Data link layer

Main functionalities of Data Link Layer in Short

1. Node to node communication. (hop to hop delivery): data link layer is not responsible for final node (destination node).

2. Data link layer is sufficient to make connection between two devices in the same network. (it work maximum within the network)

3. **Flow control**: Speed of message (Stop N Wait, Go-Back-N, Selective Repeat) but node to node, this method is applicable in transport layer but it is between source to destination.

4. **Error control**: node to node error control and methods are (CRC, Checksum) but checksum method is for transport layer method for error control.

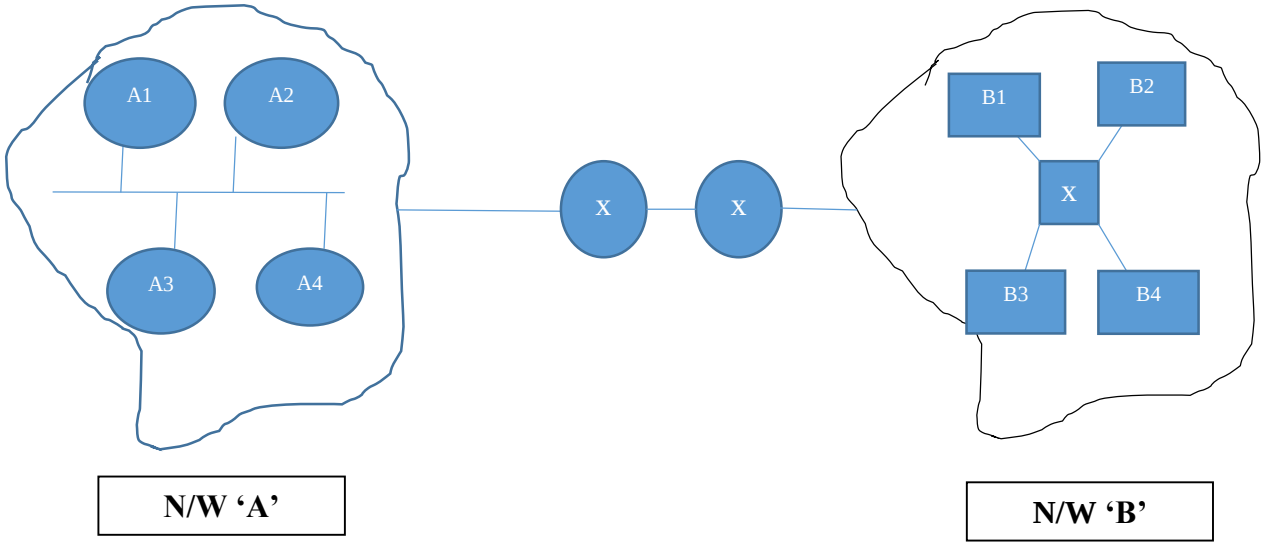
(Note: Error control in data link layer is more efficient because in every node it is happened.)

5. **Access Control**: Methods are (Pure Aloha, Slotted Aloha, Token Ring, CSMA/CD (Carrier Sense Multiple Access/Collision Detection)).

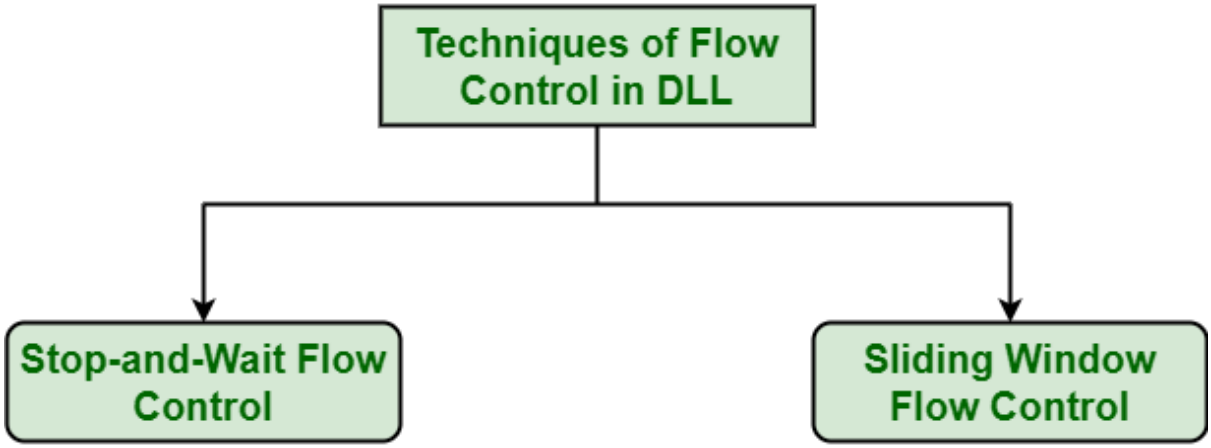
6. **Physical address (MAC)/ NIC Card**: fixed address for all the nodes.

7. **Framing**: for security data is transferred in the form of frame inside that frame header and tailor is added with the data) but it is between hop to hop.

We can learn through the diagram:



Flow Control Methods



1. Stop-and-wait ARQ Protocol

1. It is the simplest flow control method. In this, the sender will transmit one frame at a time to the receiver. The sender will stop and wait for the acknowledgement from the receiver.

2. This time (i.e. the time joining message transmitting and acknowledgement receiving) is the sender's waiting time, and the sender is idle during this time.

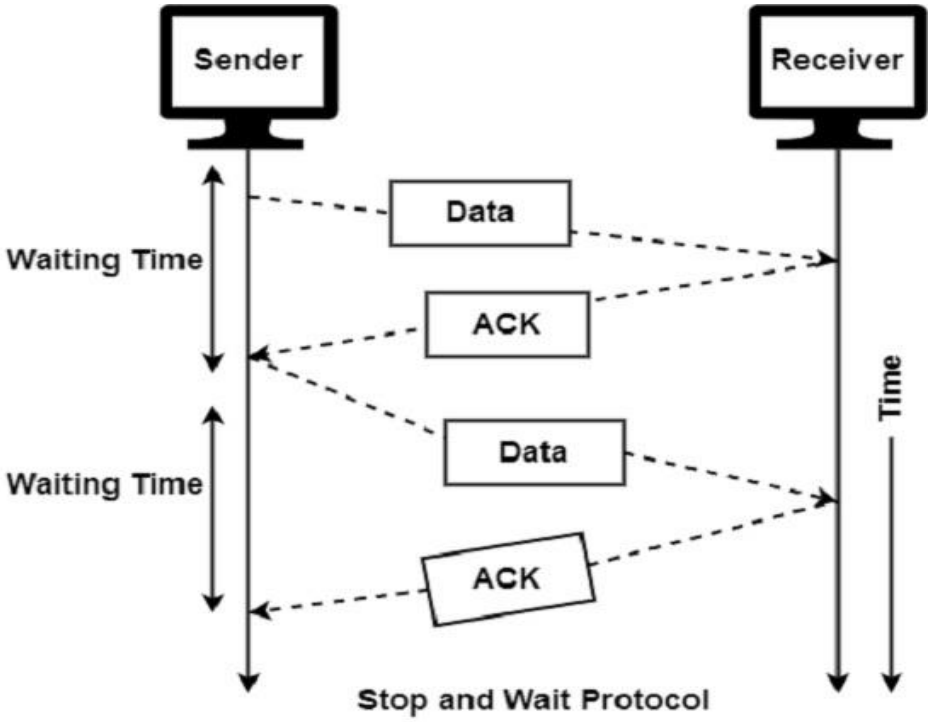
3. When the sender gets the acknowledgement (ACK), it will send the next data packet to the receiver and wait for the disclosure again, and this process will continue as long as the sender has the data to send.

4. While sending the data from the sender to the receiver, the data flow needs to be controlled.

5. If the sender is transmitting the data at a rate higher than the receiver can receive and process it, the data will get lost.

6. The Flow-control methods will help in ensuring that the data doesn't get lost. The flow control method will check that the senders send the data only at a rate that the receiver can receive and process.

The working of Stop and Wait Protocol is shown in the figure below –



Advantage: The main advantage of stop & wait protocols is their accuracy. The next frame is transmitted only when the first frame is acknowledged. So there is no chance of the frame being lost.

Drawback: It makes the transmission process slow. An individual frame travels from source to destination in this method, and a single acknowledgement travels from destination to source. As a result, each frame sent and received uses the entire time needed to traverse the link. Moreover, if two devices are a distance apart, a lot of time is wasted waiting for ACKs leading to an increase in total transmission time.

Features

The features of Stop and Wait Protocol are as follows −

- It is used in Connection-oriented communication.
- It uses sequence number to get the proper sequence of data.
- It uses a link between sender and receiver as a half-duplex link
- It offers error and flows control.
- It can be used in data Link and transport Layers.
- Stop and Wait ARQ executes Sliding Window Protocol with Window Size 1.

Failure Cases

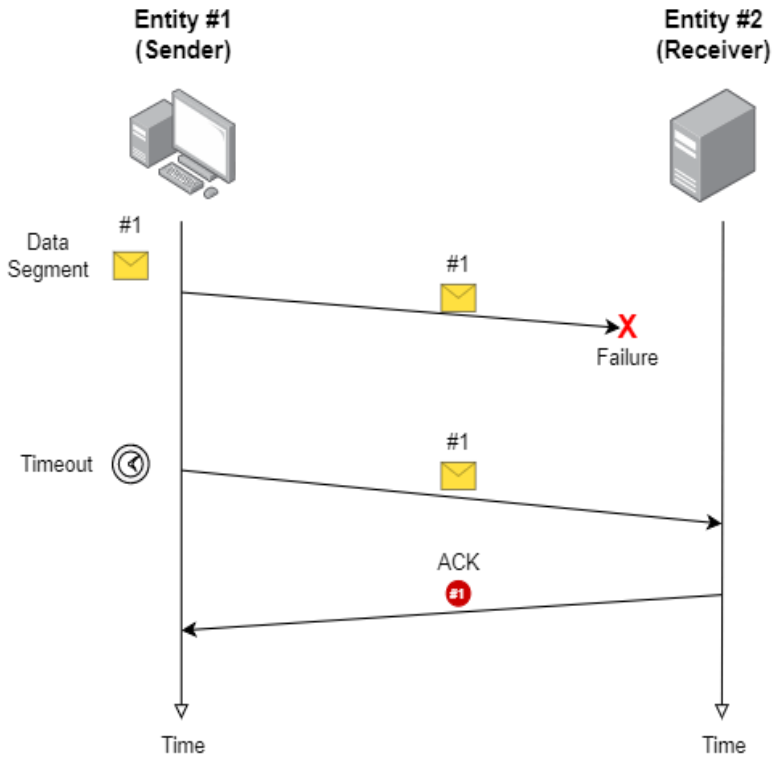
The stop-and-wait mechanism infers that a transmission failed based on timeouts timer (which is 2xroundtrip). We have three main message-transmitting failure types:

(i) not receiving a data segment in the destination;

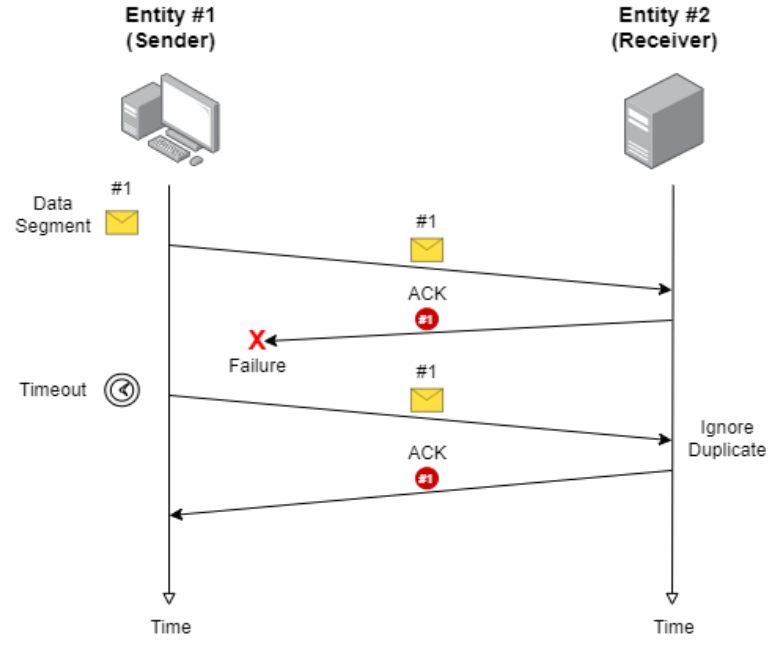
(ii) not receiving the ACK message in the source, and

(iii) receiving an ACK message after the timeout event.

If a data segment does not reach the destination, a timeout event will occur in the sender entity, thus indicating that the ACK message never arrived. In such a case, this entity will send the message again. We can see this scenario in the image next:

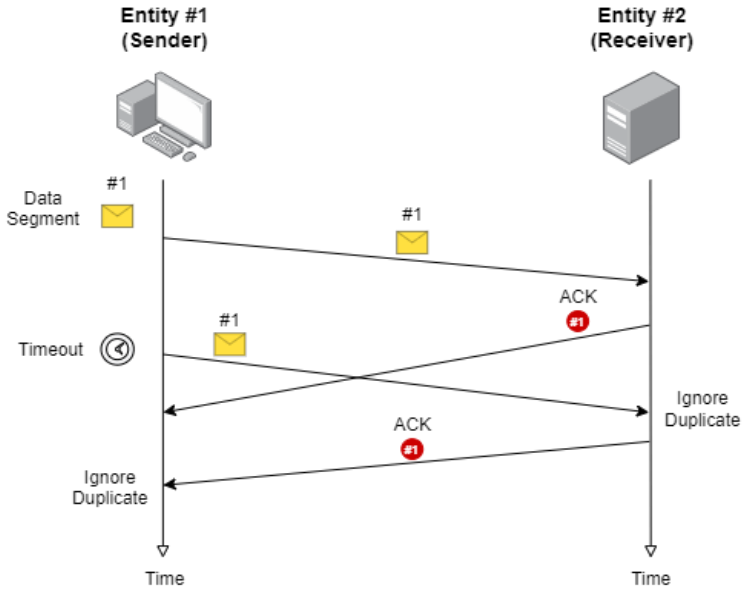


A second case is when the data segment reaches the destination, but there is a failure in the ACK message transmission. So, a timeout occurs on the sender side, which retransmits the data segment. Thus, the receiver will get a duplicate segment, ignore it, and send the corresponding ack message again. We can check the described scenario in the following image:



Finally, the third scenario we'll study is when the ACK messages arrive after the timeout. In such a scenario, the sender retransmits the data segment, receiving the ack from the first transmission after that. At this point, the sender entity can already send the following segment. The other entity will receive a duplicate packet, ignore it, and send a new ACK. So, the sender will receive a duplicate ACK message, ignoring it.

The image next shows the last message-transmitting failure scenario:



It is important to note that potential problems related to data segment transmission can happen. For example, when the receiver entity gets different segments with the same sequence number in the second or third scenario, caused due to some interference in the transmission. We can use other strategies to detect and correct data errors in these scenarios.

2. Sliding Window Protocol:

Sliding window protocols are data link layer protocols for reliable and sequential delivery of data frames. The sliding window is also used in Transmission Control Protocol.

In this protocol, multiple frames can be sent by a sender at a time before receiving an acknowledgment from the receiver. The term sliding window refers to the imaginary boxes to hold frames. Sliding window method is also known as windowing.

Working Principle

In these protocols, the sender has a buffer called the sending window and the receiver has buffer called the receiving window.

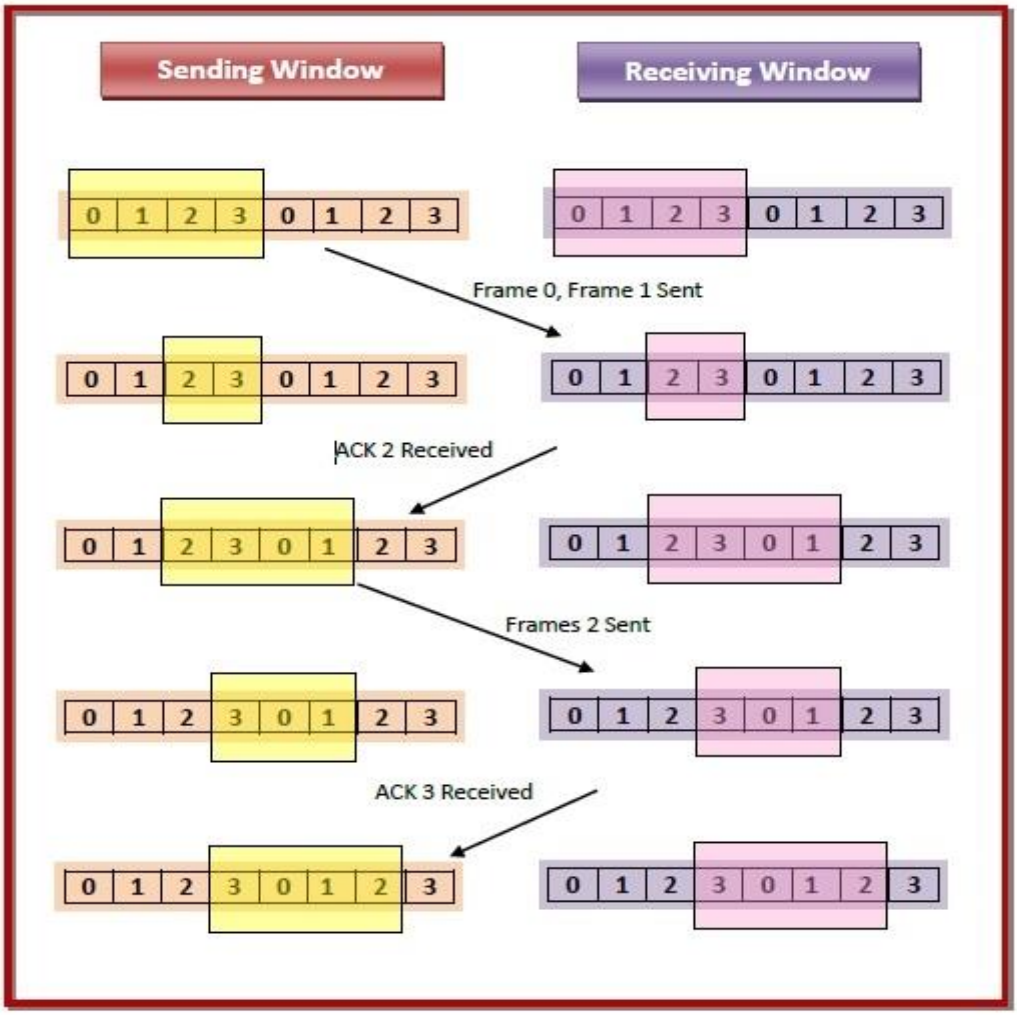
The size of the sending window determines the sequence number of the outbound frames. If the sequence number of the frames is an n-bit field, then the range of sequence numbers that can be assigned is 0 to 2^n-1. Consequently, the size of the sending window is 2^n-1. Thus in order to accommodate a sending window size of 2^n-1, a n-bit sequence number is chosen.

The sequence numbers are numbered as modulo-n. For example, if the sending window size is 4, then the sequence numbers will be 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, and so on. The number of bits in the sequence number is 2 to generate the binary sequence 00, 01, 10, 11.

The size of the receiving window is the maximum number of frames that the receiver can accept at a time. It determines the maximum number of frames that the sender can send before receiving acknowledgment.

Example

Suppose that we have sender window and receiver window each of size 4. So the sequence numbering of both the windows will be 0,1,2,3,0,1,2 and so on. The following diagram shows the positions of the windows after sending the frames and receiving acknowledgments.



Types of Sliding Window Protocol:

2.1. Go-Back-N ARQ Protocol

1. In the Go-Back-N protocol, the sequence numbers are modulo 2^n, where n is the size of the sequence number field in bits.
2. Sender window size = 2^n-1.
3. Receiver window size = 1.
4. It will not accept out of order packet.
5. It does not consider the corrupted frames and simply discards them.
6. If the sender does not receive the acknowledgment, it leads to the retransmission of all the current window frames.

Working Principle of Go-Back-N protocol

1. Go – Back – N ARQ uses the concept of protocol pipelining, i.e. sending multiple frames before receiving the acknowledgment for the first frame.

2. The frames are sequentially numbered and a finite number of frames. The maximum number of frames that can be sent depends upon the size of the sending window.

3. If the acknowledgment of a frame is not received within an agreed upon time period, all frames starting from that frame are retransmitted.

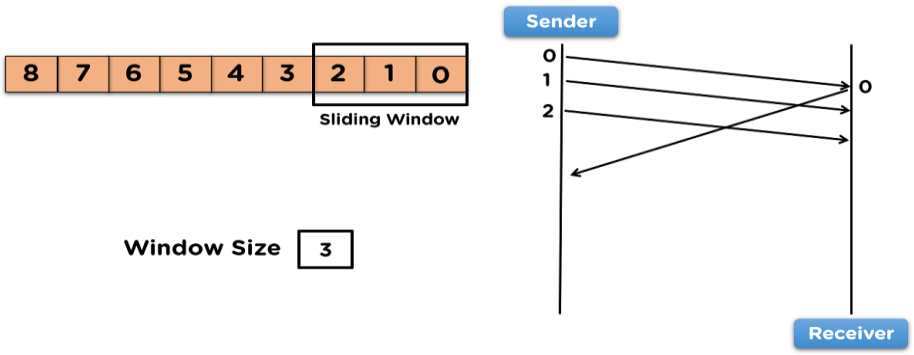
4. The size of the sending window determines the sequence number of the outbound frames. If the sequence number of the frames is an n-bit field, then the range of sequence numbers that can be assigned is 0 to 2^n-1. Consequently, the size of the sending window is 2^n-1. Thus in order to accommodate a sending window size of 2^n-1, an n-bit sequence number is chosen.

5. The sequence numbers are numbered as modulo-n. For example, if the sending window size is 4, then the sequence numbers will be 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, and so on. The number of bits in the sequence number is 2 to generate the binary sequence 00, 01, 10, 11.

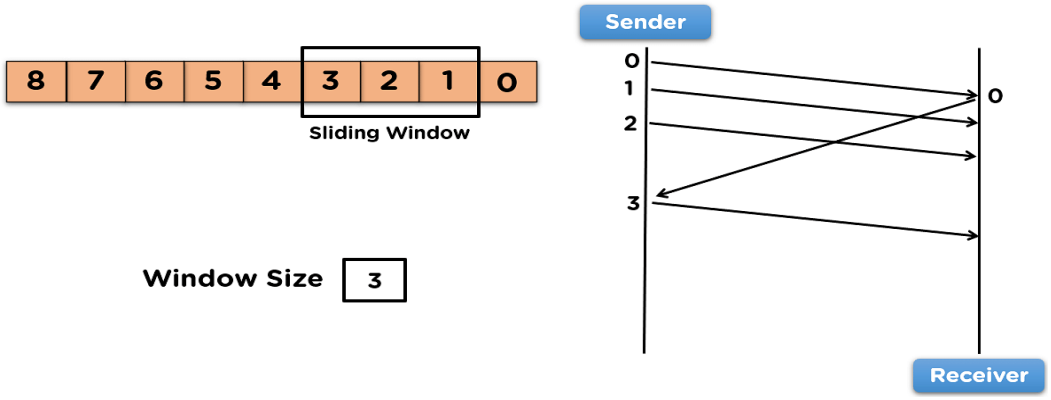
6. The size of the receiving window is 1.

Example:

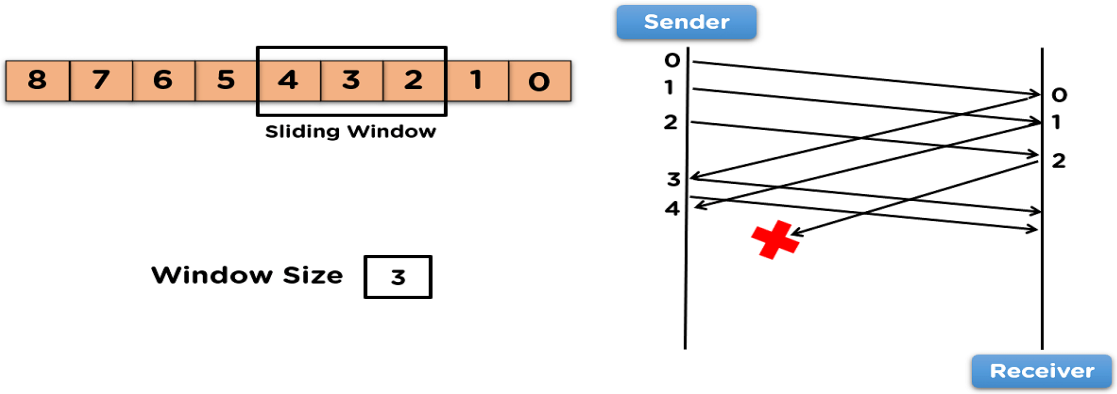
The working of the Go-Back-N ARQ protocol involves applying the sliding window method for the basis of sharing data, and the number of frames to be shared is decided by the window size. Then using the main points we discussed and the mentioned features, let's discuss the steps involved in the working of the protocol:



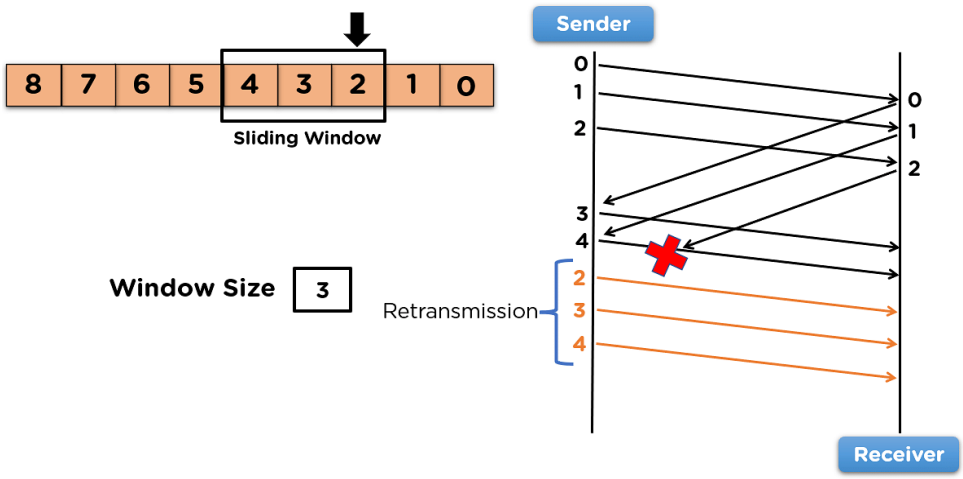
1. To begin with, the sender side will share the data frames simultaneously according to the window size assigned, over to the receiver side, and wait for the acknowledgment.
2. After the receiver side receives the frames, it will use the first frame and send the acknowledgment to the sender side.



3. After the sender receive's the acknowledgment for the first frame, the sender will share the next frame with the receiver.



4. This exchange continues until, due to some external or internal interruption in the network, the acknowledgment is not received by the sender side.



5. Then, the sender side will go back to the unacknowledged frame and retransmit that frame, along with all the frames shared after that frame with the receiver. This represents the Go-Back-N ARQ protocol method.

Let’s move on to some advantages and disadvantages of applying the Go-Back-N ARQ protocol in the network.

Advantages:

- Multiple frames can be simultaneous to the receiver side.
- Increase the efficiency of the data transfer and has more control over the flow of frames.
- Time delay is less for sharing data frames.

Disadvantages:

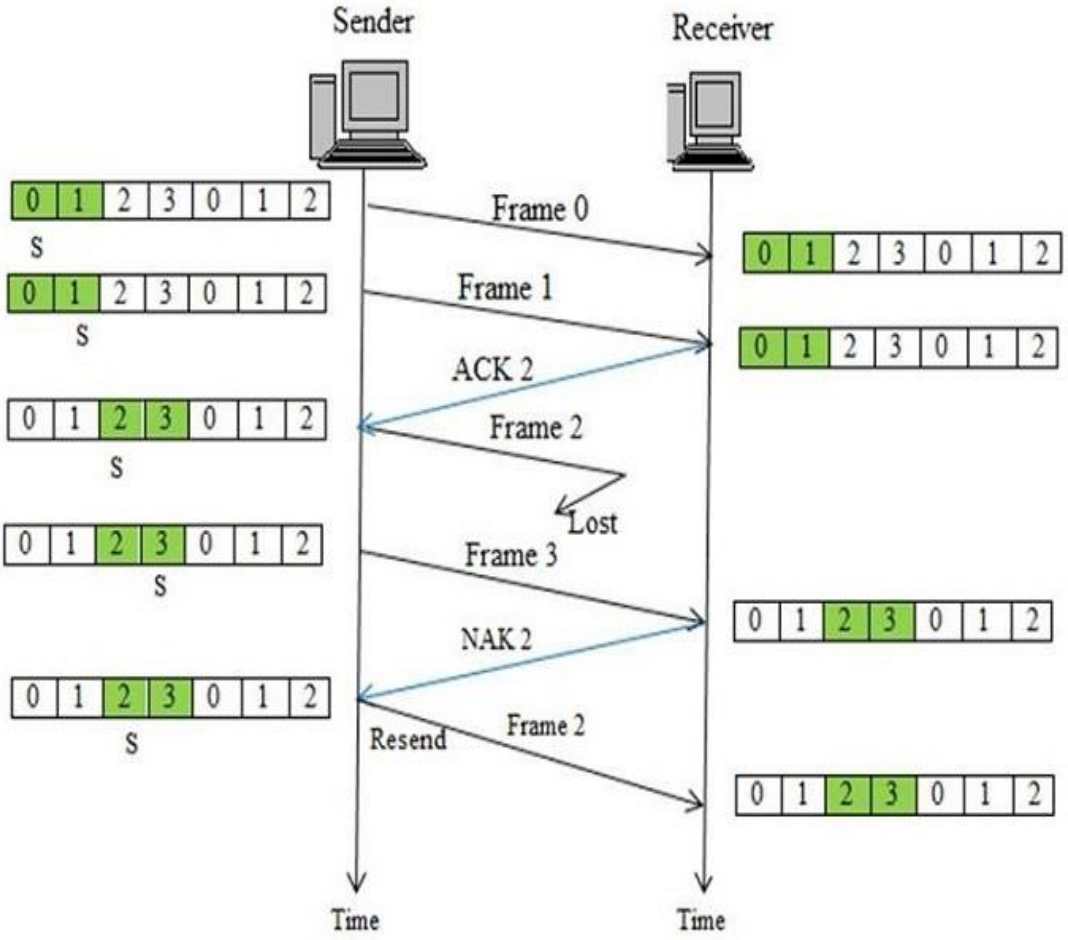
- The storage of data frames at the receiver side.
- Retransmission of frames, when the acknowledgement is not received by the sender end.

2.2. Selective Repeat Protocol

1. Sender and receiver window size is 2^{n-1}.
2. if n=3, means 8 sequence number (0-7).

Working Principle of Selective Repeat Protocol:
The selective repeat protocol is an implementation of the sliding window protocol. In the selective repeat protocol, both the sender and the receiver maintain a window of outstanding and acceptable sequence numbers.

- In SRP, the sender's window size starts at 0 and it grows to some predefined maximum.
- The receiver's window is always fixed in size and equal to the predetermined maximum.
- The receiver has the buffer reserved for each sequence number within its fixed window.
- The sender and the receiver maintain a buffer of their window size.
- If there is an error, the receiver checks the lower edge to the last sequence number before the lost frame sequence number.
- The receiver continues to receive and acknowledge incoming frames.
- The sender maintains a timeout clock for the unacknowledged frame number and retransmits that frame after the timeout.
- The acknowledgment will be piggybacked to the sender. But when there is no traffic in the reverse direction, piggyback is impossible, a special timer will time out for the ACK so that the ACK is sent back as an independent packet. If the receiver suspects that the transmission has an error, it immediately sends back a negative acknowledgment (NAK) to the sender.



Note − SRP works better when the link is very unreliable. Because in this case, retransmission tends to happen more frequently, selectively retransmitting frames is more efficient than retransmitting all of them. In selective repeat protocol, the size of the sender and receiver windows must be at most one-half of 2^n.

3. Error detection and correction:

1. Error Detection Techniques:

1.1. Simple parity check.
1.2. Checksum.
1.3. CRC (Cyclic Redundancy Check).

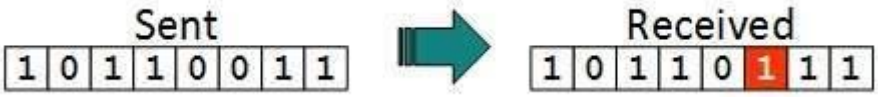
2. Error Correction Techniques:

2.1. Hamming codes.
2.2. Binary convolutional codes.
2.3. Reed-Solomon codes.
2.4. Low-Density Parity Check codes.

Types of Errors

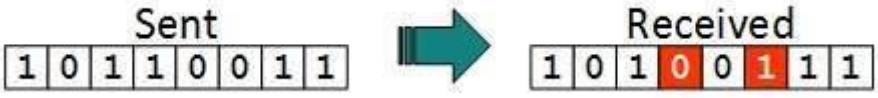
There may be three types of errors:

- **Single bit error**



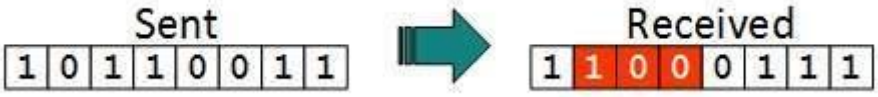
In a frame, there is only one bit, anywhere though, which is corrupt.

- **Multiple bits error**



Frame is received with more than one bits in corrupted state.

- **Burst error**



Frame contains more than1 consecutive bits corrupted.

1. Error Detection Techniques: Errors in the received frames are detected by means of Parity Check and Cyclic Redundancy Check (CRC). In both cases, few extra bits are sent along with actual data to confirm that bits received at other end are same as they were sent. If the counter-check at receiver' end fails, the bits are considered corrupted.

1.1 Parity Check

One extra bit is sent along with the original bits to make number of 1s either even in case of even parity, or odd in case of odd parity. The sender while creating a frame counts the number of 1s in it. For example, if even parity is used and number of 1s is even then one bit with value

0 is added. This way number of 1s remains even.If the number of 1s is odd, to make it even a bit with value 1 is added.

The receiver simply counts the number of 1s in a frame. If the count of 1s is even and even parity is used, the frame is considered to be not-corrupted and is accepted. If the count of 1s is odd and odd parity is used, the frame is still not corrupted. If a single bit flips in transit, the receiver can detect it by counting the number of 1s. But when more than one bits are erro neous, then it is very hard for the receiver to detect the error.

1.2 Checksum: Checksum is an error detection method. Error detection using checksum method involves the following steps-

Step-01:

At sender side,

- If m bit checksum is used, the data unit to be transmitted is divided into segments of m bits.
- All the m bit segments are added.
- The result of the sum is then complemented using 1's complement arithmetic.
- The value so obtained is called as **checksum**.

Step-02:

- The data along with the checksum value is transmitted to the receiver.

Step-03:

At receiver side,

- If m bit checksum is being used, the received data unit is divided into segments of m bits.
- All the m bit segments are added along with the checksum value.
- The value so obtained is complemented and the result is checked.

Then, following two cases are possible-

Case-01: Result = 0

If the result is zero,

- Receiver assumes that no error occurred in the data during the transmission.
- Receiver accepts the data.

Case-02: Result ≠ 0

If the result is non-zero,

- Receiver assumes that error occurred in the data during the transmission.
- Receiver discards the data and asks the sender for retransmission.

Checksum Example-

Consider the data unit to be transmitted is-

10011001111000100010010010000100

Consider 8 bit checksum is used.

Step-01:

At sender side,

The given data unit is divided into segments of 8 bits as-

10011001	11100010	00100100	10000100

Now, all the segments are added and the result is obtained as-

- 10011001 + 11100010 + 00100100 + 10000100 = 1000100011
- Since the result consists of 10 bits, so extra 2 bits are wrapped around.
- 00100011 + 10 = 00100101 (8 bits)
- Now, 1's complement is taken which is 11011010.
- Thus, checksum value = 11011010

Step-02:

- The data along with the checksum value is transmitted to the receiver.

Step-03:

At receiver side,

- The received data unit is divided into segments of 8 bits.
- All the segments along with the checksum value are added.
- Sum of all segments + Checksum value = 00100101 + 11011010 = 11111111
- Complemented value = 00000000
- Since the result is 0, receiver assumes no error occurred in the data and therefore accepts it.

1.3 Cyclic Redundancy Check (CRC)

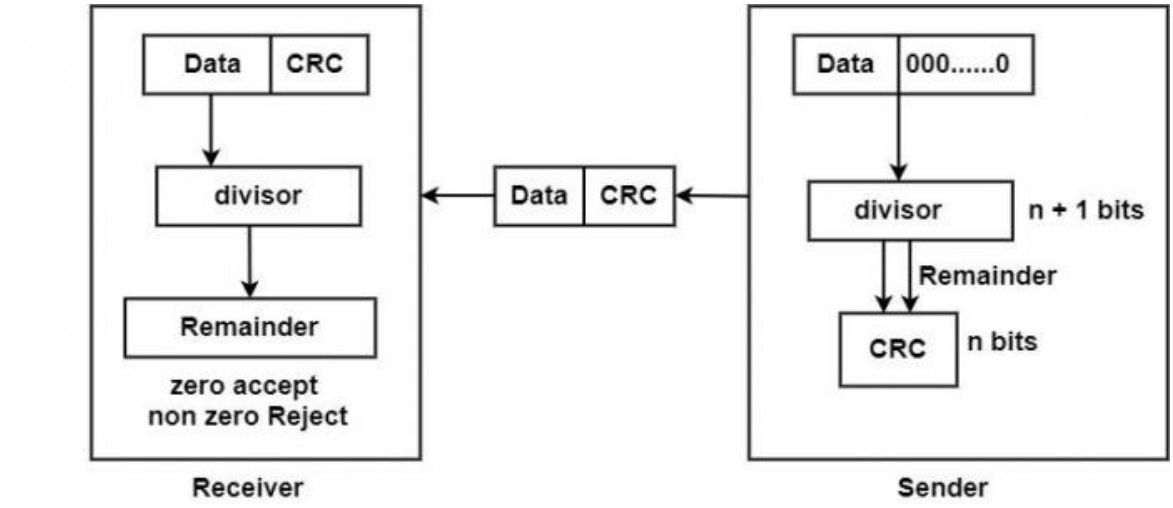
The Cyclic Redundancy Checks (CRC) is the most powerful method for Error-Detection and Correction. It is given as a kbit message and the transmitter creates an (n – k) bit sequence called frame check sequence. The out coming frame, including n bits, is precisely divisible by some fixed number. Modulo 2 Arithmetic is used in this binary addition with no carries, just like the XOR operation.

Redundancy means **duplicacy.** The redundancy bits used by CRC are changed by splitting the data unit by a fixed divisor. The remainder is CRC.

Qualities of CRC

- It should have accurately one less bit than the divisor.
- Joining it to the end of the data unit should create the resulting bit sequence precisely divisible by the divisor.

CRC generator and checker



Process

- A string of n 0s is added to the data unit. The number n is one smaller than the number of bits in the fixed divisor.
- The new data unit is divided by a divisor utilizing a procedure known as binary division; the remainder appearing from the division is CRC.
- The CRC of n bits interpreted in phase 2 restores the added 0s at the end of the data unit.

Example

Message D = 1010001101 (10 bits)

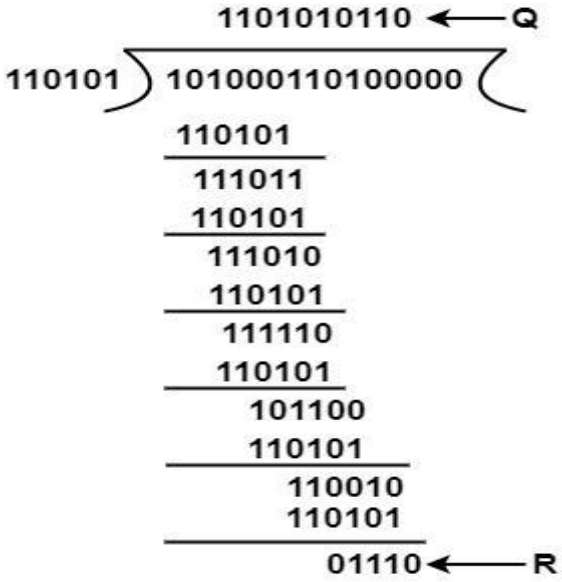
Predetermined P = 110101 (6 bits)

FCS R = to be calculated 5 bits

Hence, n = 15 K = 10 and (n – k) = 5

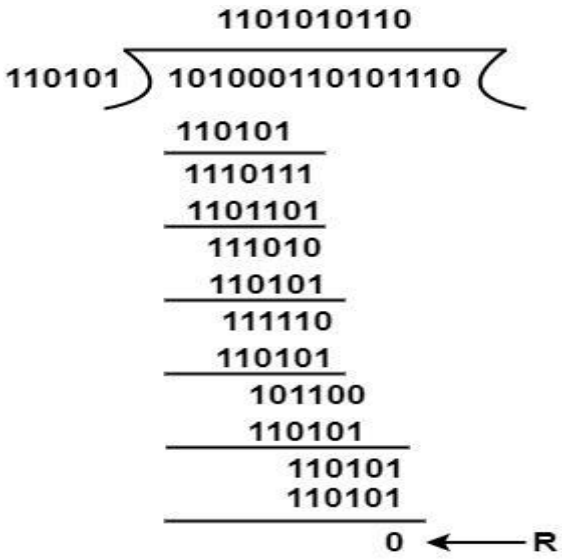
The message is generated through 2^5: accommodating 1010001101000

The product is divided by P.



The remainder is inserted to 2^5D to provide T = 101000110101110 that is sent.

Suppose that there are no errors, and the receiver gets T perfect. The received frame is divided by P.



Because of no remainder, there are no errors.

2. Error Correction Technique:

2.1 Hamming Code:

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 a **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:

1. **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.
2. **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: Hamming Code is simply the use of extra parity bits to allow the identification of an error.

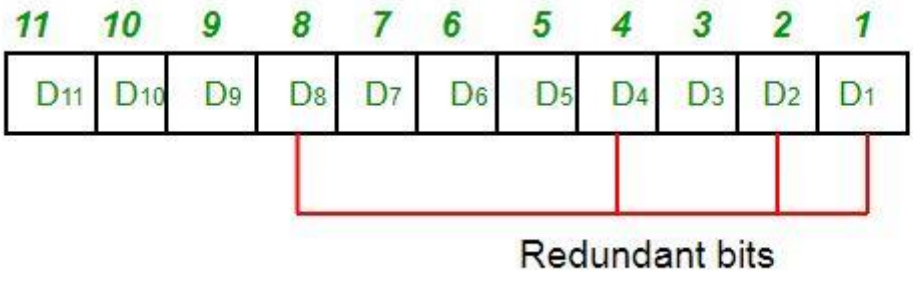
1. Write the bit positions starting from 1 in binary form (1, 10, 11, 100, etc).
2. All the bit positions that are a power of 2 are marked as parity bits (1, 2, 4, 8, etc).
3. All the other bit positions are marked as data bits.
4. Each data bit is included in a unique set of parity bits, as determined 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.
5. 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.
6. Set a parity bit to 0 if the total number of ones in the positions it checks is even.

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

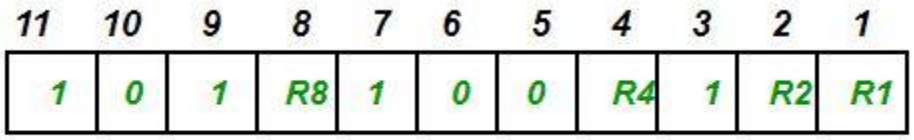
R1 -> 1,3,5,7,9,11
R2 -> 2,3,6,7,10,11
R3 -> 4,5,6,7
R4 -> 8,9,10,11

Determining the position of redundant bits – These redundancy bits are placed at positions that 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

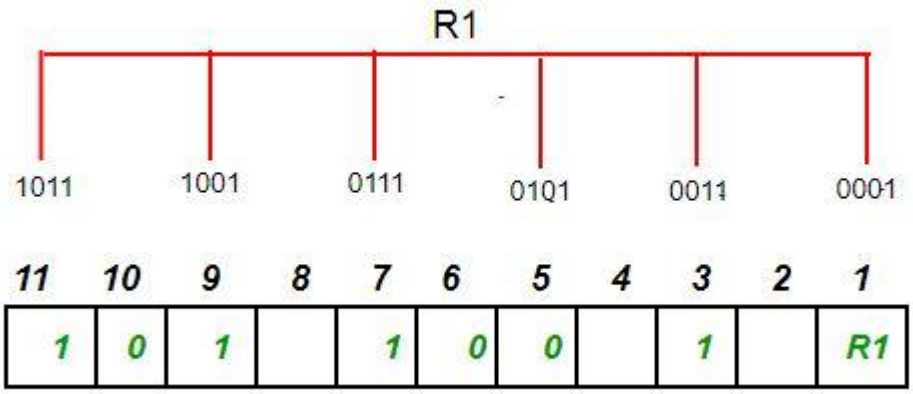


- Suppose the data to be transmitted is 1011001, the bits will be placed as follows:

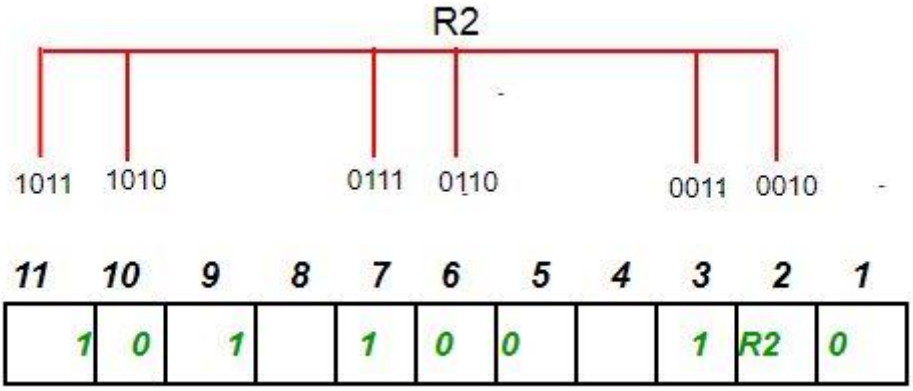


Determining the Parity bits:

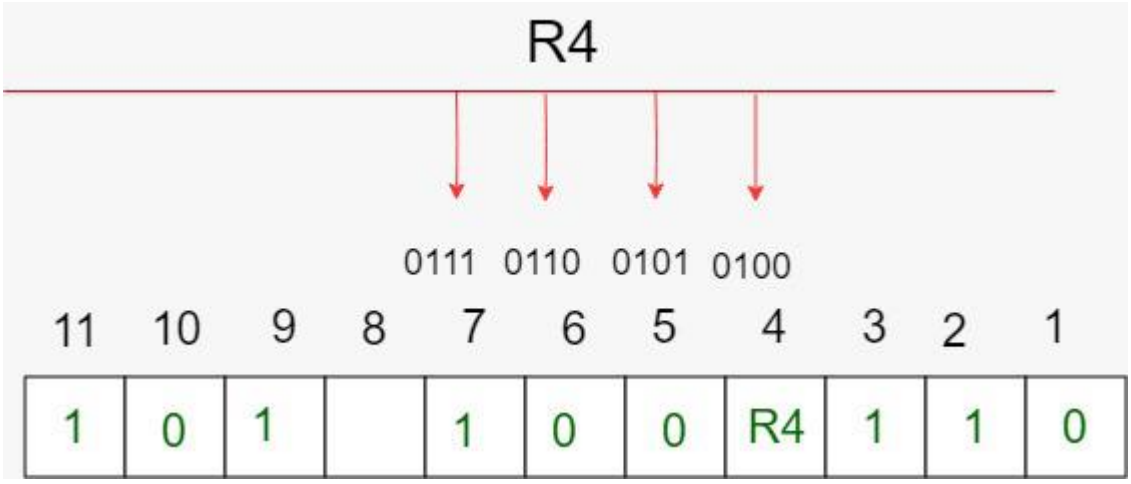
- R1 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the least significant position. R1: bits 1, 3, 5, 7, 9, 11



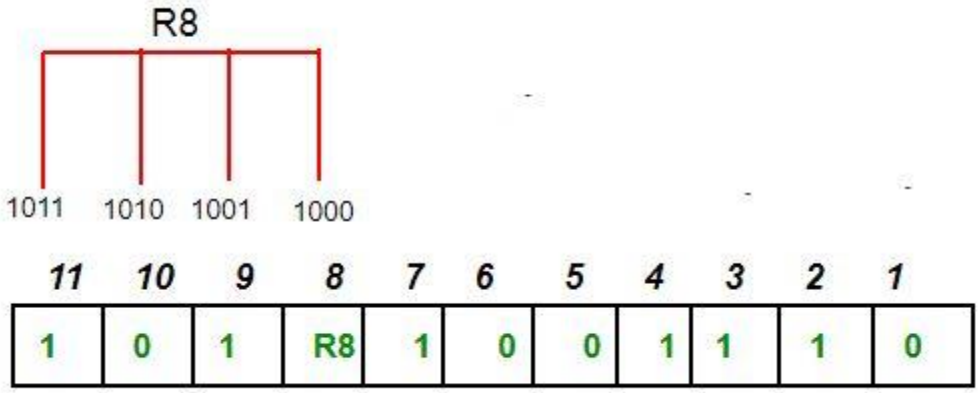
- To find the redundant bit R1, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R1 is an even number the value of R1 (parity bit's value) = 0
- R2 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the second position from the least significant bit. R2: bits 2,3,6,7,10,11



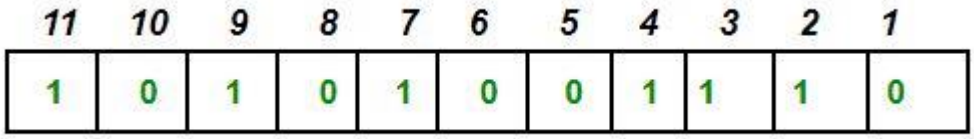
- To find the redundant bit R2, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R2 is odd the value of R2(parity bit's value)=1
- R4 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the third position from the least significant bit. R4: bits 4, 5, 6, 7



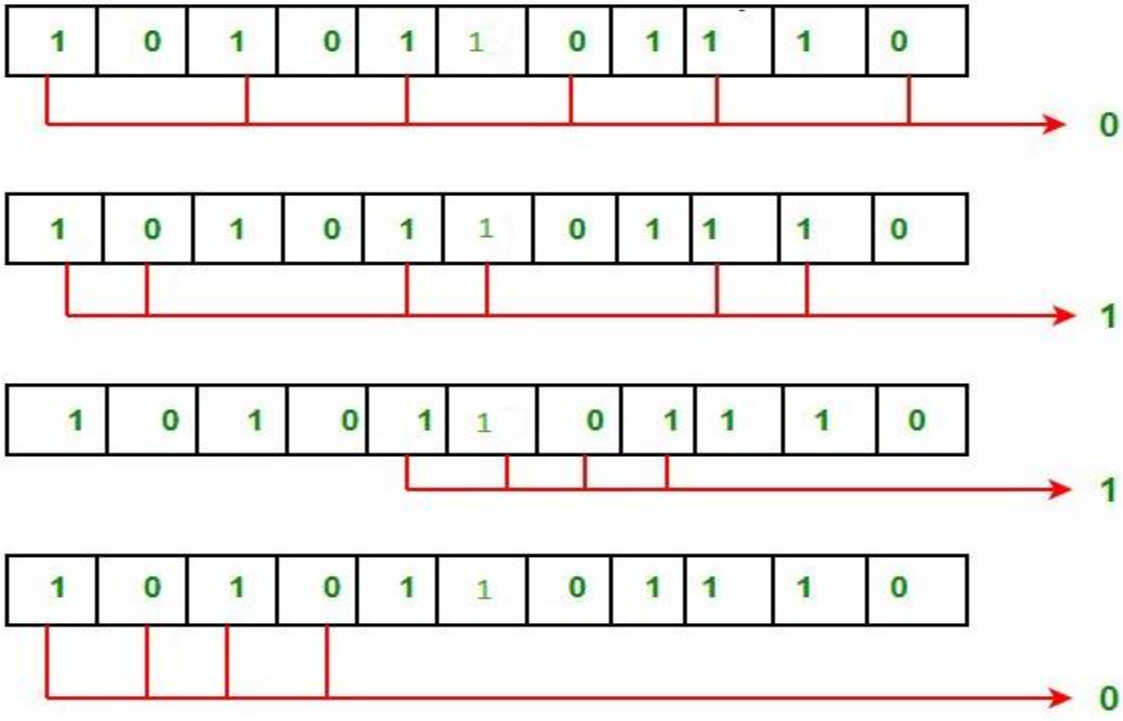
1. To find the redundant bit R4, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R4 is odd the value of R4(parity bit's value) = 1
2. R8 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the fourth 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:



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:

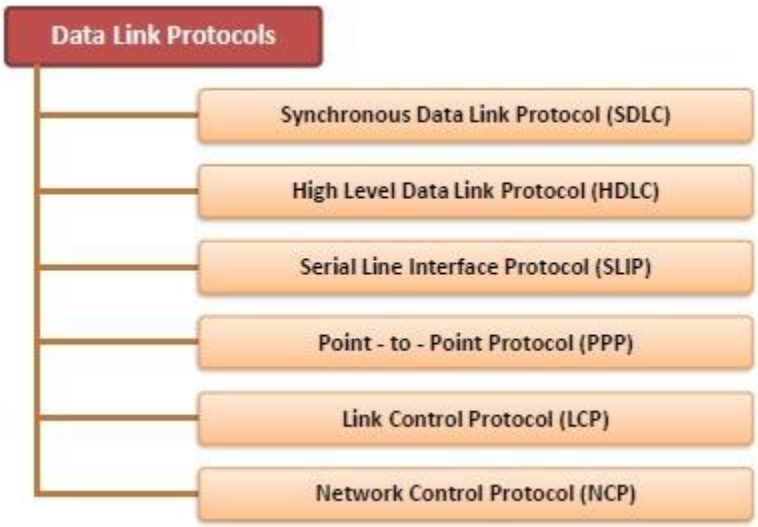


The bits give the binary number 0110 whose decimal representation is 6. Thus, bit 6 contains an error. To correct the error the 6th bit is changed from 1 to 0.

Here are some of the features of Hamming code:

- **Error Detection and Correction:** Hamming code is designed to detect and correct single-bit errors that may occur during the transmission of data. This ensures that the recipient receives the same data that was transmitted by the sender.
- **Redundancy:** Hamming code uses redundant bits to add additional information to the data being transmitted. This redundancy allows the recipient to detect and correct errors that may have occurred during transmission.
- **Efficiency:** Hamming code is a relatively simple and efficient error-correction technique that does not require a lot of computational resources. This makes it ideal for use in low-power and low-bandwidth communication networks.
- **Widely Used:** Hamming code is a widely used error-correction technique and is used in a variety of applications, including telecommunications, computer networks, and data storage systems.
- **Single Error Correction:** Hamming code is capable of correcting a single-bit error, which makes it ideal for use in applications where errors are likely to occur due to external factors such as electromagnetic interference.
- **Limited Multiple Error Correction:** Hamming code can only correct a limited number of multiple errors. In applications where multiple errors are likely to occur, more advanced error-correction techniques may be required.

Example of Data Link Protocol : Common Data Link Protocols



- **Synchronous Data Link Protocol (SDLC)** – SDLC is basically a communication protocol of computer. It usually supports multipoint links even error recovery or error correction also. It is usually used to carry SNA (Systems Network Architecture) traffic and is present precursor to HDLC. It is also used to connect all of the remote devices to mainframe computers at central locations may be in point-to-point (one-to-one) or point-to-multipoint (one-to-many) connections. It is also used to make sure that the data units should arrive correctly and with right flow from one network point to next network point.
- **High-Level Data Link Protocol (HDLC) –** HDLC is basically a protocol that is now assumed to be an umbrella under which many Wide Area protocols sit. It is also adopted as a part of X.25 network. This protocol is generally based on SDLC. HDLC is a bit-oriented protocol that is applicable for point-to-point and multipoint communications both.
- **Serial Line Interface Protocol (SLIP) –** SLIP is generally an older protocol that is just used to add a framing byte at end of IP packet. It is basically a data link control facility that is required for transferring IP packets usually among Internet Service Providers (ISP) and a home user over a dial-up link. It is an encapsulation of the TCP/IP especially designed to work with over serial ports and several router connections simply for communication. It is some limitations like it does not provide mechanisms such as error correction or error detection.
- **Point to Point Protocol (PPP) –** PPP is a protocol that is basically used to provide same functionality as SLIP. It is most robust protocol that is used to transport other types of packets also along with IP Packets. It can also be required for dial-up and leased router-router lines. It basically provides framing method to describe frames. It is a character-oriented protocol that is also used for error detection. It is also used to provides two protocols i.e. NCP and LCP. LCP is used for bringing lines up, negotiation of options, bringing them down whereas NCP is used for negotiating network-layer protocols. It is required for same serial interfaces like that of HDLC.
- **Link Control Protocol (LCP) –** It was originally developed and created by IEEE 802.2. It is also used to provide HDLC style services on LAN (Local Area Network). LCP is basically a PPP protocol that is used for establishing, configuring, testing, maintenance, and ending or terminating links for transmission of data frames.
- **Link Access Procedure (LAP) –** LAP protocols are basically a data link layer protocols that are required for framing and transferring data across point-to-point links. It also

includes some reliability service features. There are basically three types of LAP i.e. LAPB (Link Access Procedure Balanced), LAPD (Link Access Procedure D-Channel), and LAPF (Link Access Procedure Frame-Mode Bearer Services). It is actually originated from IBM SDLC, which is being submitted by IBM to the ISP simply for standardization.

- **Network Control Protocol (NCP) –** NCP was also an older protocol that was implemented by ARPANET. It basically allows users to have access to use computers and some of the devices at remote locations and also to transfer files among two or more computers. It is generally a set of protocols that is forming a part of PPP. NCP is always available for each and every higher-layer protocol that is supported by PPP. NCP was replaced by TCP/IP in the 1980s. It was used to connect remote devices to mainframe computers. It ascertained that data units arrive correctly and with right flow from one network point to the next.

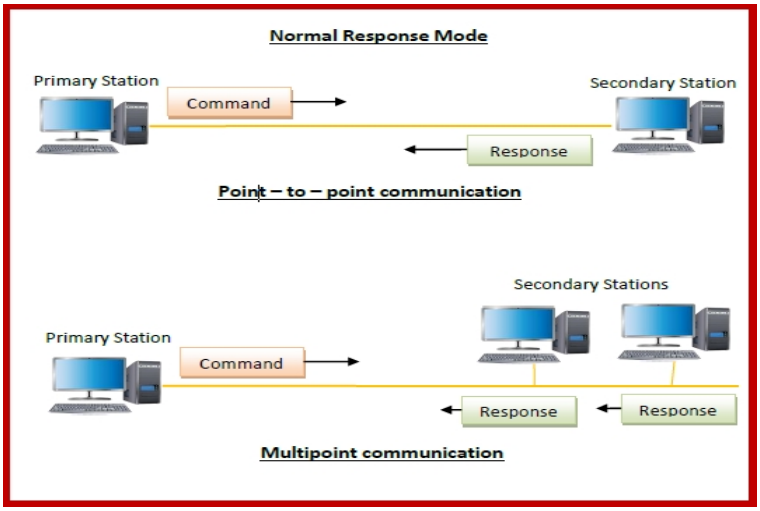
High-level Data Link Control (HDLC)

High-level Data Link Control (HDLC) is a group of communication protocols of the data link layer for transmitting data between network points or nodes. Since it is a data link protocol, data is organized into frames. A frame is transmitted via the network to the destination that verifies its successful arrival. It is a bit - oriented protocol that is applicable for both point - to - point and multipoint communications.

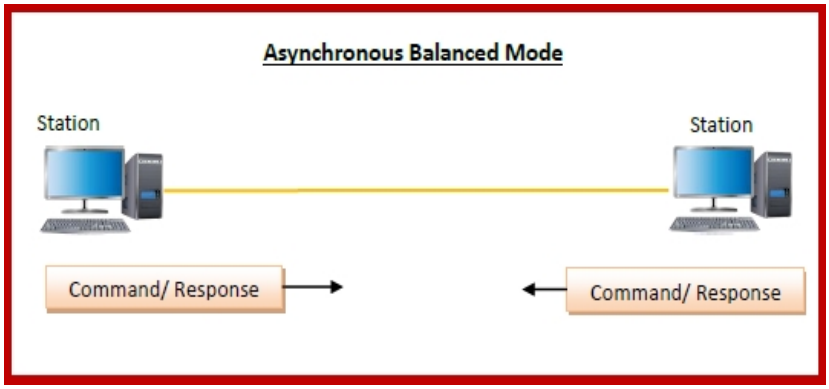
Transfer Modes

HDLC supports two types of transfer modes, normal response mode and asynchronous balanced mode.

- **Normal Response Mode (NRM)** − Here, two types of stations are there, a primary station that send commands and secondary station that can respond to received commands. It is used for both point - to - point and multipoint communications.



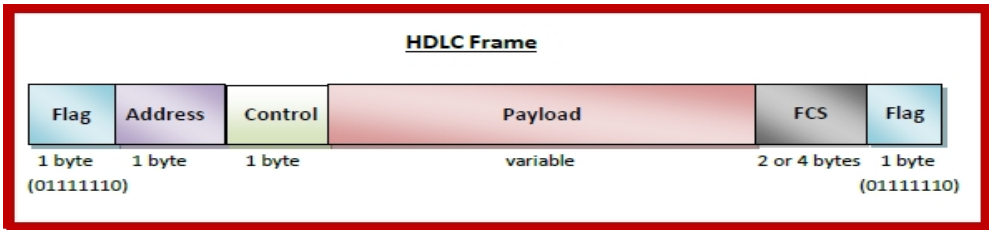
- **Asynchronous Balanced Mode (ABM)** − Here, the configuration is balanced, i.e. each station can both send commands and respond to commands. It is used for only point - to - point communications.



HDLC Frame

HDLC is a bit - oriented protocol where each frame contains up to six fields. The structure varies according to the type of frame. The fields of a HDLC frame are −

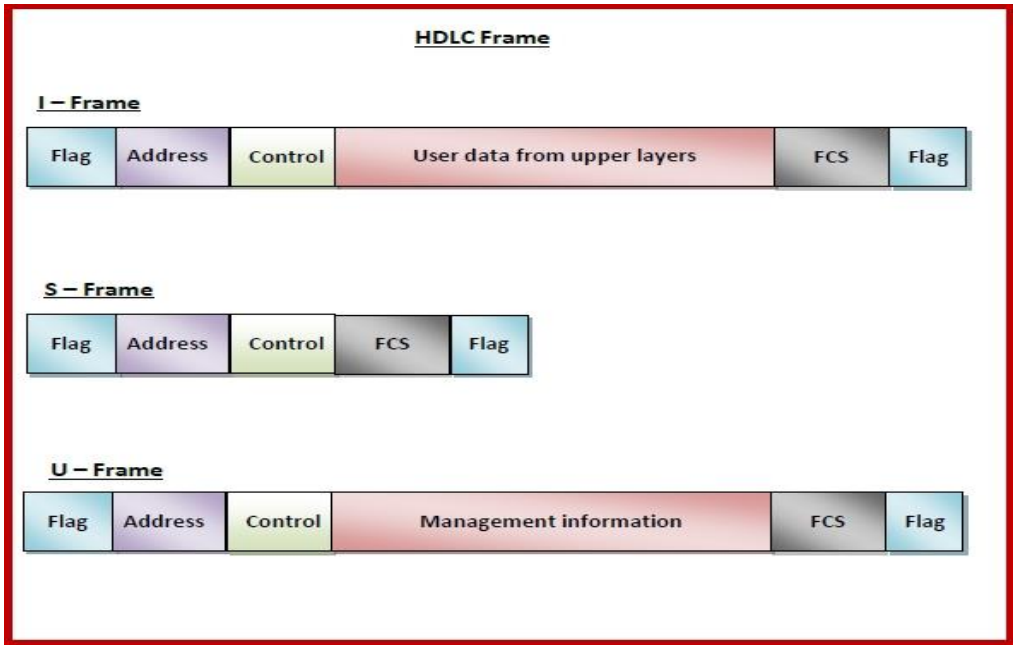
- **Flag** − It is an 8-bit sequence that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
- **Address** − It contains the address of the receiver. If the frame is sent by the primary station, it contains the address(es) of the secondary station(s). If it is sent by the secondary station, it contains the address of the primary station. The address field may be from 1 byte to several bytes.
- **Control** − It is 1 or 2 bytes containing flow and error control information.
- **Payload** − This carries the data from the network layer. Its length may vary from one network to another.
- **FCS** − It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code)



Types of HDLC Frames

There are three types of HDLC frames. The type of frame is determined by the control field of the frame −

- **I-frame** − I-frames or Information frames carry user data from the network layer. They also include flow and error control information that is piggybacked on user data. The first bit of control field of I-frame is 0.
- **S-frame** − S-frames or Supervisory frames do not contain information field. They are used for flow and error control when piggybacking is not required. The first two bits of control field of S-frame is 10.
- **U-frame** − U-frames or Un-numbered frames are used for myriad miscellaneous functions, like link management. It may contain an information field, if required. The first two bits of control field of U-frame is 11.



Medium Access Control Sublayer (MAC sublayer)

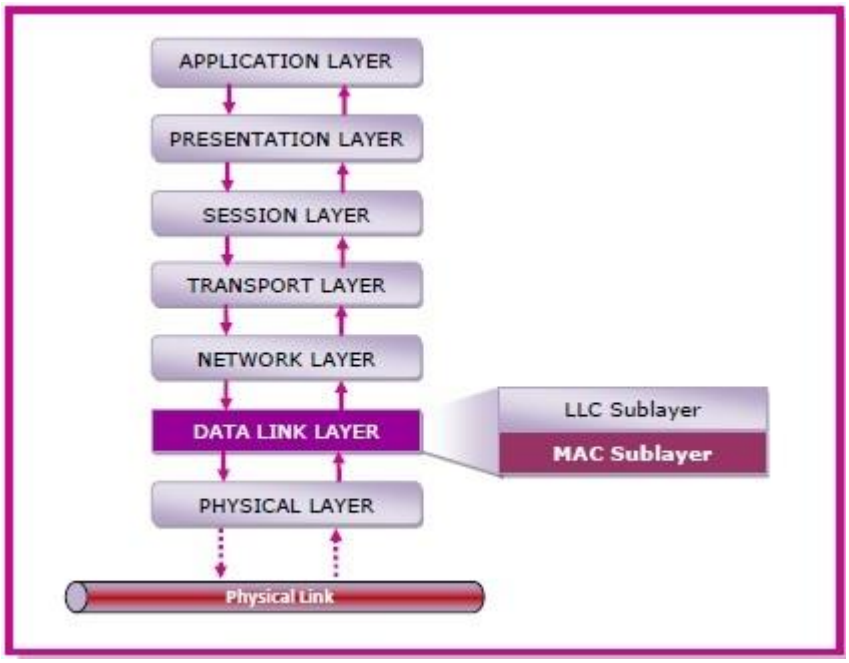
The medium access control (MAC) is a sublayer of the data link layer of the open system interconnections (OSI) reference model for data transmission. It is responsible for flow control and multiplexing for transmission medium. It controls the transmission of data packets via remotely shared channels. It sends data over the network interface card.

MAC Layer in the OSI Model

The Open System Interconnections (OSI) model is a layered networking framework that conceptualizes how communications should be done between heterogeneous systems. The data link layer is the second lowest layer. It is divided into two sublayers −

- The logical link control (LLC) sublayer
- The medium access control (MAC) sublayer

The following diagram depicts the position of the MAC layer −



Functions of MAC Layer

- It provides an abstraction of the physical layer to the LLC and upper layers of the OSI network.
- It is responsible for encapsulating frames so that they are suitable for transmission via the physical medium.
- It resolves the addressing of source station as well as the destination station, or groups of destination stations.
- It performs multiple access resolutions when more than one data frame is to be transmitted. It determines the channel access methods for transmission.
- It also performs collision resolution and initiating retransmission in case of collisions.
- It generates the frame check sequences and thus contributes to protection against transmission errors.

MAC Addresses

MAC address or media access control address is a unique identifier allotted to a network interface controller (NIC) of a device. It is used as a network address for data transmission within a network segment like Ethernet, Wi-Fi, and Bluetooth.

MAC address is assigned to a network adapter at the time of manufacturing. It is hardwired or hard-coded in the network interface card (NIC). A MAC address comprises of six groups of two hexadecimal digits, separated by hyphens, colons, or no separators. An example of a MAC address is 00:0A:89:5B:F0:11.

THE CHANNEL ALLOCATION PROBLEM

The central theme of this chapter is how to allocate a single broadcast channel among competing users. The channel might be a portion of the wireless spectrum in a geographic region, or a single wire or optical fiber to which multiple nodes are connected. It does not matter. In both cases, the channel connects each user to all other users and any user who makes full use of the channel interferes with other users who also wish to use the channel. We will first look at the shortcomings of static allocation schemes for bursty traffic. Then, we will lay out the key assumptions used to model the dynamic schemes that we examine in the following sections.

Channel Allocation Schemes

Channel Allocation may be done using two schemes −

- Static Channel Allocation
- Dynamic Channel Allocation

4.1.1 Static Channel Allocation

The traditional way of allocating a single channel, such as a telephone trunk, among multiple competing users is to chop up its capacity by using one of the multiplexing schemes we described in Sec. 2.5, such as **FDM** (Frequency Division Multiplexing). If there are *N* users, the bandwidth is divided into *N* equal-sized portions, with each user being assigned one portion. Since each user has a private frequency band, there is now no interference among users. When there is only a small and constant number of users, each of which has a steady stream or a heavy load of traffic, this division is a simple and efficient allocation mechanism. A wireless example is FM radio stations. Each station gets a portion of the FM band and uses it most of the time to broadcast its signal.

disadvantages

- However, when the number of senders is large and varying or the traffic is bursty, FDM presents some problems.
- If the spectrum is cut up into *N* regions and fewer than *N* users are currently interested in communicating, a large piece of valuable spectrum will be wasted. And if more than *N* users want to communicate, some of them will be denied permission for lack of bandwidth, even if some of the users who have been assigned a frequency band hardly ever transmit or receive anything.
- Even assuming that the number of users could somehow be held constant at *N*, dividing the single available channel into some number of static subchannels is inherently inefficient. The basic problem is that when some users are quiescent, their bandwidth is simply lost. They are not using it, and no one else is allowed to use it either.
- A static allocation is a poor fit to most computer systems, in which data traffic is extremely bursty, often with peak traffic to mean traffic ratios of 1000:1. Consequently, most of the channels will be idle most of the time

Dynamic Channel Allocation

In dynamic channel allocation scheme, frequency bands are not permanently assigned to the users. Instead channels are allotted to users dynamically as needed, from a central pool. The allocation is done considering a number of parameters so that transmission interference is

minimized. This allocation scheme optimises bandwidth usage and results is faster transmissions.

Dynamic channel allocation is further divided into centralised and distributed allocation. Before we get to the first of the many channel allocation methods in this chapter, it is worthwhile to carefully formulate the allocation problem. Underlying all the work done in this area are the following five key assumptions:

1. **Independent Traffic**. The model consists of *N* independent **stations** (e.g., computers, telephones), each with a program or user that generates frames for transmission. Once a frame has been generated, the station is blocked and does nothing until the frame has been successfully transmitted.

2. **Single Channel**. A single channel is available for all communication. All stations can transmit on it and all can receive from it. The stations are assumed to be equally capable, though protocols may assign them different roles (e.g., priorities).
3. **Observable Collisions**. If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled. This event is called a **collision**. All stations can detect that a collision has occurred. A collided frame must be transmitted again later. No errors other than those generated by collisions occur.
4. **Continuous or Slotted Time**. Time may be assumed continuous, in which case frame transmission can begin at any instant. Alternatively, time may be slotted or divided into discrete intervals (called slots). Frame transmissions must then begin at the start of a slot. A slot may contain 0, 1, or more frames, corresponding to an idle slot, a successful transmission, or a collision, respectively.
5. **Carrier Sense or No Carrier Sense**. With the carrier sense assumption, stations can tell if the channel is in use before trying to use it. No station will attempt to use the channel while it is sensed as busy. If there is no carrier sense, stations cannot sense the channel before trying to use it. They just go ahead and transmit. Only later can they determine whether the transmission was successful.

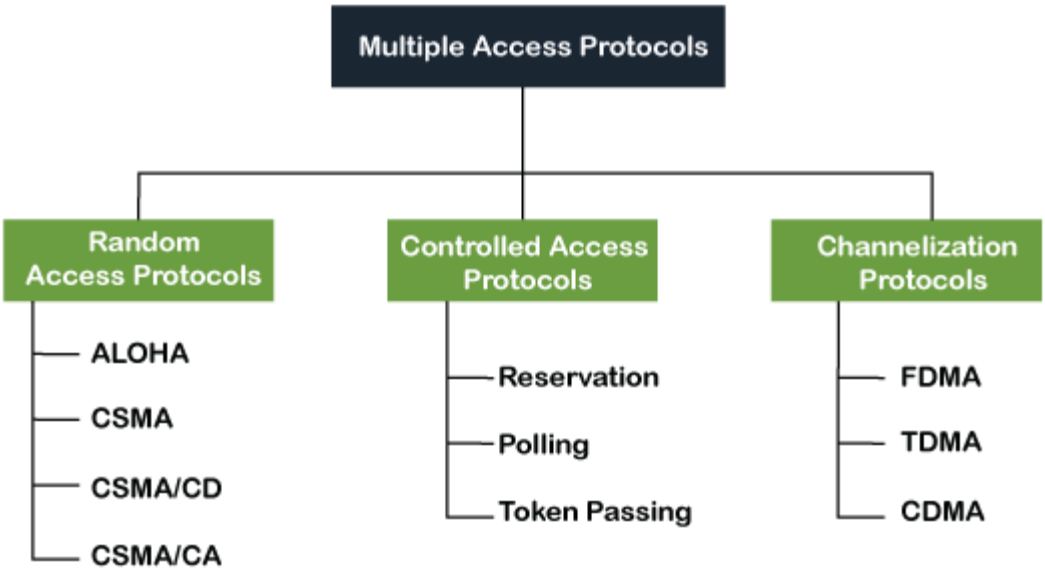
MULTIPLE ACCESS PROTOCOLS

What is a multiple access protocol?

When a sender and receiver have a dedicated link to transmit data packets, the data link control is enough to handle the channel. Suppose there is no dedicated path to communicate or transfer the data between two devices. In that case, multiple stations access the channel and simultaneously transmits the data over the channel. It may create collision and cross talk. Hence, the multiple access protocol is required to reduce the collision and avoid crosstalk between the channels.

For example, suppose that there is a classroom full of students. When a teacher asks a question, all the students (small channels) in the class start answering the question at the same time (transferring the data simultaneously). All the students respond at the same time due to which data is overlap or data lost. Therefore it is the responsibility of a teacher (multiple access protocol) to manage the students and make them one answer.

Following are the types of multiple access protocol that is subdivided into the different process as:



A. Random Access Protocol

In this protocol**, all the station has the equal priority** to send the data over a channel. In random access protocol, one or more stations cannot depend on another station nor any station control another station. Depending on the channel's state (idle or busy), each station transmits the data frame. However, if more than one station sends the data over a channel, there may be a collision or data conflict. Due to the collision, the data frame packets may be lost or changed. And hence, it does not receive by the receiver end.

Following are the different methods of random-access protocols for broadcasting frames on the channel.

- Aloha
- CSMA
- CSMA/CD
- CSMA/CA

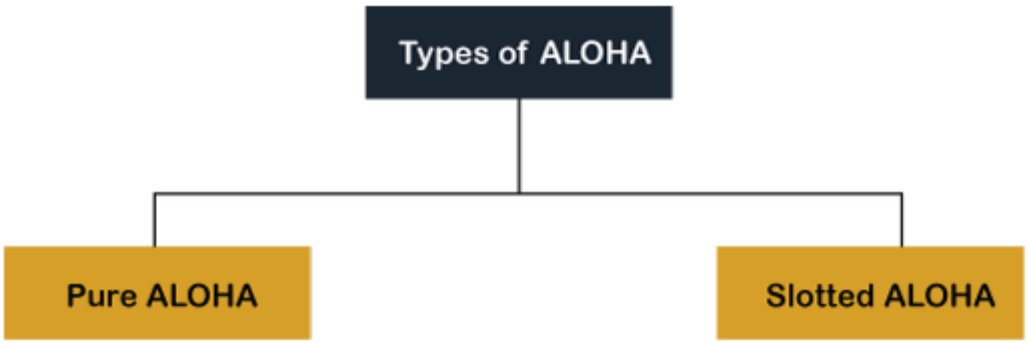
ALOHA Random Access Protocol

It is designed for wireless LAN (Local Area Network) but can also be used in a shared medium to transmit data. Using this method, any station can transmit data across a network simultaneously when a data frameset is available for transmission.

Aloha Rules

1. Any station can transmit data to a channel at any time.
2. It does not require any carrier sensing.
3. Collision and data frames may be lost during the transmission of data through multiple stations.
4. Acknowledgment of the frames exists in Aloha. Hence, there is no collision detection.

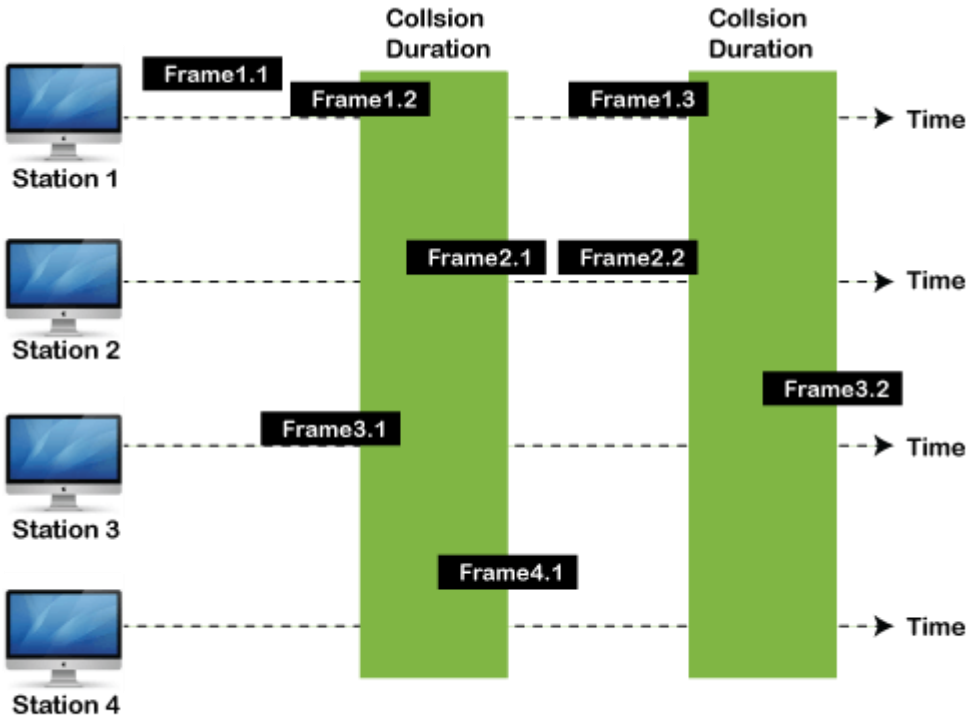
5. It requires retransmission of data after some random amount of time.



Pure Aloha

Whenever data is available for sending over a channel at stations, we use Pure Aloha. In pure Aloha, when each station transmits data to a channel without checking whether the channel is idle or not, the chances of collision may occur, and the data frame can be lost. When any station transmits the data frame to a channel, the pure Aloha waits for the receiver's acknowledgment. If it does not acknowledge the receiver end within the specified time, the station waits for a random amount of time, called the backoff time (Tb). And the station may assume the frame has been lost or destroyed. Therefore, it retransmits the frame until all the data are successfully transmitted to the receiver.

1. The total vulnerable time of pure Aloha is 2 * Tfr.
2. Maximum throughput occurs when G = 1/ 2 that is 18.4%.
3. Successful transmission of data frame is S = G * e ^ - 2 G.



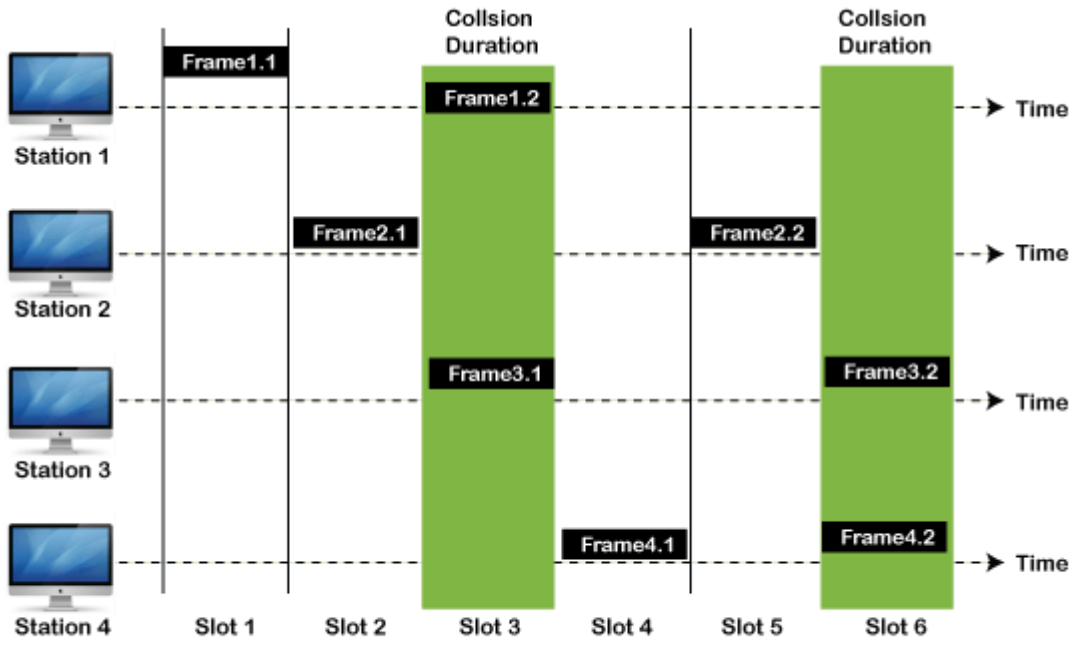
Frames in Pure ALOHA

As we can see in the figure above, there are four stations for accessing a shared channel and transmitting data frames. Some frames collide because most stations send their frames at the same time. Only two frames, frame 1.1 and frame 2.2, are successfully transmitted to the receiver end. At the same time, other frames are lost or destroyed. Whenever two frames fall on a shared channel simultaneously, collisions can occur, and both will suffer damage. If the new frame's first bit enters the channel before finishing the last bit of the second frame. Both frames are completely finished, and both stations must retransmit the data frame.

Slotted Aloha

The slotted Aloha is designed to overcome the pure Aloha's efficiency because pure Aloha has a very high possibility of frame hitting. In slotted Aloha, the shared channel is divided into a fixed time interval called **slots**. So that, if a station wants to send a frame to a shared channel, the frame can only be sent at the beginning of the slot, and only one frame is allowed to be sent to each slot. And if the stations are unable to send data to the beginning of the slot, the station will have to wait until the beginning of the slot for the next time. However, the possibility of a collision remains when trying to send a frame at the beginning of two or more station time slot.

1. Maximum throughput occurs in the slotted Aloha when G = 1 that is 37%.
2. The probability of successfully transmitting the data frame in the slotted Aloha is S = G * e ^ - G.
3. The total vulnerable time required in slotted Aloha is Tfr.



Frames in Slotted ALOHA

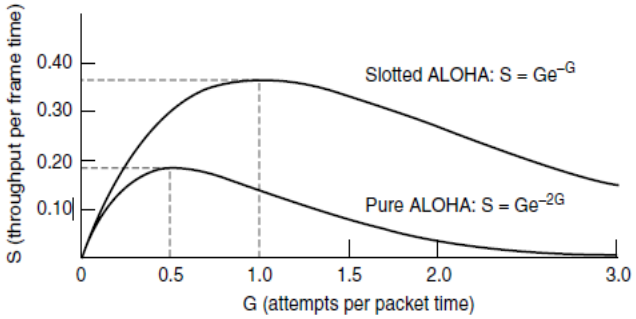


Figure 4-3. Throughput versus offered traffic for ALOHA systems.

CSMA (Carrier Sense Multiple Access)

It is a **carrier sense multiple access** based on media access protocol to sense the traffic on a channel (idle or busy) before transmitting the data. It means that if the channel is idle, the station can send data to the channel. Otherwise, it must wait until the channel becomes idle. Hence, it reduces the chances of a collision on a transmission medium.

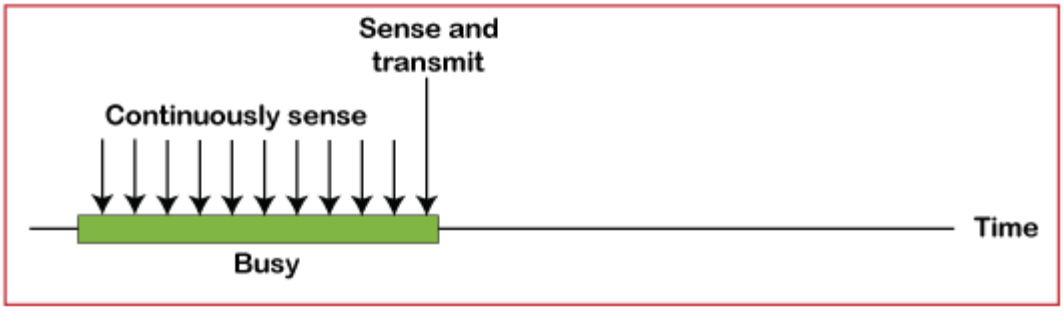
CSMA Access Modes

1-Persistent: In the 1-Persistent mode of CSMA that defines each node, first sense the shared channel and if the channel is idle, it immediately sends the data. Else it must wait and keep track of the status of the channel to be idle and broadcast the frame unconditionally as soon as the channel is idle.

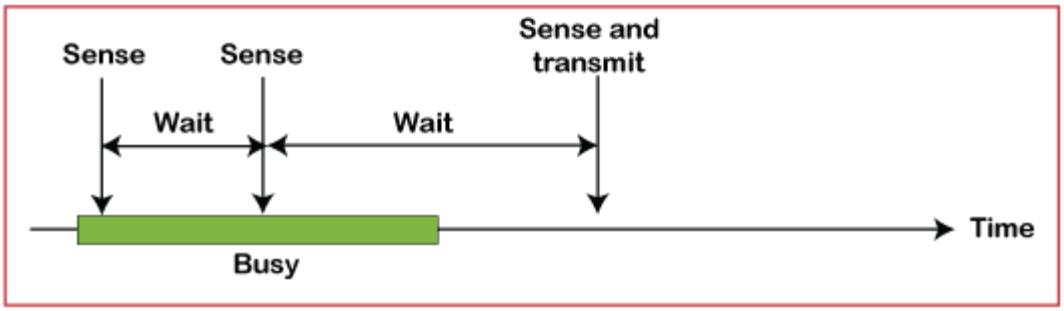
Non-Persistent: It is the access mode of CSMA that defines before transmitting the data, each node must sense the channel, and if the channel is inactive, it immediately sends the data. Otherwise, the station must wait for a random time (not continuously), and when the channel is found to be idle, it transmits the frames.

P-Persistent: It is the combination of 1-Persistent and Non-persistent modes. The P-Persistent mode defines that each node senses the channel, and if the channel is inactive, it sends a frame with a **P** probability. If the data is not transmitted, it waits for a (**q = 1-p probability**) random time and resumes the frame with the next time slot.

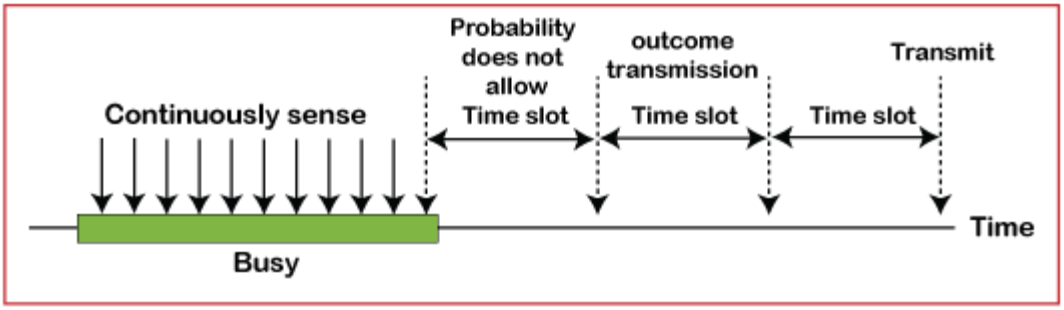
O- Persistent: It is an O-persistent method that defines the superiority of the station before the transmission of the frame on the shared channel. If it is found that the channel is inactive, each station waits for its turn to retransmit the data.



a. 1-persistent



b. Nonpersistent



c. p-persistent

CSMA/ CD

It is a **carrier sense multiple access/ collision detection** network protocol to transmit data frames. The CSMA/CD protocol works with a medium access control layer. Therefore, it first senses the shared channel before broadcasting the frames, and if the channel is idle, it transmits a frame to check whether the transmission was successful. If the frame is successfully received, the station sends another frame. If any collision is detected in the CSMA/CD, the station sends a jam/ stop signal to the shared channel to terminate data transmission. After that, it waits for a random time before sending a frame to a channel.

CSMA/ CA

It is a **carrier sense multiple access/collision avoidance** network protocol for carrier transmission of data frames. It is a protocol that works with a medium access control layer. When a data frame is sent to a channel, it receives an acknowledgment to check whether the channel is clear. If the station receives only a single (own) acknowledgments, that means the data frame has been successfully transmitted to the receiver. But if it gets two signals (its own and one more in which the collision of frames), a collision of the frame occurs in the shared channel. Detects the collision of the frame when a sender receives an acknowledgment signal.

Following are the methods used in the CSMA/ CA to avoid the collision:

Interframe space: In this method, the station waits for the channel to become idle, and if it gets the channel is idle, it does not immediately send the data. Instead of this, it waits for some time, and this time period is called the **Interframe** space or IFS. However, the IFS time is often used to define the priority of the station.

Contention window: In the Contention window, the total time is divided into different slots. When the station/ sender is ready to transmit the data frame, it chooses a random slot number of slots as **wait time**. If the channel is still busy, it does not restart the entire process, except that it restarts the timer only to send data packets when the channel is inactive.

Acknowledgment: In the acknowledgment method, the sender station sends the data frame to the shared channel if the acknowledgment is not received ahead of time.

B. Controlled Access Protocol

It is a method of reducing data frame collision on a shared channel. In the controlled access method, each station interacts and decides to send a data frame by a particular station approved by all other stations. It means that a single station cannot send the data frames unless all other stations are not approved. It has three types of controlled access: **Reservation, Polling**, and **Token Passing**.

C. Channelization Protocols

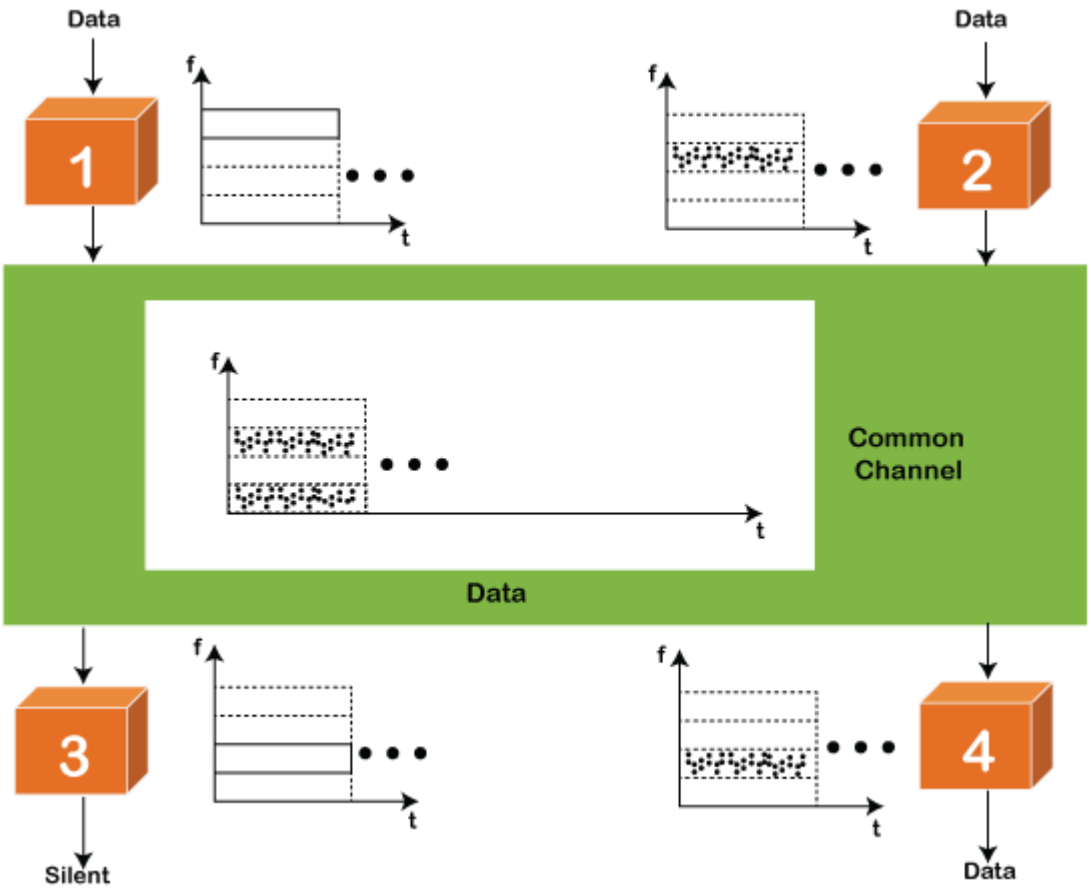
It is a channelization protocol that allows the total usable bandwidth in a shared channel to be shared across multiple stations based on their time, distance and codes. It can access all the stations at the same time to send the data frames to the channel.

Following are the various methods to access the channel based on their time, distance and codes:

1. FDMA (Frequency Division Multiple Access)
2. TDMA (Time Division Multiple Access)
3. CDMA (Code Division Multiple Access)

FDMA

It is a frequency division multiple access (**FDMA**) method used to divide the available bandwidth into equal bands so that multiple users can send data through a different frequency to the subchannel. Each station is reserved with a particular band to prevent the crosstalk between the channels and interferences of stations.



TDMA

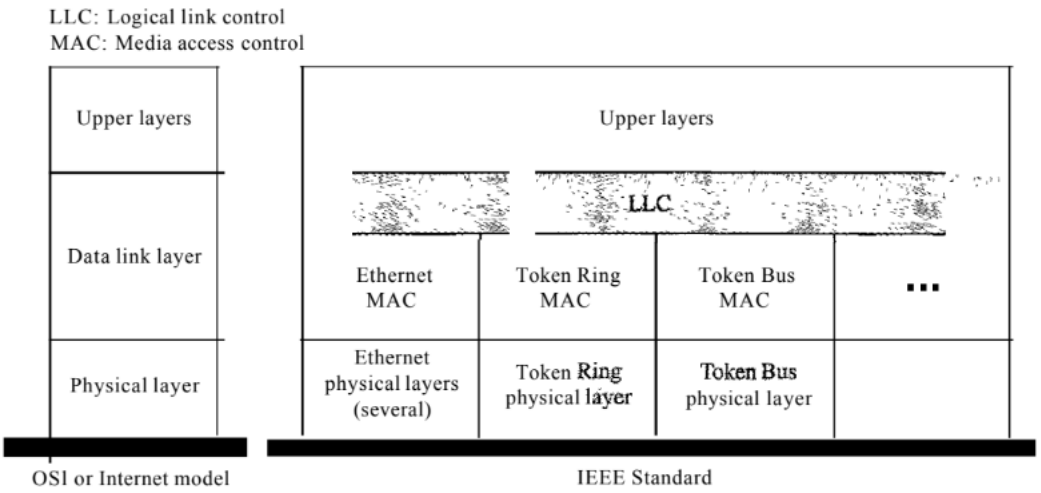
Time Division Multiple Access (**TDMA**) is a channel access method. It allows the same frequency bandwidth to be shared across multiple stations. And to avoid collisions in the shared channel, it divides the channel into different frequency slots that allocate stations to transmit the data frames. The same **frequency** bandwidth into the shared channel by dividing the signal into various time slots to transmit it. However, TDMA has an overhead of synchronization that specifies each station's time slot by adding synchronization bits to each slot.

CDMA

The code division multiple access (CDMA) is a channel access method. In CDMA, all stations can simultaneously send the data over the same channel. It means that it allows each station to transmit the data frames with full frequency on the shared channel at all times. It does not require the division of bandwidth on a shared channel based on time slots. If multiple stations send data to a channel simultaneously, their data frames are separated by a unique code sequence. Each station has a different unique code for transmitting the data over a shared channel. For example, there are multiple users in a room that are continuously speaking. Data is received by the users if only two-person interact with each other using the same language. Similarly, in the network, if different stations communicate with each other simultaneously with different code language.

Wired LANs: Ethernet

Figure 13.1 *IEEE standardfor LANs*



Logical Link Control (LLC)

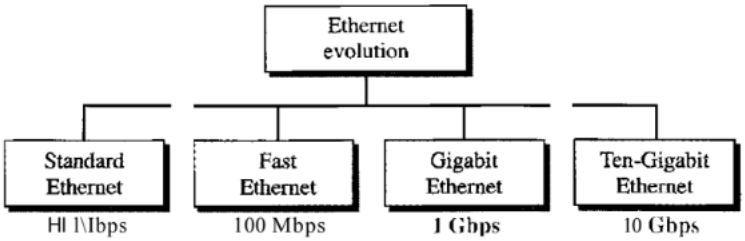
We said that data link control handles framing, flow control, and error control. In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sublayer called the logical link control. Framing is handled in both the LLC sublayer and the MAC sublayer.

Media Access Control (MAC)

we discussed multiple access methods including random access, controlled access, and channelization. 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 the token passing method for Token Ring and Token Bus LANs.

STANDARD ETHERNET

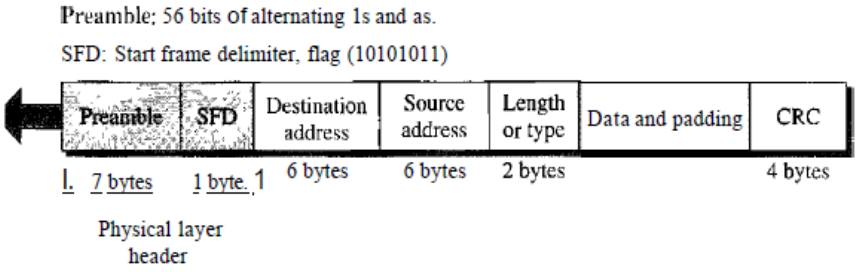
Figure 13.3 *Ethernet evolution through four generations*



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 CRe. 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 ofthe MAC frame is shown in Figure 13.4.

Figure 13.4 *802.3 MAC frame*

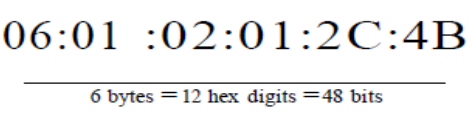


- **Preamble** – informs the receiving system that a frame is starting and enables synchronisation.
- **SFD (Start Frame Delimiter)** – signifies that the Destination MAC Address field begins with the next byte.
- **Destination MAC** – identifies the receiving system.
- **Source MAC** – identifies the sending system.
- **Type** – defines the type of protocol inside the frame, for example IPv4 or IPv6.
- **Data and Pad** – contains the payload data. Padding data is added to meet the minimum length requirement for this field (46 bytes).
- **CRC** – contains a 32-bit Cyclic Redundancy Check (CRC) which allows detection of corrupted data.

Addressing

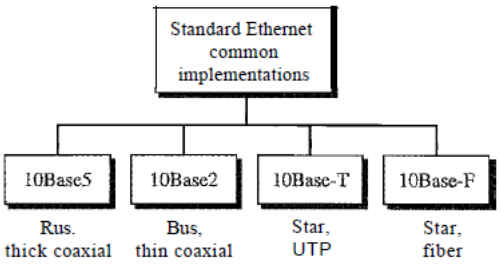
Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC). The NIC fits inside the station and provides the station with a 6-byte physical address. As shown in Figure 13.6, the Ethernet address is 6 bytes (48 bits), nonnally written in hexadecimal notation, with a colon between the bytes.

Figure 13.6 *Example of an Ethernet address in hexadecimal notation*



Standard Ethernet

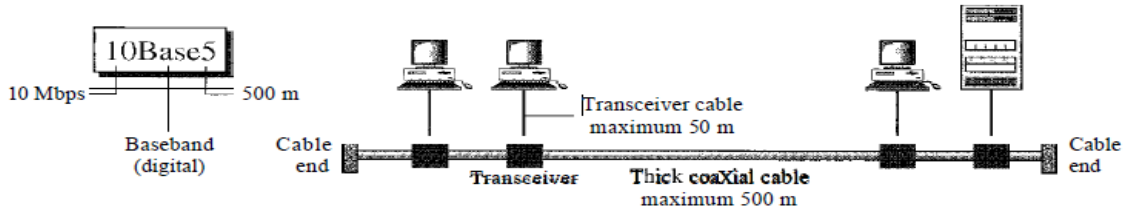
Figure 13.8 *Categories of Standard Ethernet*



lOBase5: Thick Ethernet

The first implementation is called **10BaseS, thick Ethernet, or Thicknet.** The nick-name derives from the size of the cable, which is roughly the size of a garden hose and too stiff to bend with your hands. lOBaseS 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 13.10 shows a schematic diagram of a lOBase5 implementation.

Figure 13.10 *IOBase5 implementation*



10Base2: Thin Ethernet

The second implementation is called lOBase2, **thin** Ethernet, or Cheapernet. IOBase2 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 13.11 shows the schematic diagram of a IOBase2 implementation.

Figure 13.11 *lOBase2 implementation*

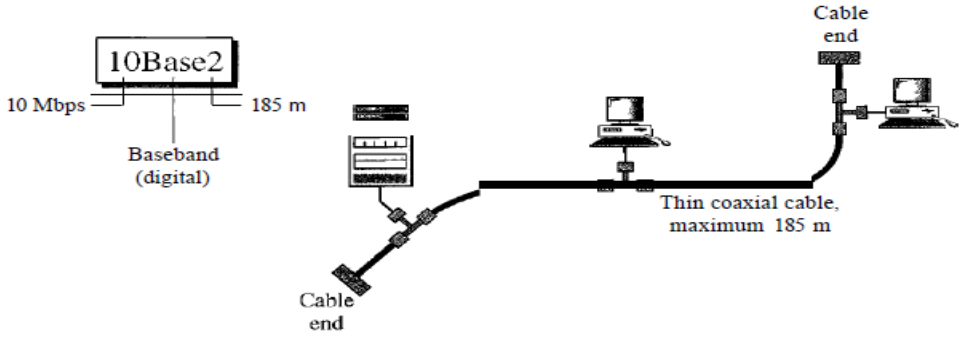
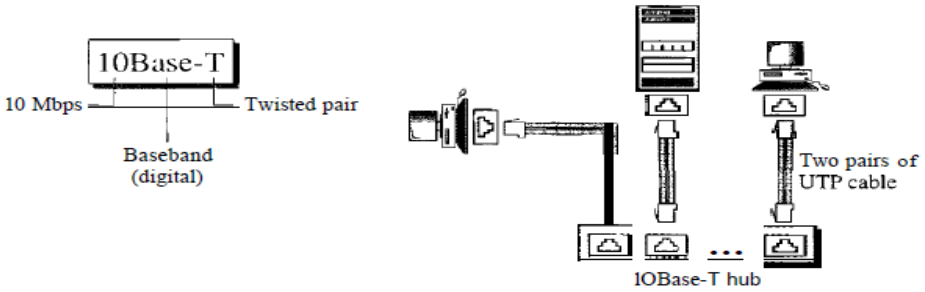


Figure 13.12 *IOBase-T implementation*

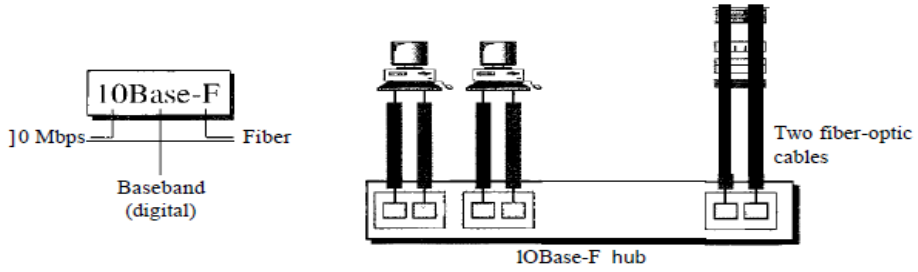


cable as far as a collision is concerned. The maximum length of the twisted cable here is defined as 100 m, to minimize the effect of attenuation in the twisted cable.

lOBase-F: Fiber Ethernet

Although there are several types of optical fiber lO-Mbps Ethernet, the most common is called 10Base-F. lOBase-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 13.13.

Figure 13.13 *IOBase-F implementation*



Fast Ethernet

Fast Ethernet was designed to compete with LAN protocols such as FDDI or Fiber Channel (or Fibre Channel, as it is sometimes spelled). IEEE created Fast Ethernet under the name 802.3u. Fast Ethernet is backward-compatible with Standard Ethernet, but it can transmit data 10 times faster at a rate of 100 Mbps. 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.

Implementation

Fast Ethernet implementation at the physical layer can be categorized as either two-wire or four-wire. 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). See Figure 13.20.

Figure 13.20 *Fast Ethernet implementations*

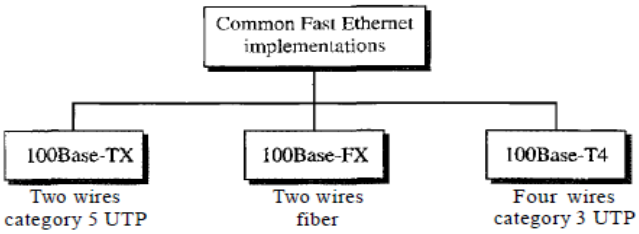


Table 13.2 *Summary of Fast Ethernet implementations*

Characteristics	*lOOBase-TX*	*lOOBase-FX*	*100Base-T4*
Media	Cat 5 UTP or STP	Fiber	Cat 4 UTP
Number of wires	2	2	4
Maximum length	100m	100m	100m

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 the Standard 802.3z. The goals of the Gigabit Ethernet design can be summarized as follows:

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

Table 13.3 *Summary of Gigabit Ethernet implementations*

Characteristics	*lOOOBase-SX*	*lOOOBase-LX*	*lOOOBase-CX*	*lOOOBase-T*
Media	Fiber short-wave	Fiber long-wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum length	550m	5000m	25m	100m

Ten-Gigabit Ethernet

The IEEE committee created Ten-Gigabit Ethernet and called it Standard 802.3ae. The goals of the Ten-Gigabit Ethernet design can be summarized as follows:

1. Upgrade the data rate to 10 Gbps.
2. Make it compatible with Standard, Fast, and Gigabit Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format.
5. Keep the same minimum and maximum frame lengths.
6. Allow the interconnection of existing LANs into a metropolitan area network (MAN) or a wide area network (WAN).
7. Make Ethernet compatible with technologies such as Frame Relay and ATM (see

Table 13.4 *Summary of Ten-Gigabit Ethernet implementations*

Characteristics	*lOGBase-S*	*lOGBase-L*	*lOGBase-E*
Media	Short-wave S50-nrn rnultimode	Long-wave 131O-nm single mode	Extended 1550-mrn single mode
Maximum length	300m	lOkm	40km

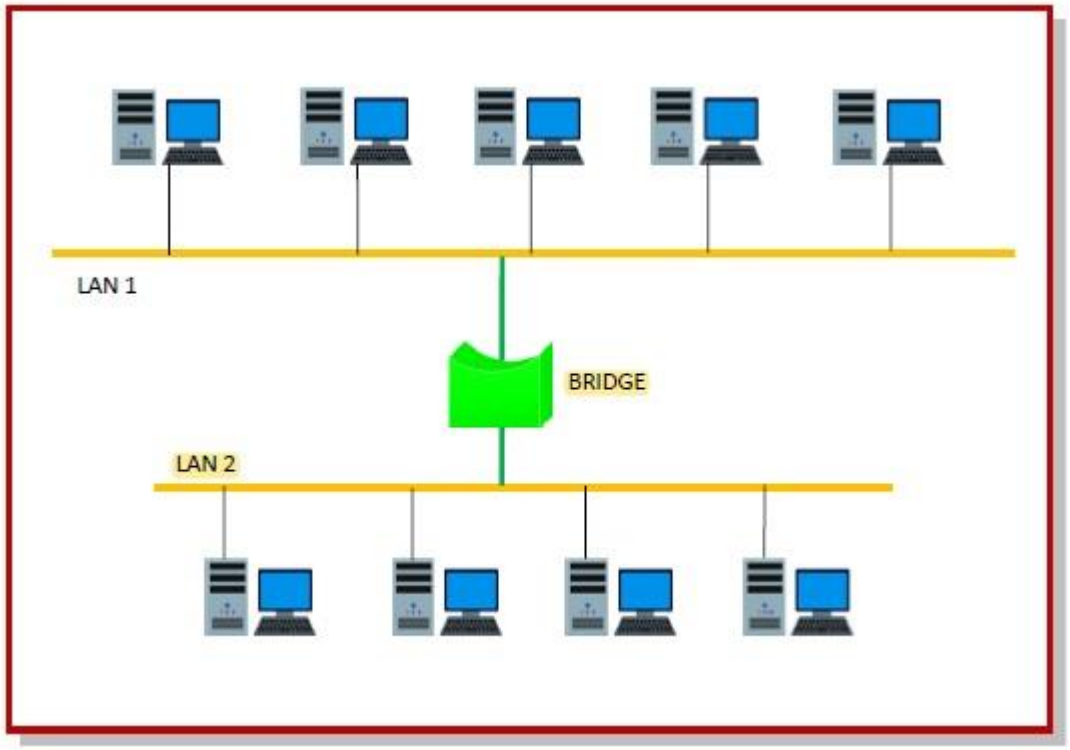
What is Data Link Layer Switching

Network switching is the process of forwarding data frames or packets from one port to another leading to data transmission from source to destination. Data link layer is the second layer of the Open System Interconnections (OSI) model whose function is to divide the stream of bits from physical layer into data frames and transmit the frames according to switching requirements. Switching in data link layer is done by network devices called **bridges**.

Bridges

A data link layer bridge connects multiple LANs (local area networks) together to form a larger LAN. This process of aggregating networks is called network bridging. A bridge connects the different components so that they appear as parts of a single network.

The following diagram shows connection by a bridge −



Switching by Bridges

When a data frame arrives at a particular port of a bridge, the bridge examines the frame's data link address, or more specifically, the MAC address. If the destination address as well as the required switching is valid, the bridge sends the frame to the destined port. Otherwise, the frame is discarded.

The bridge is not responsible for end to end data transfer. It is concerned with transmitting the data frame from one hop to the next. Hence, they do not examine the payload field of the frame. Due to this, they can help in switching any kind of packets from the network layer above.

Bridges also connect virtual LANs (VLANs) to make a larger VLAN.

If any segment of the bridged network is wireless, a wireless bridge is used to perform the switching.

There are three main ways for bridging −

- simple bridging
- multi-port bridging
- learning or transparent bridging

Wireless LAN/ Wireless Ethernet

In this chapter, we concentrate on two promising wireless technologies for LANs: IEEE 802.11 wireless LANs, sometimes called wireless Ethernet, and Bluetooth, a technology for small wireless LANs. Although both protocols need several layers to operate, we concentrate mostly on the physical and data link layers.

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

Architecture

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

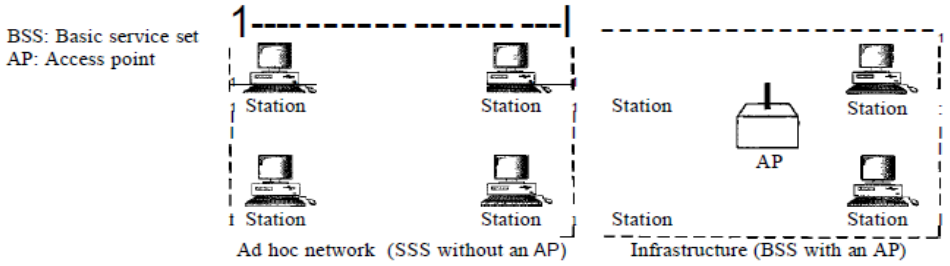
Basic Service Set

IEEE 802.11 defines the basic service set (BSS) as the building block 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 14.1 shows two sets in this standard.

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

A BSS without an AP is called an ad hoc network;
a BSS with an AP is called an infrastructure network.

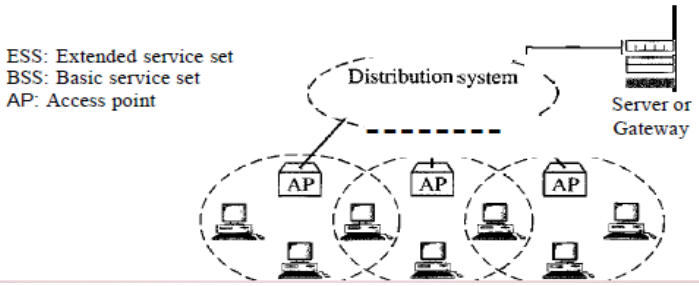
Figure 14.1 *Basic service sets (BSSs)*



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 usually a wired LAN. The distribution system connects the APs in the BSSs. IEEE 802.11 does not restrict the distribution system; it can be any IEEE LAN such as an Ethernet. Note that the extended service set uses two types of stations: mobile and stationary. The mobile stations are normal stations inside a BSS. The stationary stations are AP stations that are part of a wired LAN. Figure 14.2 shows an ESS.

Figure 14.2 *Extended service sets (ESSs)*



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

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

Bluetooth was originally started as a project by the Ericsson Company. It is named for Harald Blaatand, the king of Denmark (940-981) who united Denmark and Norway. *Blaatand* translates to *Bluetooth* in English.

Today, Bluetooth technology is the implementation of a protocol defined by the IEEE 802.15 standard. The standard defines a wireless personal-area network (PAN) operable in an area the size of a room or a hall.

Architecture

Bluetooth defines two types of networks: piconet and scatternet.

Piconets

A Bluetooth network is called a piconet, or a small net. A piconet can have up to eight stations, one of which is called the primary; the rest are called secondaries. All the secondary stations synchronize their clocks and hopping sequence with the primary. Note that a piconet can have only one primary station. The communication between the primary and the secondary can be one-to-one or one-to-many. Figure 14.19 shows a piconet.

Figure 14.19 *Piconet*