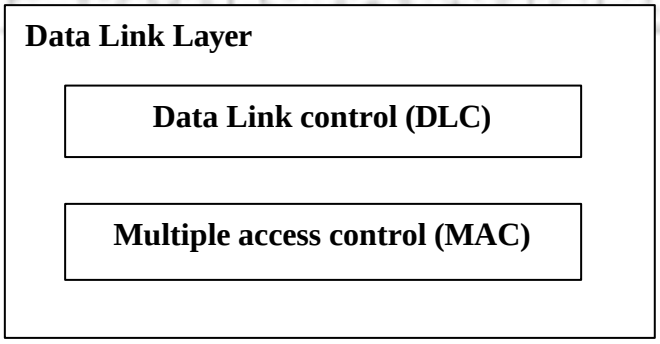
7	6	5	4	3	2	1
1	0	0	1	0	0	0

P1 bit positions(1,3,5,7), even parity value=1
P2 bit positions(2,3,6,7), even parity value=1
P3 bit positions(4,5,6,7), even parity value=0

The bits give the binary number as 011 whose decimal representation is 3. Thus, the bit 3 contains an error. To correct the error the 3rd bit is changed from 0 to 1.

Data Link Control (DLC)

The data-link layer is divided into two sublayers. The upper sublayer of the data-link layer (DLC). The lower sublayer, multiple access control (MAC).



1. DLC Services

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

❖ **Framing:** Framing means how to organize the bits that are carried by the physical layer. The data-link layer 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 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.

When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole frame. When a message is divided into smaller frames, a single-bit error affects only that small frame.

➢ **Frame Size**

Frames can be of fixed or variable size. In ***fixed-size framing***, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. In ***variable-size framing***, we need a way to define the end of one frame and the beginning of the next.

Two approaches were used for this purpose: a ***character-oriented*** approach and a ***bit-oriented*** approach.

- ***Character-Oriented Framing:*** In *character-oriented (or byte-oriented) framing,* data to be carried are 8-bit characters from a coding system such as ASCII.

The *header*, which normally carries the source and destination addresses and other control information, and the *trailer*, which carries error detection redundant bits.
To separate one frame from the next, an 8-bit (1-byte) **flag** is added at the beginning and the end of a frame. Figure shows the format of a frame in a character-oriented protocol.

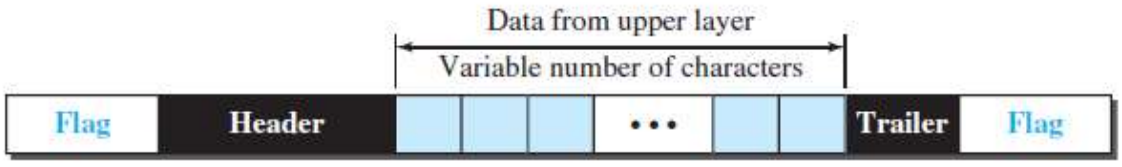


Fig: A frame in a character-oriented protocol

Character-oriented framing was popular when only text was exchanged by the data-link layers. The flag could be selected to be any character not used for text communication.

Now we send other types of information such as graphs, audio, and video; any character used for the flag could also be part of the information.

To solve this problem, a byte-stuffing strategy was added to character-oriented framing. In **byte stuffing** is the process of adding one extra byte whenever there is a flag or escape character in the text. The data section is stuffed with an extra byte. This byte is usually called the *escape character (ESC)* and has a predefined bit pattern.

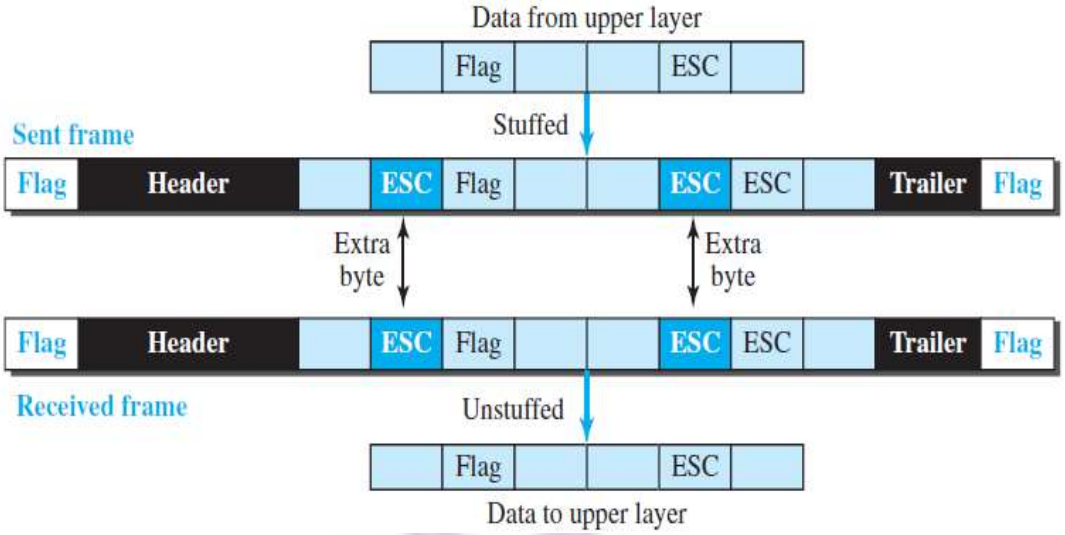


Fig: Byte stuffing and unstuffing

- ***Bit-Oriented Framing:*** In *bit-oriented framing*, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on. However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other.

Most protocols use a special 8-bit pattern **flag, 01111110**, as the delimiter to define the beginning and the end of the frame, as shown in Figure.

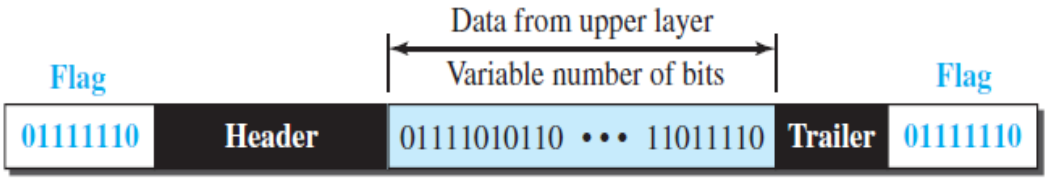


Fig: A frame in a bit-oriented protocol

If the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit to prevent the pattern from looking like a flag. The strategy is called **bit stuffing.** In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added.

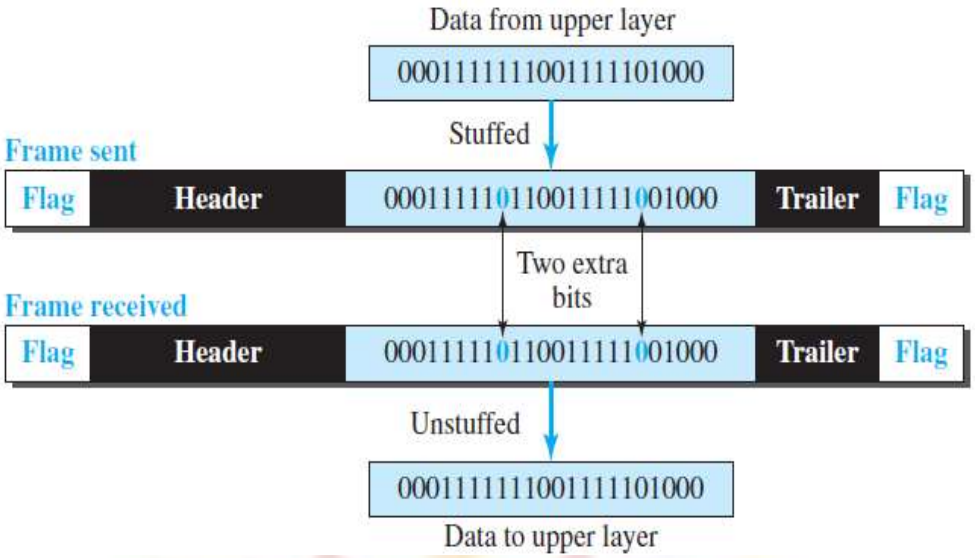


Fig: Bit stuffing and unstuffing

❖ **Flow and Error Control:** One of the responsibilities of the data-link control sublayer is flow and error control at the data-link layer.

Flow Control: Whenever an entity produces items and another entity consumes them, there should be a balance between production and consumption rates. If the items are produced faster than they can be consumed, the consumer can be overwhelmed and may need to discard some items. We need to prevent losing the data items at the consumer site.

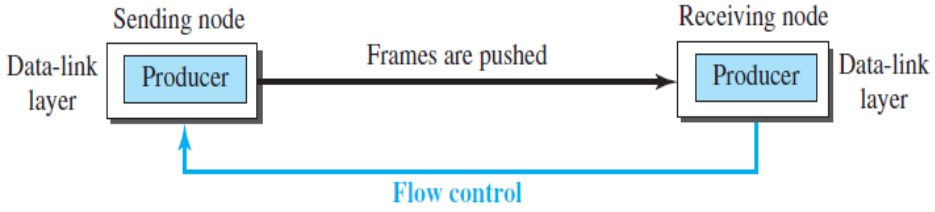


Fig: Flow control at the data-link layer

Buffers

Flow control can be implemented in several ways, one of the solutions is normally to use two *buffers*; one at the sending data-link layer and the other at the receiving data-link layer.

A buffer is a set of memory locations that can hold packets at the sender and receiver. The flow control communication can occur by sending signals from the consumer to the producer. When the buffer of the receiving data-link layer is full, it informs the sending data-link layer to stop pushing frames.

Error Control: we need to implement error control at the data-link layer to prevent the receiving node from delivering corrupted packets to its network layer. Error control at the data-link layer is normally very simple and implemented using one of the following two methods.

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

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

2. noiseless channels and noisy channels protocols

The how the data link layer can combine framing, flow control, and error control to achieve the delivery of data from one node to another. The protocols are normally implemented in software by using one of the common programming languages.
We divide the discussion of protocols into those that can be used for noiseless (error-free) channels and those that can be used for noisy (error-creating) channels.

- Protocols for noiseless channel: Simplest , Stop-and-Wait
- Protocols for noisy channel: Stop-and-Wait ARQ, Go-Hack-N ARQ, Selective Repeat ARQ

All the protocols we discuss are unidirectional in the sense that the data frames travel from one node, called the sender, to another node, called the receiver. Although special frames, called acknowledgment (ACK) and negative acknowledgment (NAK) can flow in the opposite direction for flow and error control purposes, data flow in only one direction.

❖ **Noiseless Channels:** Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel: ***Simplest , Stop-and-Wait***

- **Simplest Protocol:** Our first protocol, which we call the Simplest, is one that has no flow or error control. It is a unidirectional protocol in which data frames are traveling in only one direction-from the sender to receiver.

We assume that the receiver can immediately handle any frame it receives with a processing time. The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately.

your tools to success...

Design:

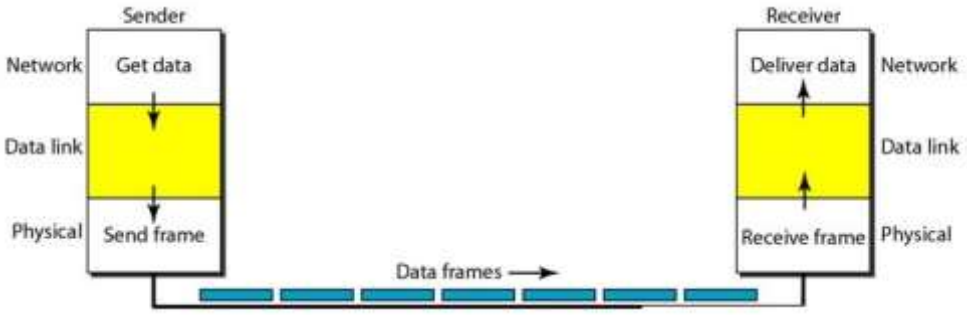
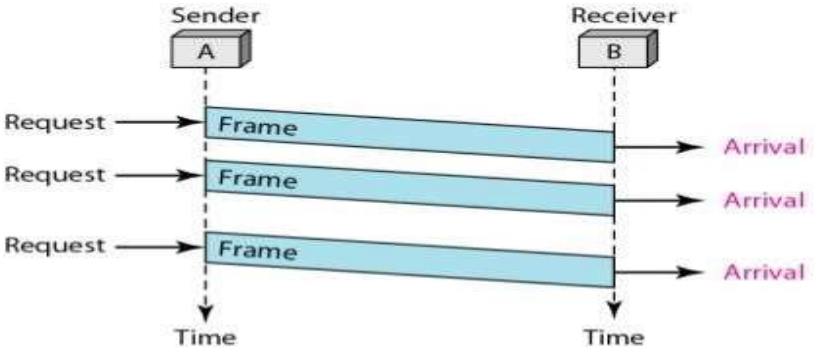


Fig: The design of the simplest protocol with no flow or error control

Flow diagram:



- **Stop-and-Wait:** The receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either the discarding of frames or denial of service.

To prevent the receiver from becoming overwhelmed with frames, we somehow need to tell the sender to slow down. There must be feedback from the receiver to the sender.

The protocol we discuss now is called the Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver, and then sends the next frame.

We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction. We add flow control to protocol.

Design:

your roots to success...

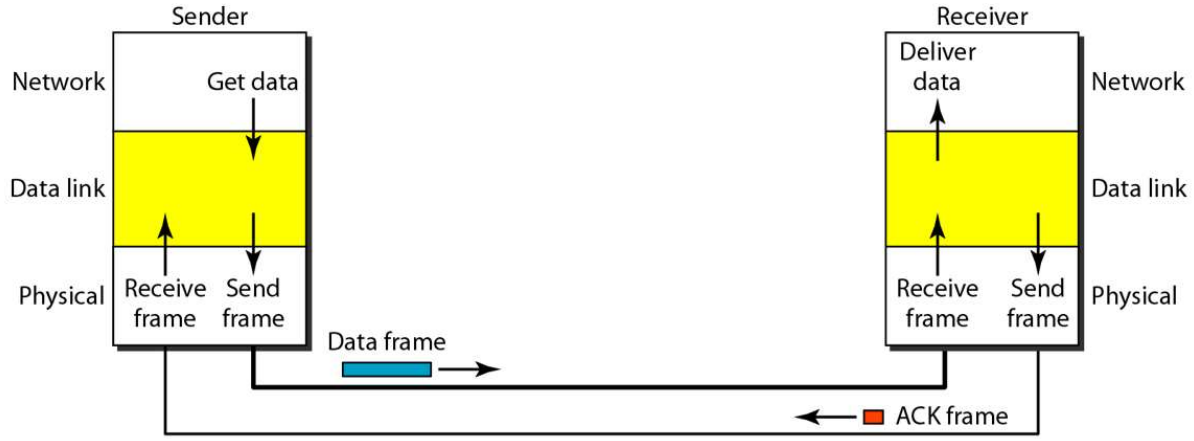
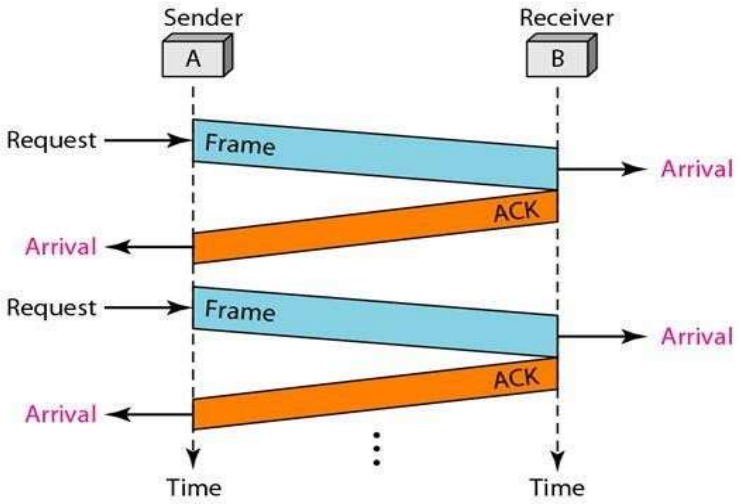


Fig: Design of Stop-and-Wait Protocol

Flow diagram:



❖ **Noisy Channels**: Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We can ignore the error or we need to add error control to our protocols. We discuss three protocols in this section that use error control.

- **Stop-and-Wait Automatic Repeat Request:** the Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ), adds a simple error control mechanism to the Stop-and-Wait Protocol. Let us see how this protocol detects and corrects errors.

To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded.

The completed and lost frames need to be resent in this protocol. The sender keeps a copy of the sent frame. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted.

Sequence Numbers: the protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame.

Acknowledgment Numbers Since the sequence numbers must be suitable for both data frames and ACK frames, we use this convention: The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver. For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1.

Design:

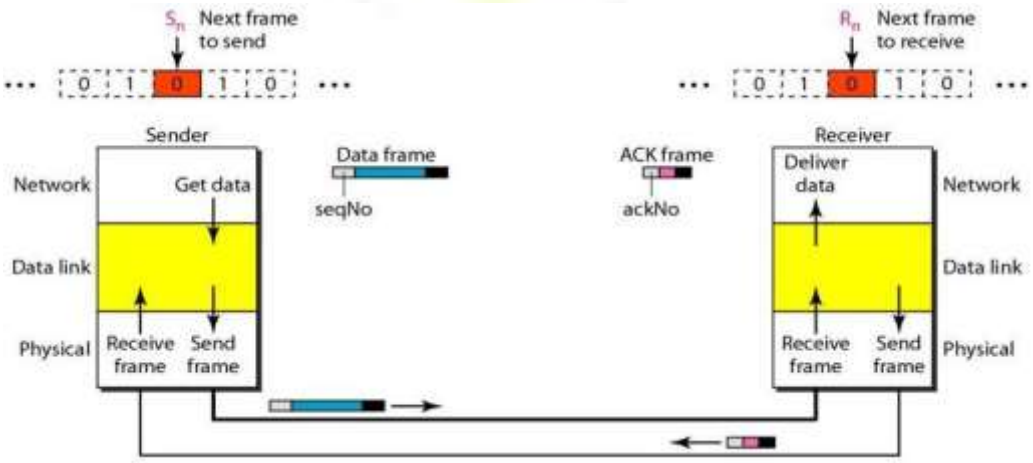
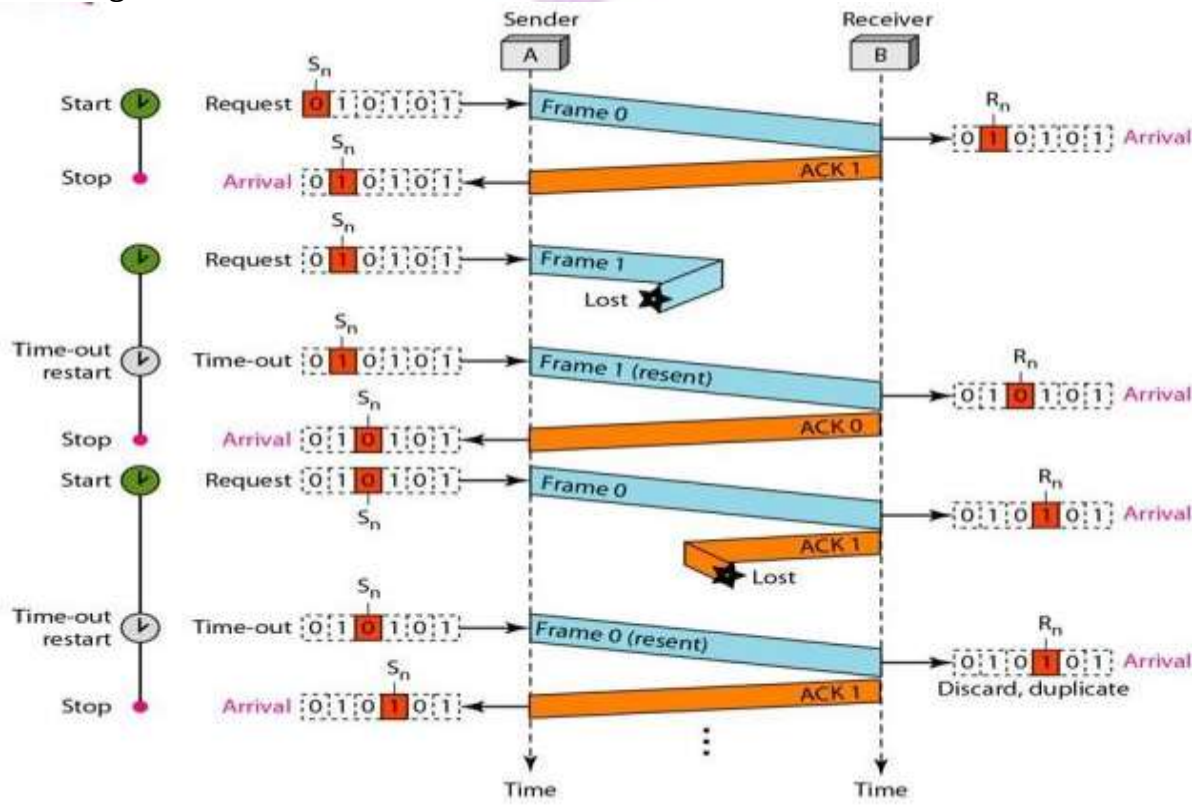


Fig: Design of the Stop-and-Wait ARQ Protocol

Flow diagram:



- **Go-Back-N Automatic Repeat Request:** To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment. The first is called Go-Back-N Automatic Repeat Request protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.

Sequence Numbers: Frames from a sending station are numbered sequentially. In the Go-Back-N Protocol, the sequence numbers are modulo 2^m, where m is the size of the sequence number field in bits.

Sliding Window: In this protocol , the sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver. In other words, the sender and receiver need to deal with only part of the possible sequence numbers.

The send window the maximum size of the window is 2m.The receive window makes sure that the correct data frames are received and that the correct acknowledgments are sent. The size of the receive window is always 1.

Design:

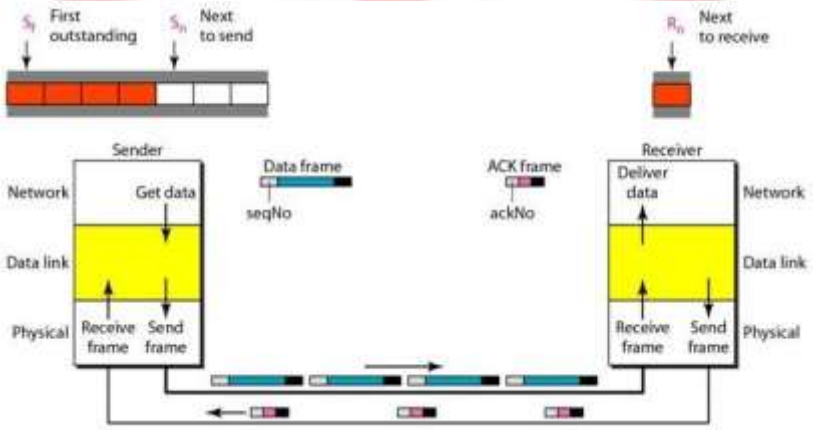
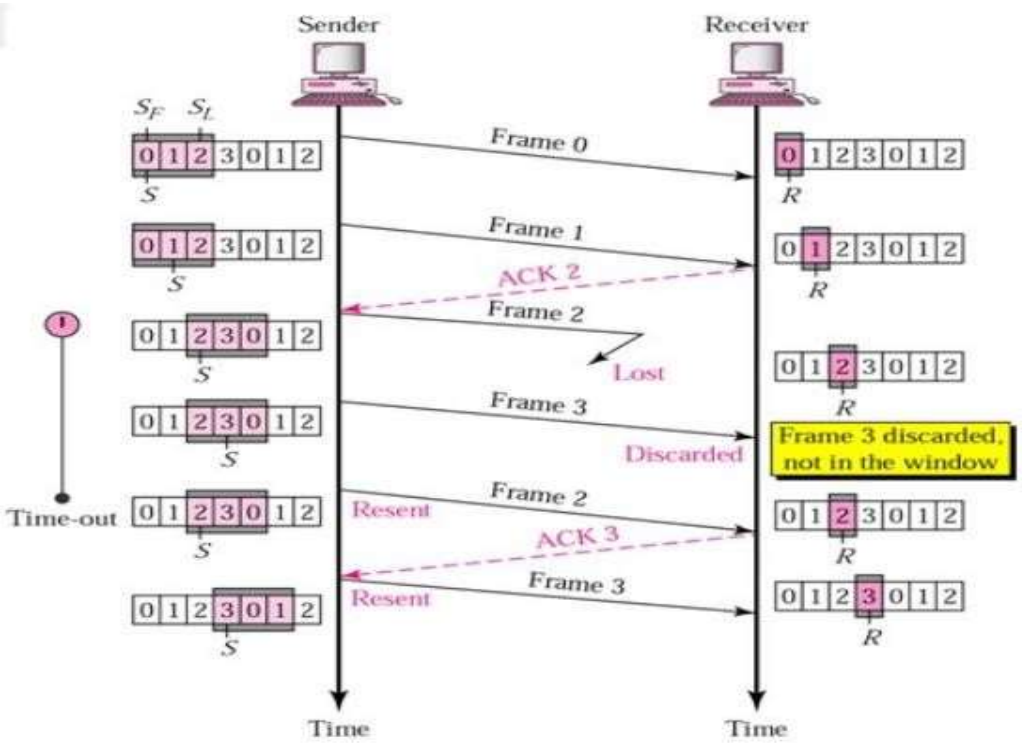


Fig: Design of Go-Back-NARQ

Flow diagram:

your roots to success...



- **Selective Repeat Automatic Repeat Request:** Go-Back-N ARQ simplifies the process at the receiver site. The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link. In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames.

For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective Repeat ARQ.

Windows: The Selective Repeat Protocol also uses two windows: a send window and a receive window. The receive window is the same size as the send window.

Design:

NRCM
your roots to success...

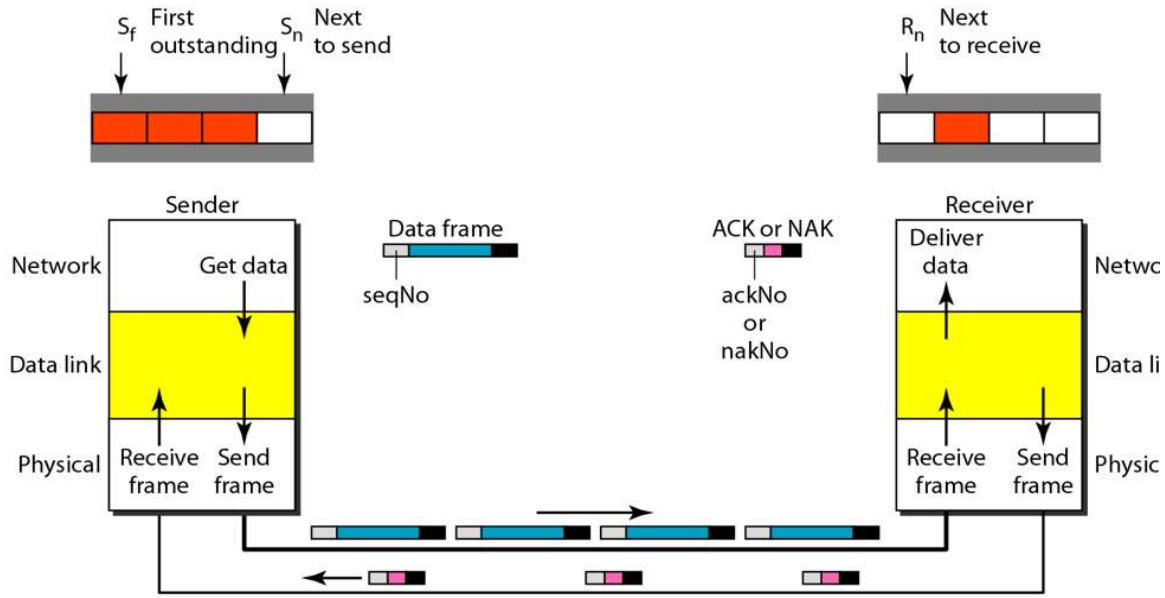
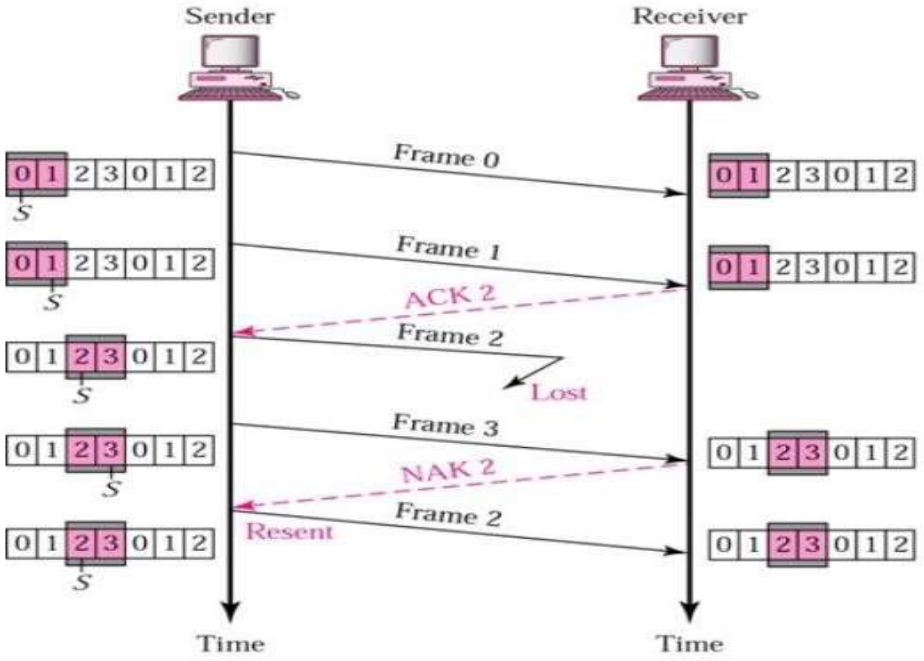


Fig: Design of Selective Repeat ARQ

Flow diagram:



3. 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.

❖ Configurations and Transfer Modes

HDLC provides two common transfer modes that can be used in different configurations: *normal response mode (NRM)* and *asynchronous balanced mode (ABM)*.

In ***normal response mode (NRM)***, the station configuration is unbalanced. We have one primary station and multiple secondary stations. A *primary station* can send commands; a *secondary station* can only respond. The NRM is used for both point-to-point and multipoint links, as shown in Figure.

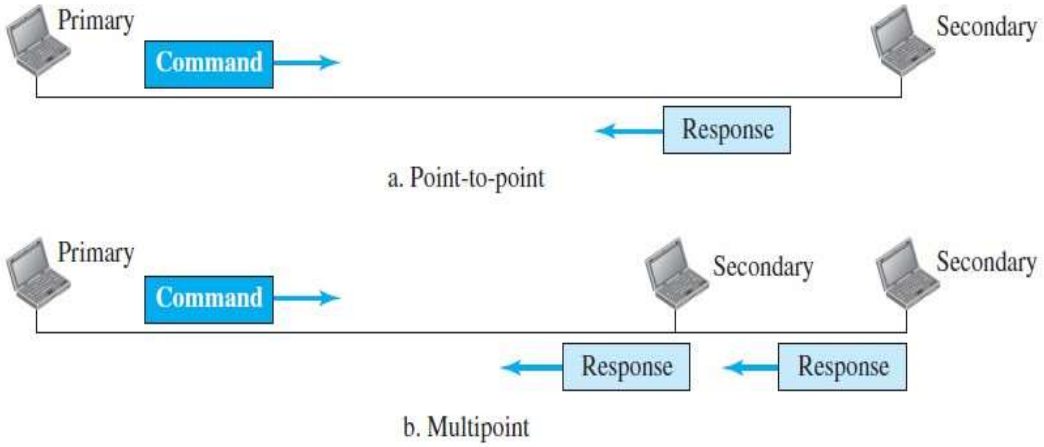


Fig: Normal response mode

In ***ABM,*** the configuration is balanced. The link is point-to-point, and each station can function as a primary and a secondary (acting as peers), as shown in Figure. This is the common mode today.

Fig: Asynchronous balanced mode

❖ Framing

To provide the flexibility necessary to support all the options possible in the modes and configurations, HDLC defines ***three types*** of frames: ***information frames (I-frames), supervisory frames (S-frames),*** **and** ***unnumbered frames (U-frames).***

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

- ***Iframes*** are used to data-link user data and control information relating to user data (piggybacking).
- ***S-frames*** are used only to transport control information.
- **U-frames** are reserved for system management. Information carried by U-frames is intended for managing the link itself.

Each frame in HDLC may contain up to ***six fields***, as shown in Figure. 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.

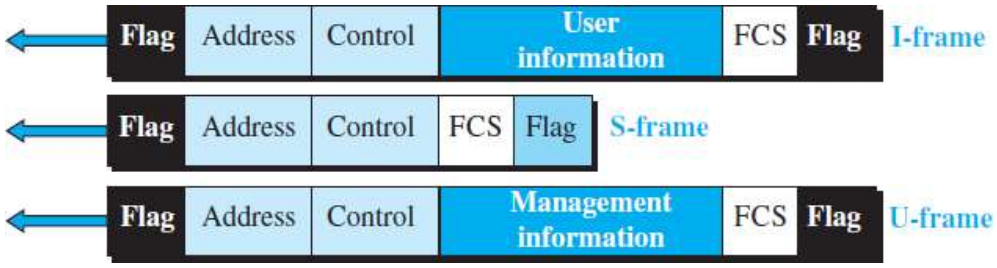


Fig: HDLC frames

- ***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.
- ***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.

❖ Control field

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.

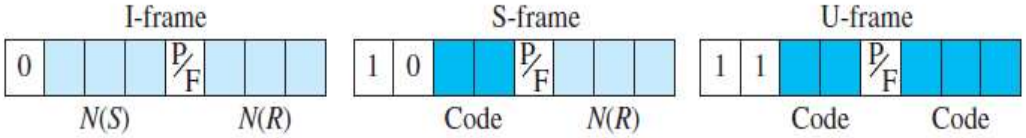


Fig: Control field format for the different frame types

- ***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 **first bit** defines the type. If the first bit of the control field is **0**, this means the frame is an I-frame.
The next 3 bits, called ***N(S)***, define the sequence number of the frame. Note that with 3 bits, we can define a sequence number between 0 and 7.
The last 3 bits, called ***N(R)***, correspond to the acknowledgment number when piggybacking is used.
The **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. It means ***final*** when the frame is sent by a secondary to a primary.

- ***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.

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

- ***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. U-frames contain an information field, but one used for system management information, not user data.

If the **first 2 bits** of the control field are **11**, this means the frame is an U-frame.
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.

Example:

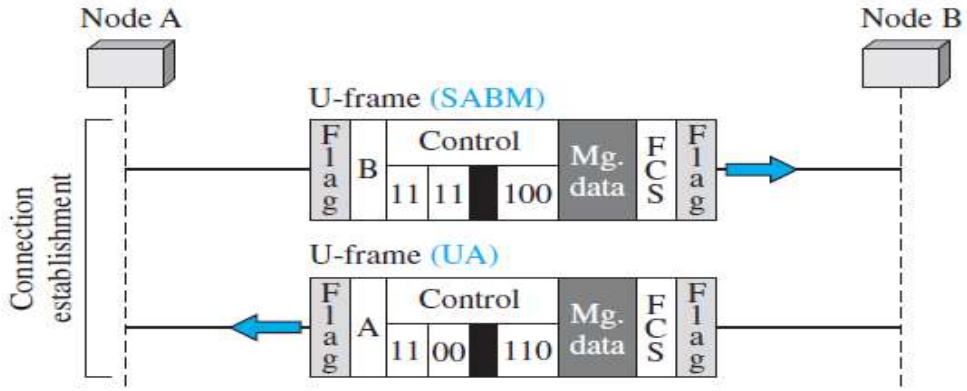


Fig: Example of connection and disconnection

4. 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.

❖ Services

The designers of PPP have included several services to make it suitable for a point-to-point protocol.

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. Authentication is also provided in the protocol.
- The new version of PPP, called *Multilink PPP*, provides connections over multiple links.
- One interesting feature of PPP is that it provides network address configuration.
- This is particularly useful when a home user needs a temporary network address to connect to the Internet.

❖ Framing

PPP uses a character-oriented (or byte-oriented) frame. Figure shows the format of a PPP frame.

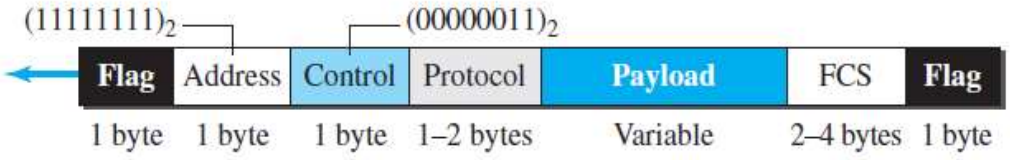


Fig: PPP frame format

The description of each field follows:

- ***Flag.*** A PPP frame starts and ends with a 1-byte flag with the bit pattern 01111110.
- ***Address.*** The address field in this protocol is a constant value and set to 11111111 (broadcast address).
- ***Control.*** This field is set to the constant value 00000011. PPP does not provide any flow control. Error control is also limited to error detection.
- ***Protocol.*** The protocol field defines what is being carried in the data field: either user data or other information. This field is by default 2 bytes long, but the two parties can agree to use only 1 byte.
- ***Payload field.*** This field carries either the user data or other information. The data field is a sequence of bytes with the default of a maximum of 1500 bytes;
- ***FCS.*** The frame check sequence (FCS) is simply a 2-byte or 4-byte standard CRC.

Byte Stuffing

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

❖ Transition Phases

A PPP connection goes through phases which can be shown in a *transition phase* diagram. The transition diagram starts with the *dead* state. In this state, there is no active carrier (at the physical layer) and the line is quiet.

When one of the two nodes starts the communication, the connection goes into the *establish* state. In this state, options are negotiated between the two parties.

If the two parties agree that they need *authentication* (for example, if they do not know each other), then the system needs to do authentication (an extra step); otherwise, the parties can simply start communication.

The link-control protocol packets are used for this purpose. Several packets may be exchanged here. Data transfer takes place in the *open* state. When a connection reaches this state, the exchange of data packets can be started.

The connection remains in this state until one of the endpoints wants to terminate the connection. In this case, the system goes to the *terminate* state. The system remains in this state until the carrier is dropped, which moves the system to the *dead* state again.

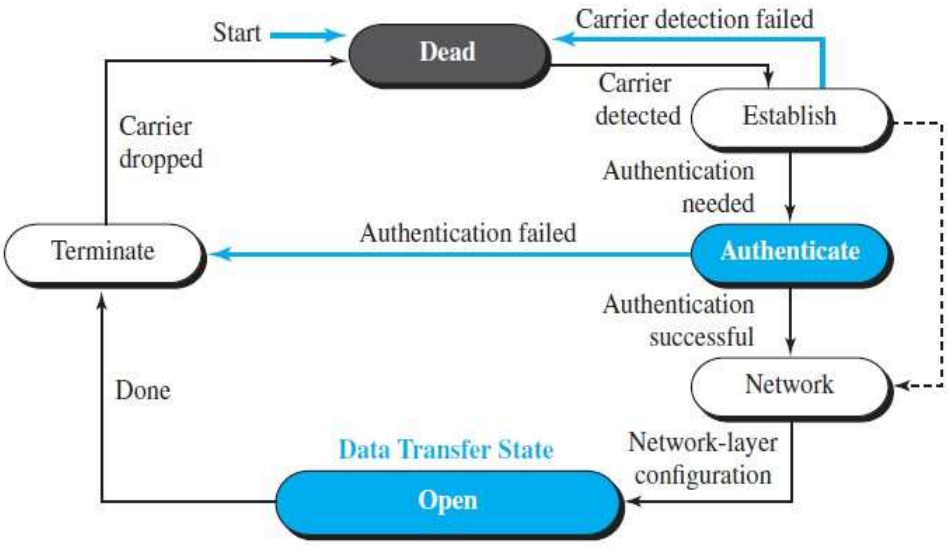


Fig: Transition phases

❖ 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).

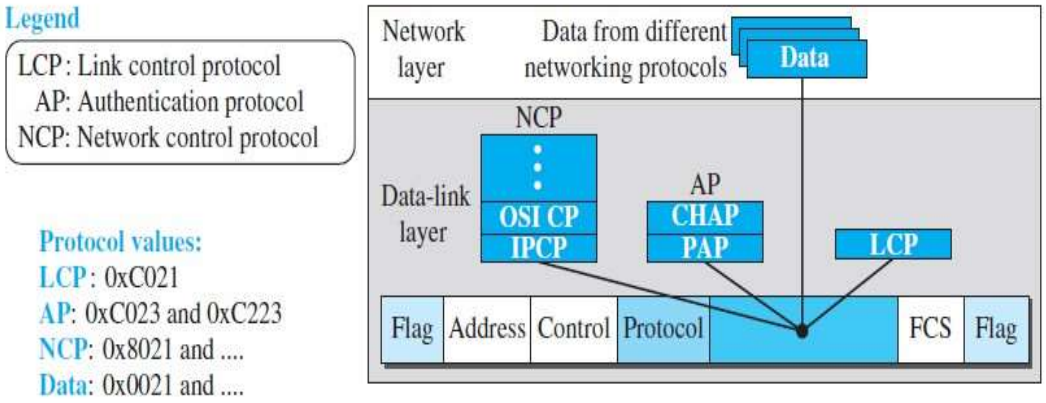


Fig: Multiplexing in PPP

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.

All LCP packets are carried in the payload field of the PPP frame with the protocol field set to C021 in hexadecimal.

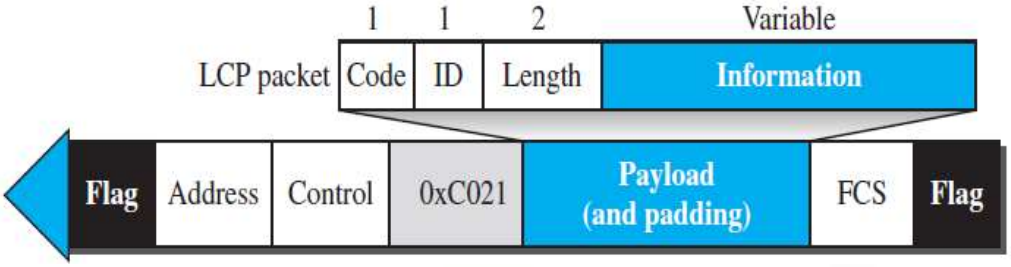


Fig: LCP packet encapsulated in a frame

Authentication Protocols

Authentication means validating the identity of a user who needs to access a set of resources. PPP has created ***two protocols*** for authentication: Password Authentication Protocol and Challenge Handshake Authentication Protocol.

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

a. The user who wants to access a system sends 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.

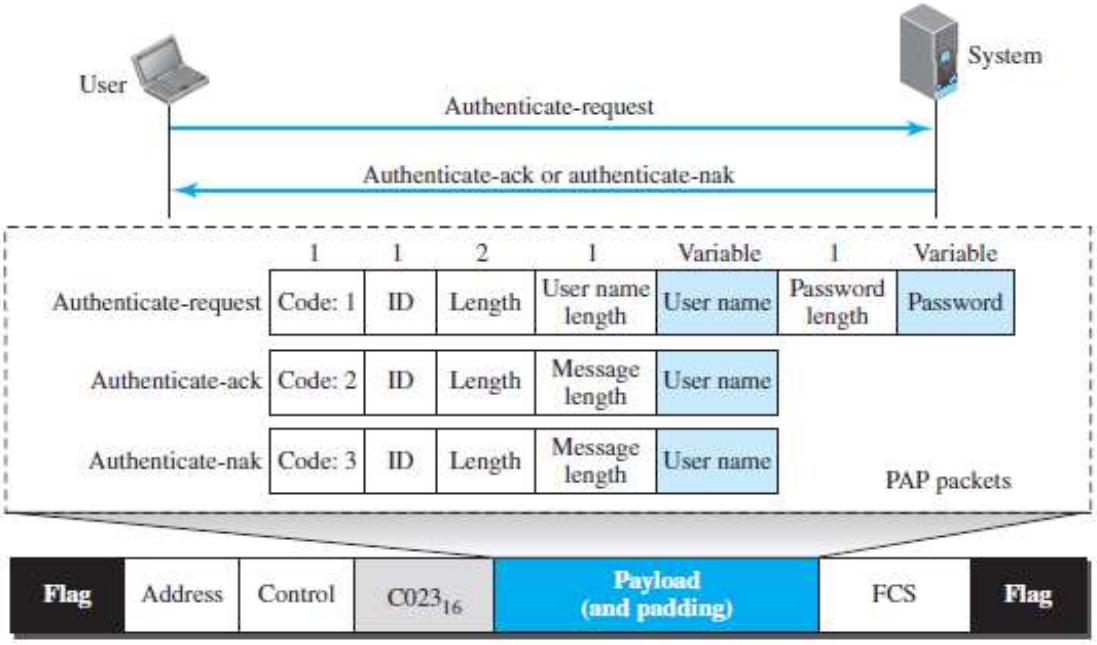


Fig: PAP packets encapsulated in a PPP frame

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.

CHAP

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

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.

your roots to success...

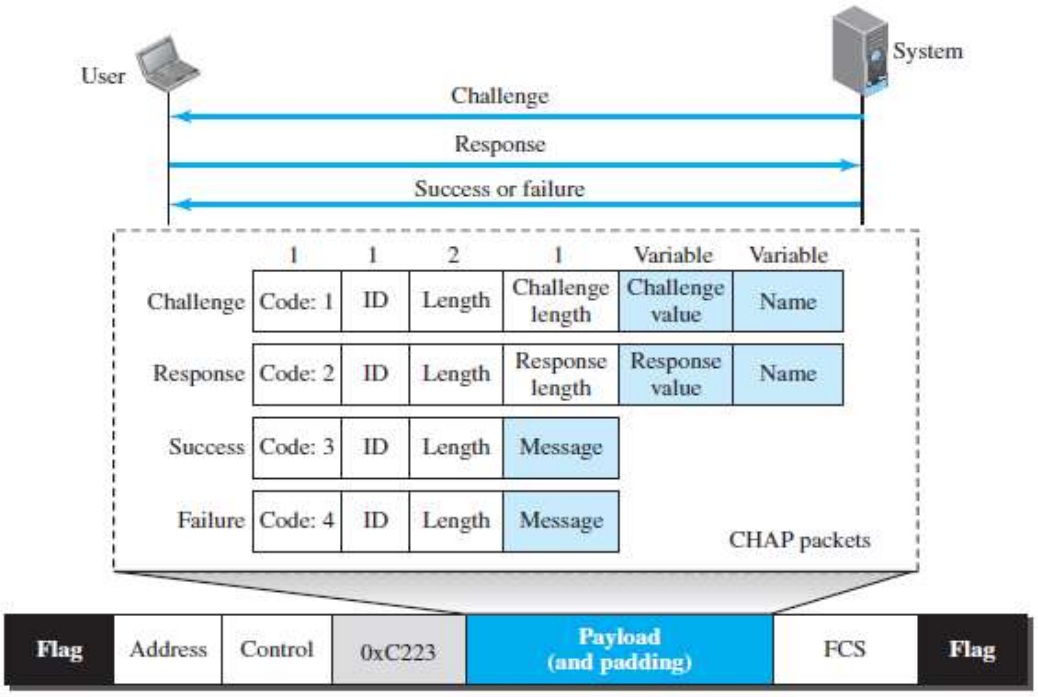


Fig: CHAP packets encapsulated in a PPP frame

CHAP packets are encapsulated in the PPP frame with the protocol value C223 in hexadecimal.

Network Control Protocols

PPP is a multiple-network-layer protocol. It can carry a network-layer data packet from protocols defined by the Internet.

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. Note that the value of the protocol field in hexadecimal is 8021.

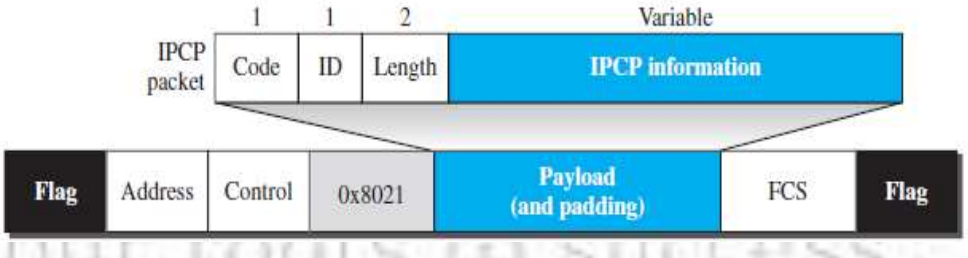


Fig: IPCP packet encapsulated in PPP frame

Media Access Control (MAC)

When nodes or stations are connected and use a common link, called a *multipoint* or *broadcast link*, we need a multiple-access protocol to coordinate access to the link. The problem of controlling the access to the medium is similar to the rules of speaking in an assembly.

Many protocols have been devised to handle access to a shared link. All of these protocols belong to a sublayer in the data-link layer called *media access control (MAC).* We categorize them into three groups, as shown in Figure.

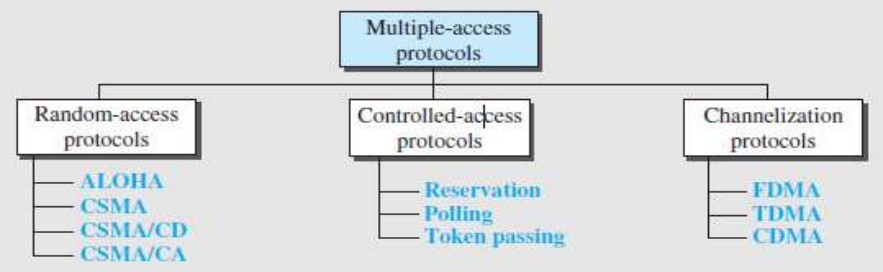


Fig: Taxonomy of multiple-access protocols

1. Random Access

In **random-access** or **contention** methods, no station is superior to another station and none is assigned control over another.

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.

The random-access methods: **ALOHA, CSMA, CSMA/CD, CSMA/CA**

❖ ALOHA

ALOHA, the earliest random access method was developed at the University of Hawaii in early 1970. 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 distorted.

➢ *Pure ALOHA*

The original ALOHA protocol is called ***pure ALOHA.*** This is a simple but well-designed 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 shows an example of frame collisions in pure ALOHA.

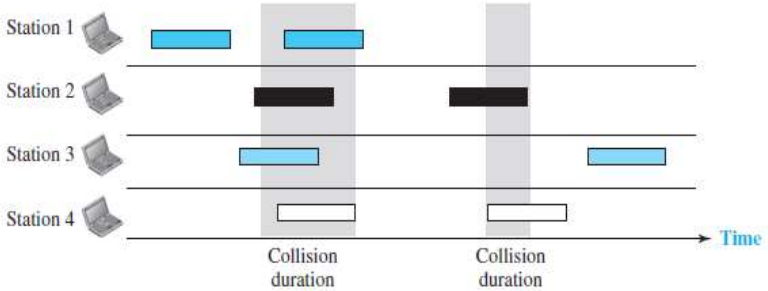


Fig: Frames in a pure ALOHA network

There are four stations 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 shows that only two frames survive: one frame from station 1 and one frame from station 3.

Pure ALOHA has a method to prevent congesting the channel with retransmitted frames. After a maximum number of retransmission attempts *Kmax*, a station must give up and try later. The value of *Kmax* is usually chosen as 15. Figure shows the procedure for pure ALOHA.

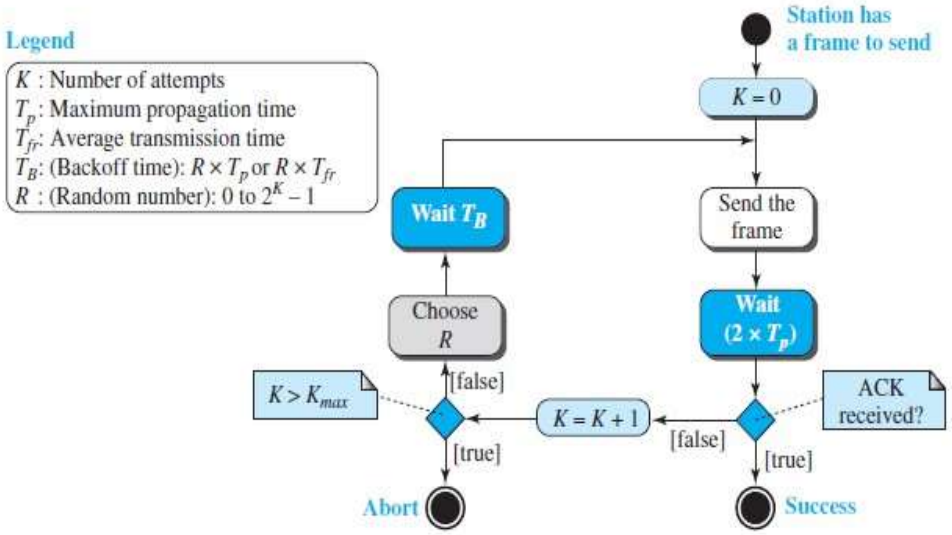


Fig: Procedure for pure ALOHA protocol

- ***Slotted ALOHA***

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 shows an example of frame collisions in slotted ALOHA.

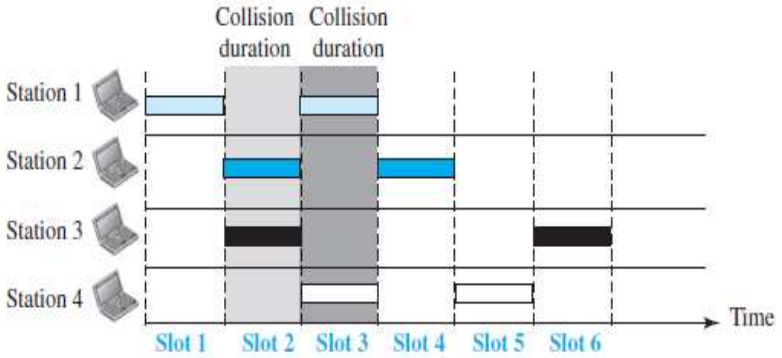


Fig: Frames in a slotted ALOHA network

Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame.

2. 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".

CSMA can reduce the possibility of collision, but it cannot eliminate it. The reason for this is shown in Figure, a space and time model of a CSMA network. Stations are connected to a shared channel.

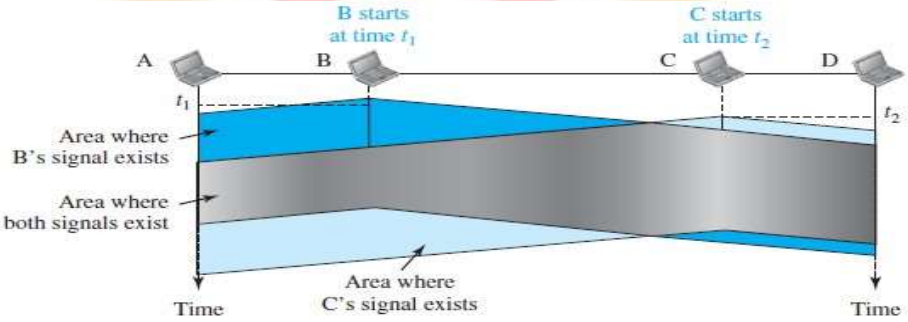


Fig: Space/time model of a collision in CSMA

The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time for the first bit to reach every station and for every station to sense it.

➢ ***Persistence Methods:*** **Three methods** have been devised to answer these questions: the **1-persistent method,** the **nonpersistent method,** and the ***p*-persistent method.** Figure shows the behavior of three persistence methods when a station finds a channel busy.

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

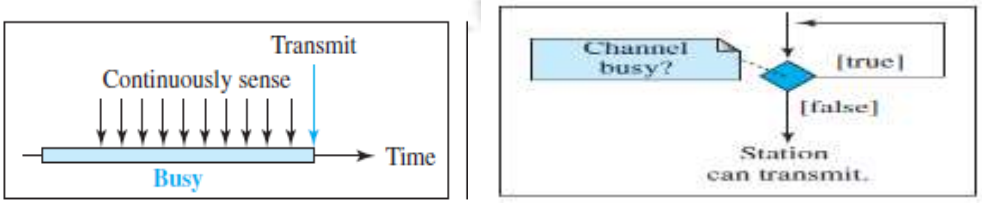


Fig: 1-Persistent

Nonpersistent: In the *nonpersistent method*, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of

time and then senses the line again. The nonpersistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously

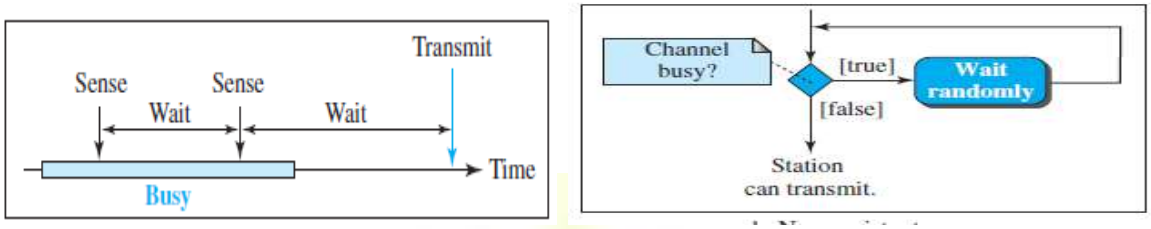


Fig: Nonpersistent

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

1. With probability *p*, the station sends its frame.

2. With probability $q = 1 - p$, the station waits for the beginning of the next time slot and checks the line again.

a. If the line is idle, it goes to step 1.

b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

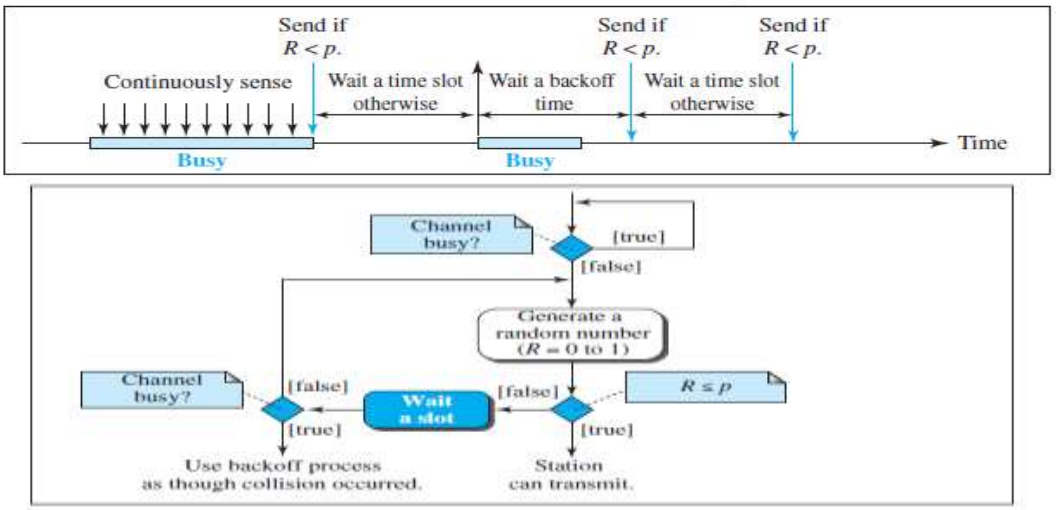


Fig: *p*-Persistent

3. 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, stations A and C are involved in the collision.

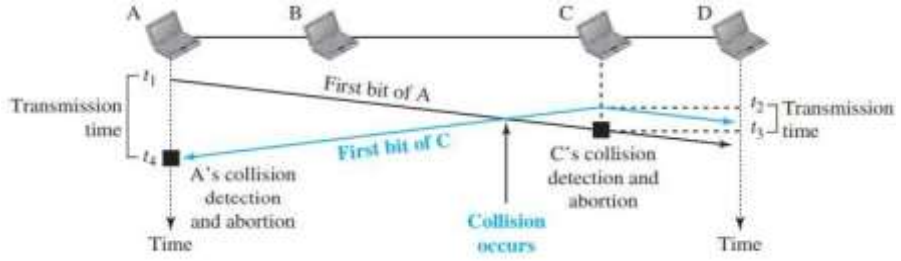


Fig: Collision of the first bits in CSMA/CD

Procedure: Now let us look at the flow diagram for CSMA/CD in Figure.

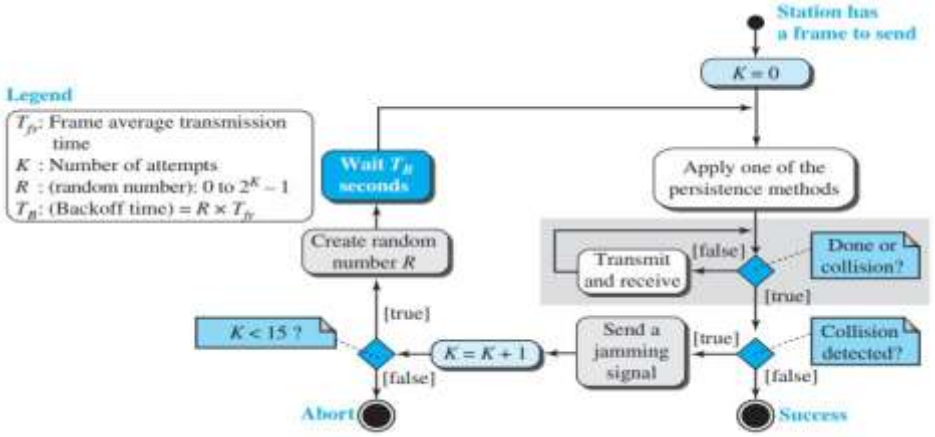


Fig: Flow diagram for the CSMA/CD

The sending of a short jamming signal to make sure that all other stations become aware of the collision.

4. 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.

Interframe Space (IFS). First, collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the ***interframe*** space or ***IFS.***

Contention Window. The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential backoff strategy.

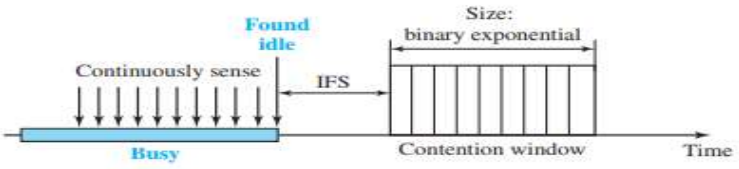


Fig: Contention window

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.

Procedure:

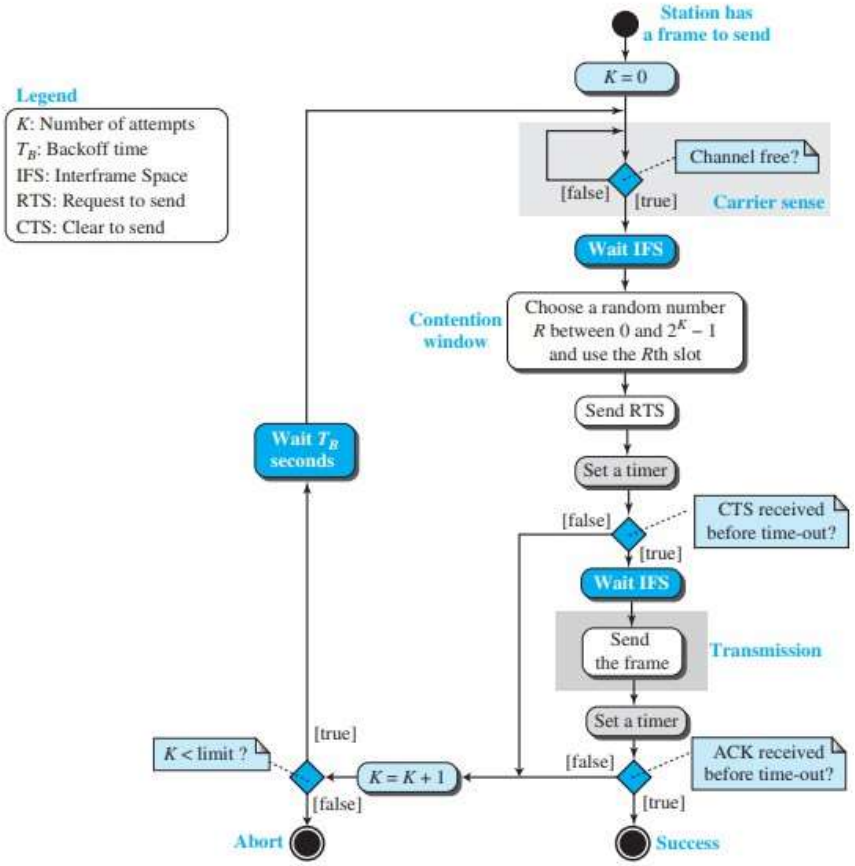


Fig: Flow diagram of CSMA/CA

5. IEEE standards

The IEEE standard that encounter in real life. 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.

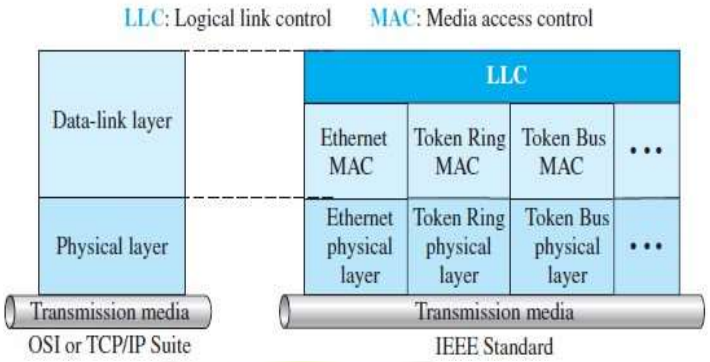


Fig:*IEEE standard for LANs*

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

Logical Link Control (LLC)

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

Media Access Control (MAC)

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

6. Standard Ethernet (802.3): 10 Mbps

The Ethernet LAN was developed in the 1970s. 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.

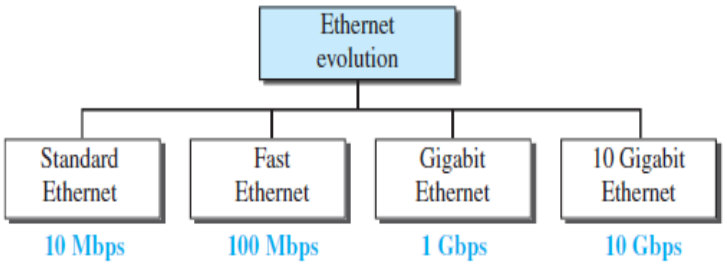


Fig: Ethernet evolution through four generations

We refer to the original Ethernet technology with the data rate of 10 Mbps as the *Standard Ethernet.*

❖ **MAC Sublayer**

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

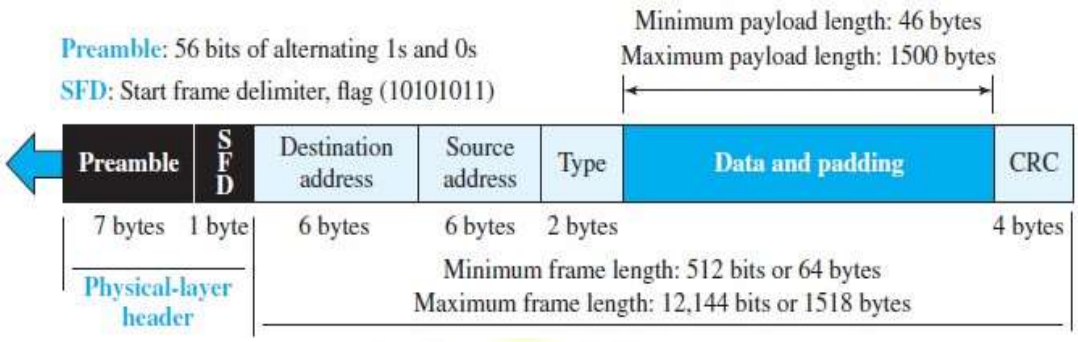


Fig: Ethernet frame

Preamble. This field contains 7 bytes (56 bits) of alternating 0s and 1s that alert the receiving system to the coming frame and enable it to synchronize its clock if it's out synchronization. The pattern provides only an alert and a timing pulse.

Start frame delimiter (SFD). This field (1 byte: 10101011) signals the beginning of the frame. The last 2 bits are $(11)_2$ and alert the receiver that the next field is the destination address. This field is actually a flag that defines the beginning of the frame.

Destination address (DA). This field is six bytes (48 bits) and contains the linklayer address of the destination station or stations to receive the packet.

Source address (SA). This field is also six bytes and contains the link-layer address of the sender of the packet.

Type. This field defines the upper-layer protocol whose packet is encapsulated in the frame. This protocol can be IP, ARP, OSPF, and so on.

Data. This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes.
CRC. The last field contains error detection information, in this case a CRC-32. The CRC is calculated over the addresses, types, and data field. If the receiver calculates the CRC and finds that it is not zero (corruption in transmission), it discards the frame.

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

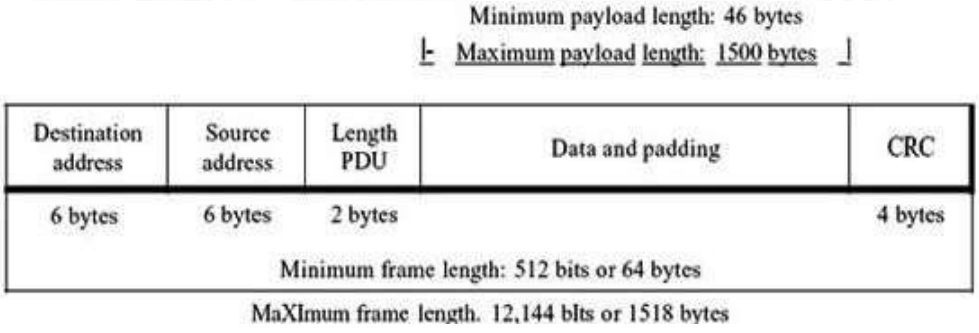


Fig: minimum and maximum lengths of a frame

- **Addressing:** Each station on an Ethernet network has its own **network interface card (NIC).** The NIC fits inside the station and provides the station with a link-layer address.

The Ethernet address is ***6 bytes (48 bits)***, normally written in **hexadecimal notation,** with a colon between the bytes.
For example, the following shows an Ethernet MAC address: **4A:30:10:21:10:1A**

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

❖ **Physical layer**

Implementation: The Standard Ethernet defined several implementations.

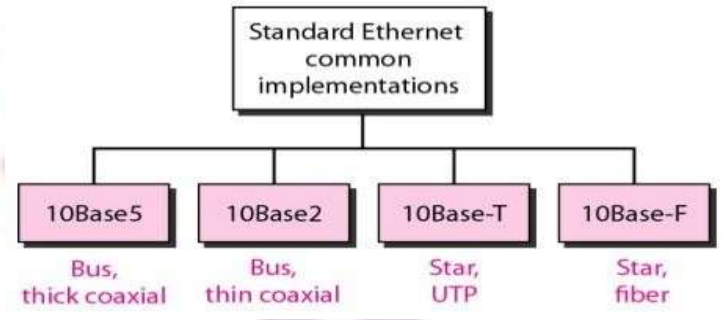


Table shows a summary of Standard Ethernet implementations.

Implementation	*Medium*	*Medium Length*	*Encoding*
10Base5	Thick coax	500 m	Manchester
10Base2	Thin coax	185 m	Manchester
10Base-T	2 UTP	100 m	Manchester
10Base-F	2 Fiber	2000 m	Manchester

10Base5: Thick Ethernet

The first implementation is called ***10Base5, thick Ethernet,*** or ***Thicknet.*** 10Base5 was the first Ethernet specification to use a bus topology with an external **transceiver** connected via a tap to a thick coaxial cable. The maximum length of the coaxial cable must not exceed 500 m, otherwise, there is excessive degradation of the signal.

7. Fast Ethernet (100 Mbps): 802.3u

Ethernet made a big jump by increasing the transmission rate to 100 Mbps, and the new generation was called the *Fast Ethernet*.
The goals of Fast Ethernet can be summarized as follows:
1. Upgrade the data rate to 100 Mbps.
2. Make it compatible with Standard Ethernet.
3. Keep the same 48-bit address.
4. Keep the same frame format.
5. Keep the same minimum and maximum frame lengths.

❖ MAC Sublayer

A decision was made to drop the bus topologies and keep only the star topology. For the star topology, there are two choices: half duplex and full duplex.
The access method is the same *(CSMAlCD)* for the half-duplex approach; for full duplex Fast Ethernet, there is no need for *CSMAlCD.*

❖ Physical layer

Implementation: Fast Ethernet implementation at the physical layer can be categorized as either two-wire or fourwire.
The two-wire implementation can be either category 5 UTP (lOOBase-TX) or fiber-optic cable (lOOBase-FX). The four-wire implementation is designed only for category 3 UTP (l00Base-T4).

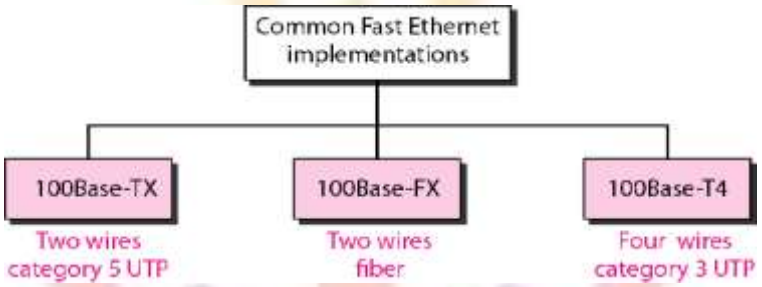


Table shows a summary of Fast Ethernet implementations.

Characteristics	*100Base-TX*	*100Base-FX*	*100Base-T4*
Media	Cat 5 UTP or STP	Fiber	Cat 4 UTP
Number of wires	2	2	4
Maximum length	100 m	100 m	100 m
Block encoding	4B/5B	4B/5B	
Line encoding	MLT-3	NRZ-I	8B/6T

11. IEEE 802.11 (*Wi-Fi)*

IEEE has defined the specifications for a wireless LAN, called IEEE 802.11, which covers the physical and data-link layers. It is sometimes called ***wireless Ethernet***.
In some countries, including the United States, the public uses the term ***Wi-Fi*** (short for wireless fidelity) as a synonym for *wireless LAN*.

❖ Architecture

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

Basic Service Set

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

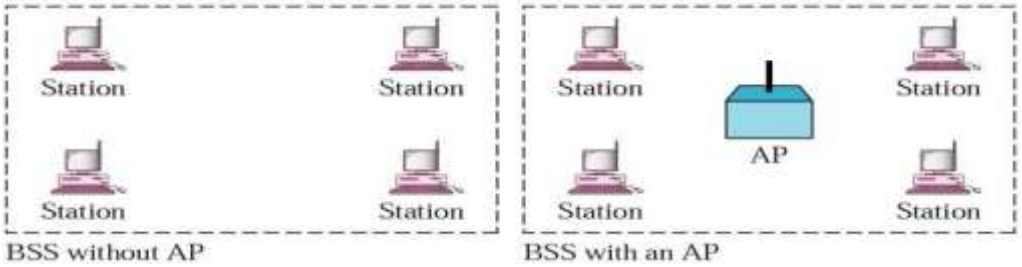


Fig: Basic service sets (BSSs)

The BSS without an AP is called an ***ad hoc architecture***. A BSS with an AP is sometimes referred to as an ***infrastructure BSS***.

Extended Service Set

An **extended service set (ESS)** is made up of two or more BSSs with APs. In this case, the BSSs are connected through a *distribution system*, which is a wired or a wireless network. The distribution system connects the APs in the BSSs.

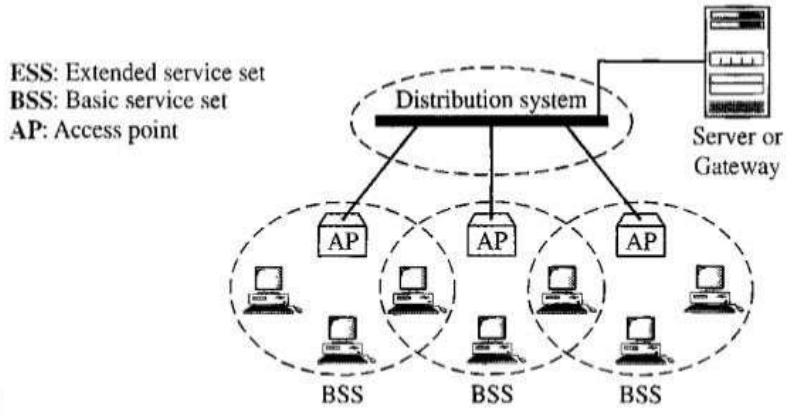


Fig: Extended service set (ESS)

❖ **MAC Sublayer:**

Frame Format

The MAC layer frame consists of nine fields, as shown in Figure.

Fig: Frame format

Frame control (FC). The FC field is 2 bytes long and defines the type of frame and some control information.

D. This field defines the duration of the transmission that is used to set the value of NAV. In one control frame, it defines the ID of the frame.

Addresses. There are four address fields, each 6 bytes long. The meaning of each address field depends on the value of the *To DS* and *From DS* subfields.

Sequence control. This field defines the sequence number, which is the same in all fragments.

Frame body. This field, which can be between 0 and 2312 bytes, contains information based on the type and the subtype defined in the FC field.
FCS. The FCS field is 4 bytes long and contains a CRC-32 error-detection sequence.

❖ **Physical Layer:**

IEEE	*Technique*	*Band*	*Modulation*	*Rate (Mbps)*
802.11	FHSS	2.400–4.835 GHz	FSK	1 and 2
	DSSS	2.400–4.835 GHz	PSK	1 and 2
	None	Infrared	PPM	1 and 2
802.11a	OFDM	5.725–5.850 GHz	PSK or QAM	6 to 54
802.11b	DSSS	2.400–4.835 GHz	PSK	5.5 and 11
802.11g	OFDM	2.400–4.835 GHz	Different	22 and 54

- ***IEEE 802.11 FHSS:*** IEEE 802.11 FHSS uses the **frequency-hopping spread spectrum (FHSS)** method. FHSS uses the 2.400–4.835 GHz ISM band. The band is divided into 79 sub bands of 1 MHz. The modulation technique in this specification FSK (frequency shift keying), which results in a data rate of 1 or 2 Mbps.
- ***IEEE 802.11 DSSS:*** IEEE 802.11 DSSS uses the **direct-sequence spread spectrum (DSSS)** method. DSSS uses the 2.400–4.835 GHz ISM band. The modulation technique in this specification is PSK (phase shift keying), which results in a data rate of 1 or 2 Mbps.
- ***IEEE 802.11 Infrared:*** IEEE 802.11 infrared uses infrared light in the range of 800 to 950 nm. The modulation technique is called ***pulse position modulation (PPM).***

- ***IEEE 802.11a OFDM:*** IEEE 802.11a OFDM describes the **orthogonal frequency-division multiplexing (OFDM)** method for signal generation in a 5.725–5.850 GHz ISM band. OFDM uses PSK and QAM for modulation. The common data rates are 6 Mbps (PSK) and 54 Mbps (QAM).

- ***IEEE 802.11b DSSS:*** IEEE 802.11b DSSS describes the **high-rate direct-sequence spread spectrum (HRDSSS)** method for signal generation in the 2.400–4.835 GHz ISM band. HR-DSSS is similar to DSSS except for the encoding method, which is called ***complementary code keying (CCK).***

- ***IEEE 802.11g:*** This new specification defines forward error correction and OFDM using the 2.400– 4.835 GHz ISM band. The modulation technique achieves a 22- or 54-Mbps data rate.

8. Controlled Access

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

❖ **Reservation**:

- In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals.

- In each interval, a reservation frame precedes the data frames sent in that interval. If there are N stations in the system, there are exactly N reservation minislots in the reservation frame. Each minislot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own minislot.
- The stations that have made reservations can send their data frames after the reservation frame. Figure shows a situation with five stations and a five-minislot reservation frame.
- In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.

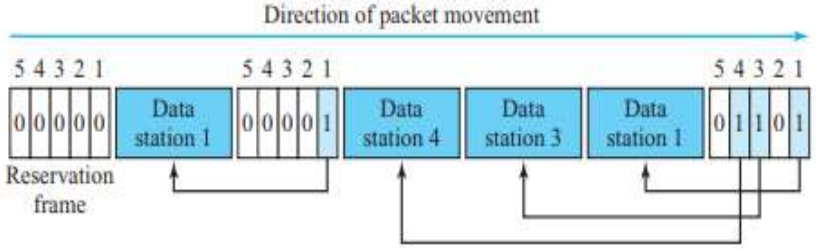


Fig: Reservation access method

❖ Polling

- Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations.
- All data exchanges must be made through the primary device even when the ultimate destination is a secondary device.
- The primary device controls the link; the secondary devices follow its instructions. It is up to the primary device to determine which device is allowed to use the channel at a given time.
- This method uses ***poll*** and ***select functions*** to prevent collisions.

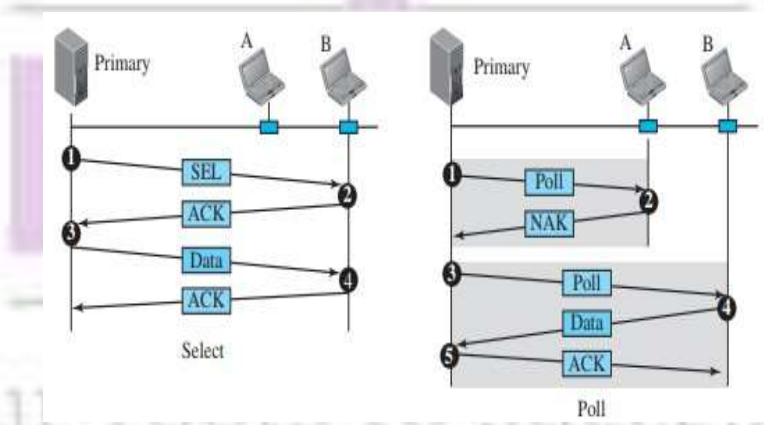


Fig: Select and poll functions in polling-access method

Select: The select function is used whenever the primary device has something to send. Remember that the primary controls the link. If the primary is neither sending nor receiving data, it knows the link is available.

- The primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status.

- Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address of the intended secondary.

Poll: The poll function is used by the primary device to request transmissions from the secondary devices.

- When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send.
- When the first secondary is approached, it responds either with a NAK frame if it has nothing to send or with data (in the form of a data frame) if it does.
- If the response is negative (a NAK frame), then the primary polls the next secondary in the same manner until it finds one with data to send.
- When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (ACK frame), verifying its receipt.

❖ Token Passing

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

But how is the right to access the channel passed from one station to another? In this method, a special packet called a ***token*** circulates through the ring.

Token management is needed for this access method. Stations must be limited in the time they can have possession of the token. The token must be monitored to ensure it has not been lost or destroyed.
Another function of token management is to assign priorities to the stations and to the types of data being transmitted. And finally, token management is needed to make low-priority stations release the token to high-priority stations.

Logical Ring: In a token-passing network, stations do not have to be physically connected in a ring; the ring can be a logical one. Figure shows ***four*** different physical topologies that can create a logical ring.

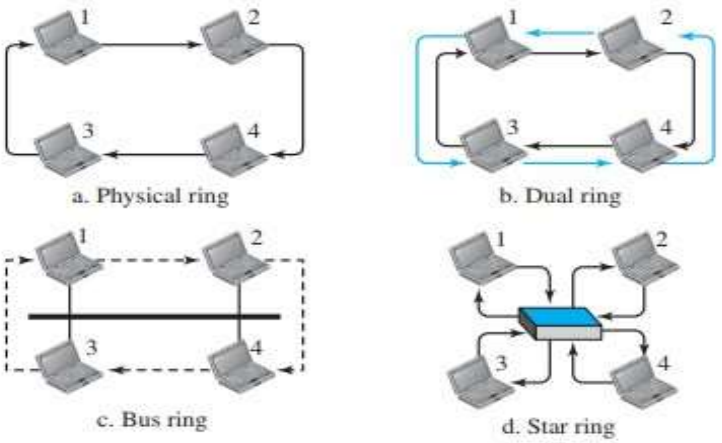


Fig: Logical ring and physical topology in token-passing access method

9. Channelization

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

❖ FDMA

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

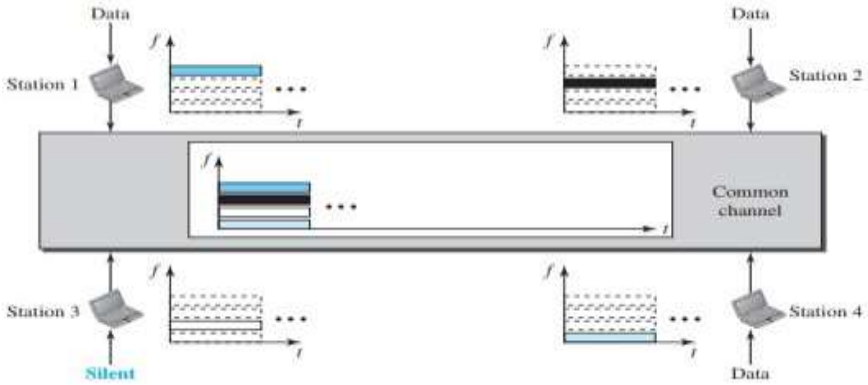


Fig: Frequency-division multiple access (FDMA)

❖ TDMA

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

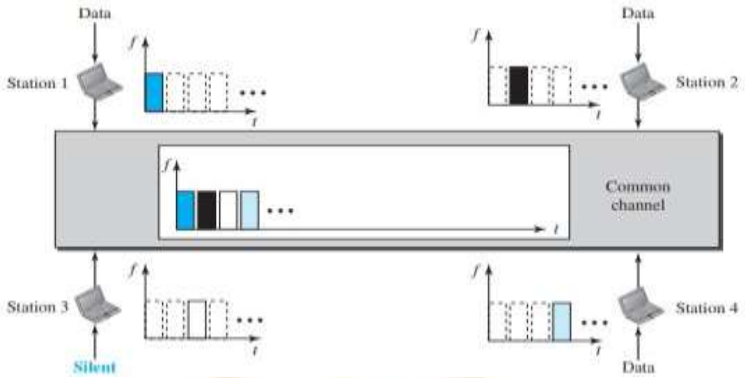


Fig: Time-division multiple access (TDMA)

❖ CDMA

Code-division multiple access (CDMA) was conceived several decades ago. In CDMA, one channel carries all transmissions simultaneously.

In CDMA, the stations use different codes to achieve multiple accesses. CDMA is based on coding theory and uses sequence of numbers called chips.

Let us assume we have four stations, 1, 2, 3, and 4, connected to the same channel. The data from station 1 are d1, from station 2 are d2, and so on. The code assigned to the first station is c1, to the second is c2, and so on.

We assume that the assigned codes have two properties.

1. If we multiply each code by another, we get 0.
2. If we multiply each code by itself, we get 4 (the number of stations).

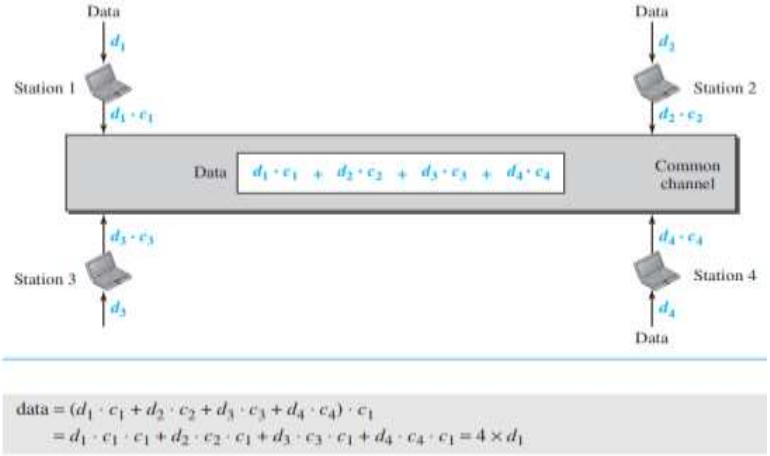


Fig: Simple idea of communication with code

your tools to success...