Unit 4: Data Link Layer [8 Hrs.]

4.1 Functions of Data Link Layer (DLL)

4.2 Overview of Logical Link Control (LLC) and Media Access Control (MAC)

4.3 Framing and its Types

4.4 Flow Control Mechanism: Simple Stop and Wait ARQ, Sliding Window, Go-Back-N ARQ, Selective Repeat ARQ

4.5 Error Detection and Correction Techniques: Parity Check, Checksum, Cyclic Redundancy Check (CRC) and Hamming Code

4.6 Numerical

4.7 Channel Allocation Techniques (Multiple Access Techniques): Random Access, ALOHA, Pure ALOHA, Slotted ALOHA

4.8 Carrier Sense Multiple Access (CSMA): CSMA/CD and CSMA/CA

4.9 VLAN

4.1 Functions of Data Link Layer (DLL)

The data link layer is the second layer in the OSI (Open System Interconnection) network architecture model responsible for the node-to-node delivery of data within the same local network.

1. Framing

The data link layer organizes raw bits from the physical layer into structured data units called frames. It ensures proper synchronization between sender and receiver by adding headers and trailers to the data. Framing helps in identifying the beginning and end of a data packet and hence, preventing data loss or corruption.

2. Addressing

Data-link layer provides layer-2 hardware addressing mechanism. Hardware address is assumed to be unique on the link. It is encoded into hardware at the time of manufacturing.

3. Synchronization
When data frames are sent on the link, both machines must be synchronized in order to transfer to take place.

4. Error Control
Sometimes signals may have encountered problem in transition and the bits are flipped. These errors are detected and attempted to recover actual data bits. It also provides error reporting mechanism to the sender.

5. Flow Control
Stations on same link may have different speed or capacity. Data link layer ensures flow control that enables both machines to exchange data on same speed.

6. Multi-Access
When host on the shared link tries to transfer the data, it has a high probability of collision. Data-link layer provides mechanism such as CSMA/CD to equip capability of accessing a shared media among multiple Systems.

4.2 Overview of Logical Link Control (LLC) and Media Access Control (MAC)

Sub-Layers of The Data Link Layer

The data link layer is further divided into two sub-layers, which are as follows:

Logical Link Control (LLC)

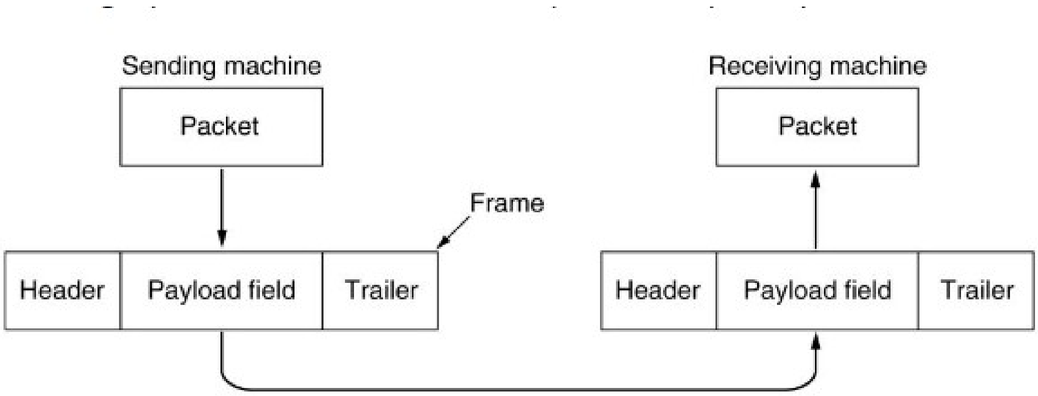
This sublayer of the data link layer deals with multiplexing, the flow of data among applications and other services, and LLC is responsible for providing error messages and acknowledgments as well. It deals with protocols, flow-control, and error control.

Media Access Control (MAC)

MAC sublayer manages the device's interaction, responsible for addressing frames, and also controls physical media access. The data link layer receives the information in the form of packets from the Network layer, it divides packets into frames and sends those frames bit-by-bit to the underlying physical layer. It deals with actual control of media.

4.3 Framing and its Types

Framing is a point-to-point connection between two computers or devices consisting of a wire in which data is transmitted as a stream of bits. However, these bits must be framed into discernible blocks of information. Framing is a function of the data link layer. It provides a way for a sender to transmit a set of bits that are meaningful to the receiver. Ethernet, token ring, frame relay, and other data link layer technologies have their own frame structures. Frames have headers that contain information such as error-checking codes.

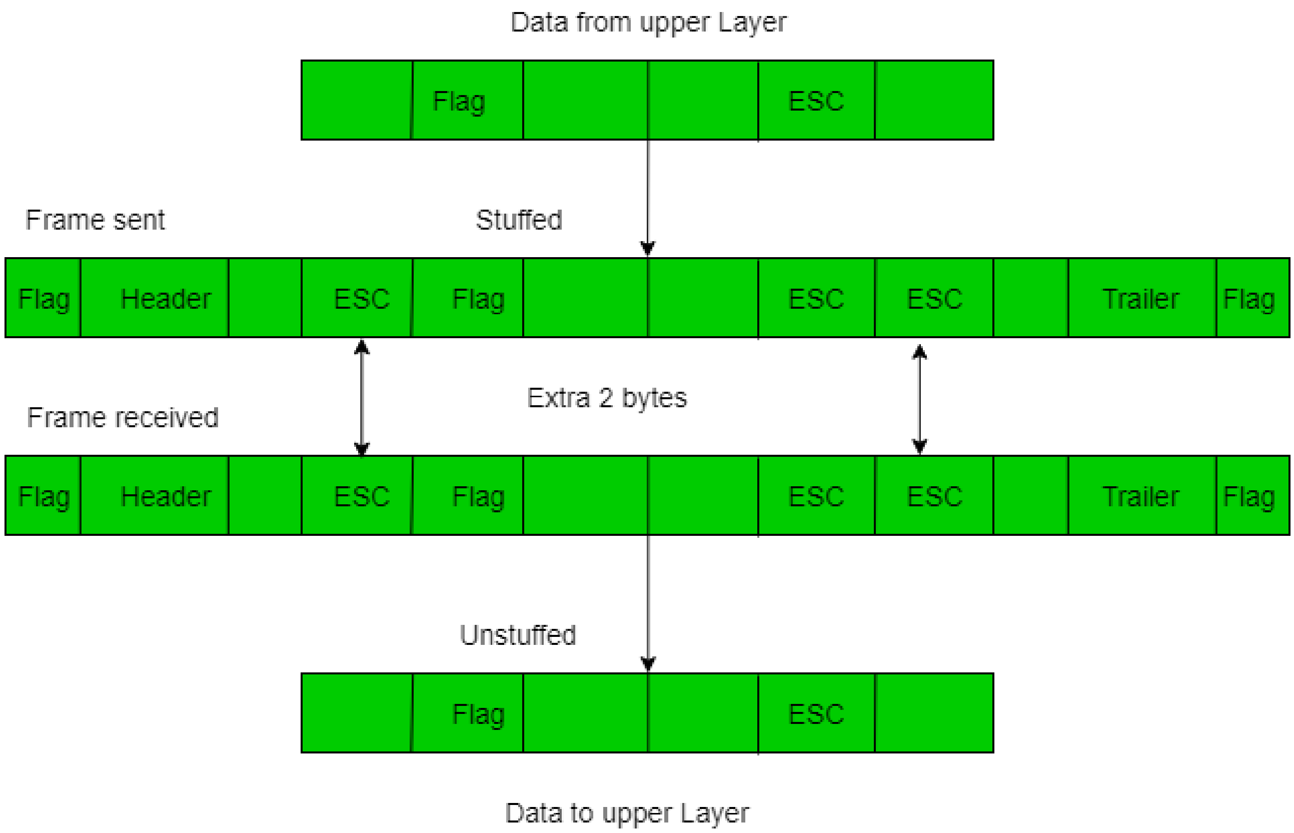


Since the physical layer merely accepts and transmits a stream of bits without any regard to meaning or structure, it is up to the data link layer to create and recognize frame boundaries. This can be accomplished by attaching special bit patterns to the beginning and end of the frame. If these bit patterns can accidentally occur in data, special care must be taken to make sure these patterns are not incorrectly interpreted as frame delimiters. The four framing methods that are widely used are :

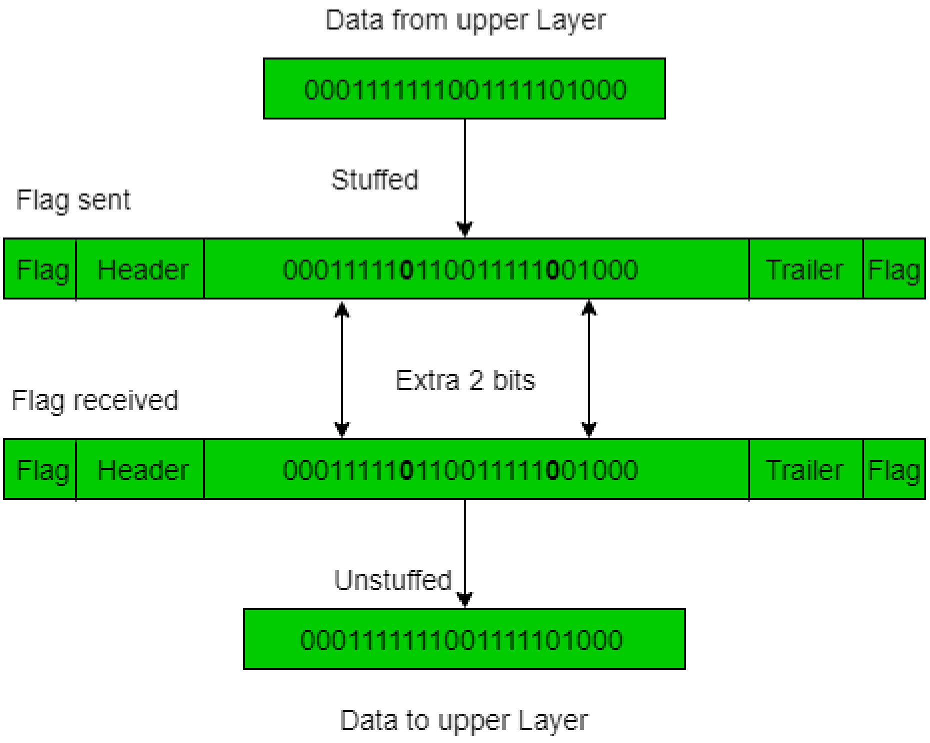
1. Character/Byte Stuffing
2. Bit Stuffing

1. Character/Byte Stuffing: Used when frames consist of characters. If data contains ED then, a byte is stuffed into data to differentiate it from ED.

Let ED = "$" --> if data contains '$' anywhere, it can be escaped using '\O' character.
--> if data contains '\O$' then, use '\O\O\O$'($ is escaped using \O and \O is escaped using \O).
Disadvantage - It is very costly and obsolete method.



2. Bit Stuffing: Let ED = 01111 and if data = 01111
--> Sender stuffs a bit to break the pattern i.e. here appends a 0 in data = 011101.
--> Receiver receives the frame.
--> If data contains 011101, receiver removes the 0 and reads the data.



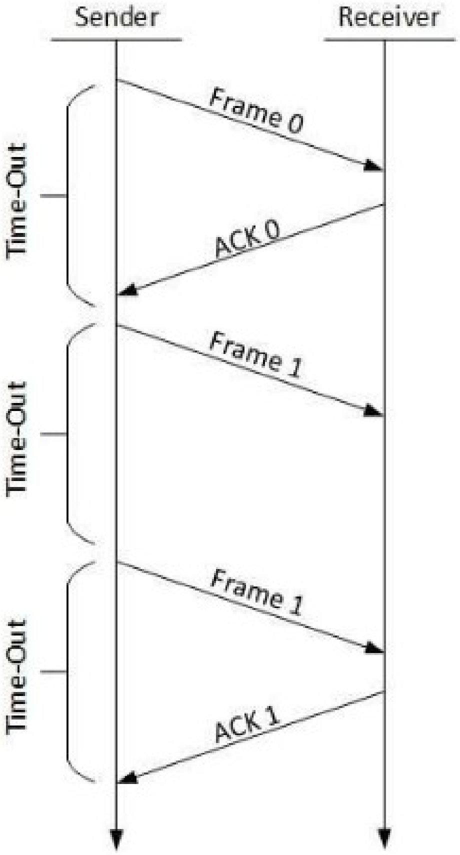
3. Frame Check Sequence (FCS)

FCS is a field added at the end of the frame used for error detection.
Common technique: Cyclic Redundancy Check (CRC).
The sender computes the FCS based on the data and sends it with the frame.
The receiver recomputes the FCS and compares it to detect any error.

4.4 Flow Control Mechanism: Simple Stop and Wait ARQ, Sliding Window, Go-Back-N ARQ, Selective Repeat ARQ

Flow control is the management of data flow between computers or devices or between nodes in a network so that the data can be handled at an efficient pace.

1. Stop-and-wait ARQ (Automatic Repeat Requests)

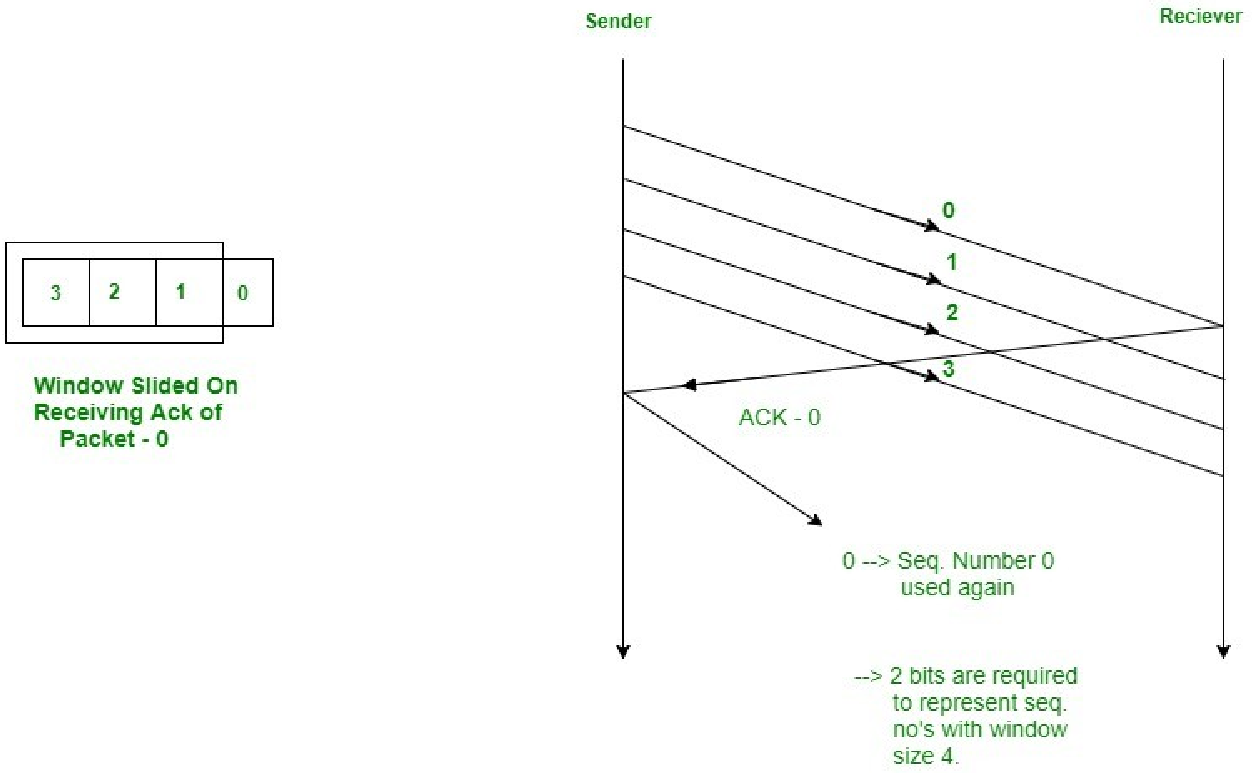


- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.

2. Sliding Window

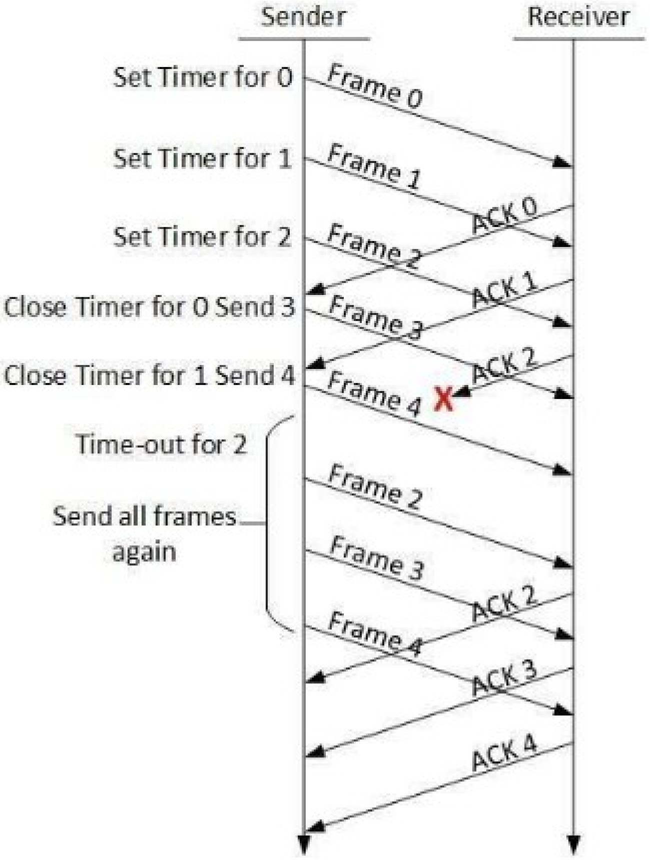
Sliding window is a technique for controlling transmitted data packets between two network computers where reliable and sequential delivery of data packets is required.

In this method, sender transmits or sends various frames or packets before receiving any acknowledgement. In this method, both the sender and receiver agree upon total number of data frames after which acknowledgement is needed to be transmitted. In this method sender sent multiple frame but receiver take one by one and after completing one frame acknowledge(which is next frame number only) for new frame.


Sender
Reciever
3
2
1
0
Window Slided On Receiving Ack of Packet - 0
0
1
2
3
ACK - 0
0 --> Seq. Number 0 used again
--> 2 bits are required to represent seq. no's with window size 4.

3. Go-Back-N ARQ

In Go-Back-N ARQ method, both sender and receiver maintain a window.
The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame’s sequence number.
When the sender sends all the frames in window, it checks up to what sequence number it has received positive acknowledgement. If all frames are positively acknowledged, the sender sends next set of frames. If sender finds that it has received NACK or has not received any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

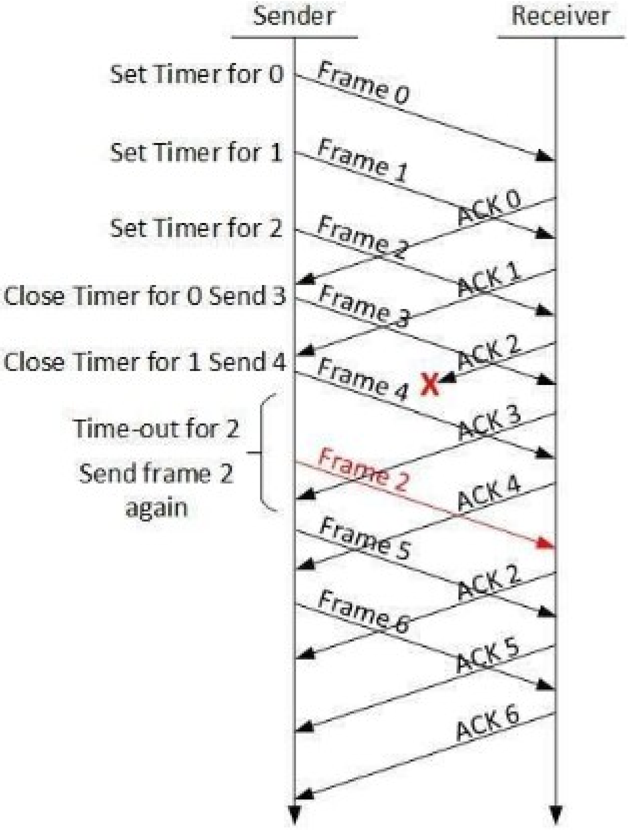


4. Selective Repeat ARQ

In Go-back-N ARQ, it is assumed that the receiver does not have any buffer space for its window size and has to process each frame as it comes. This enforces the sender to retransmit all the frames which are not acknowledged.

In Selective-Repeat ARQ, the receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.

The sender in this case, sends only packet for which NACK is received.

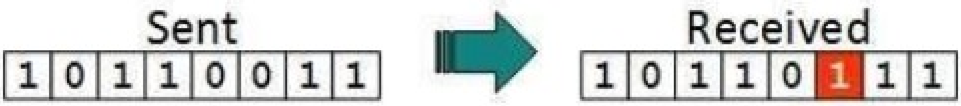


4.5 Error Detection and Correction Techniques: Parity Check, Checksum, Cyclic Redundancy Check (CRC) and Hamming Code

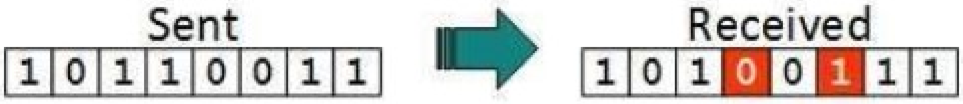
Types of Errors

There may be three types of errors:

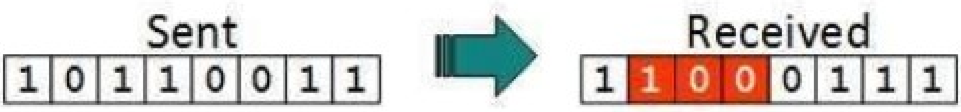
1. Single bit error: In a frame, there is only one bit, anywhere though, which is corrupt.



2. Multiple bits error: Frame is received with more than one bits in corrupted state.



3. Burst error: Frame contains more than 1 consecutive bits corrupted.



Error detection is the process of detecting the error during the transmission between the sender and the receiver.

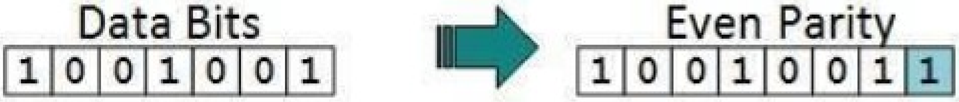
Types of error detection

1. Parity checking
2. Cyclic Redundancy Check (CRC)
3. Checksum

1. Parity checking

Parity adds a single bit that indicates whether the number of “1” bits in the preceding data is even or odd. If a single bit is changed in transmission, the message will change parity and the error can be detected at this point.It works as:

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

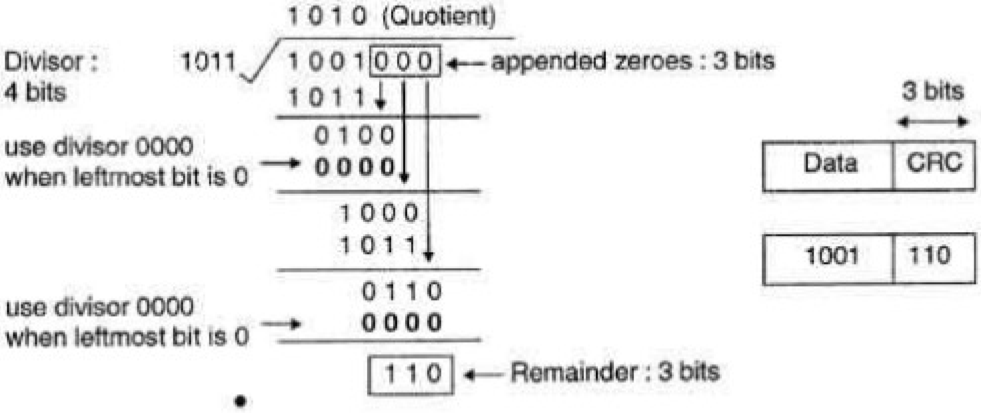


parity does not indicate which bit contained the error, even when it can detect it.

2. Cyclic Redundancy Check (CRC)

This technique involves binary division of the data bits being sent. The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder. Before sending the actual bits, the sender adds the remainder at the end of the actual bits. Actual data bits plus the remainder is called a code-word. The sender transmits data bits as code-words.

At the other end, the receiver performs division operation on code-words using the same CRC divisor. If the remainder contains all zeros the data bits are accepted, otherwise it is considered as there some data corruption occurred in transit.



CRC generated (Binary division)

Sequential steps in CRC are as follows:
We have given dataword of length n and divisor of length k.

Sender follows following steps.

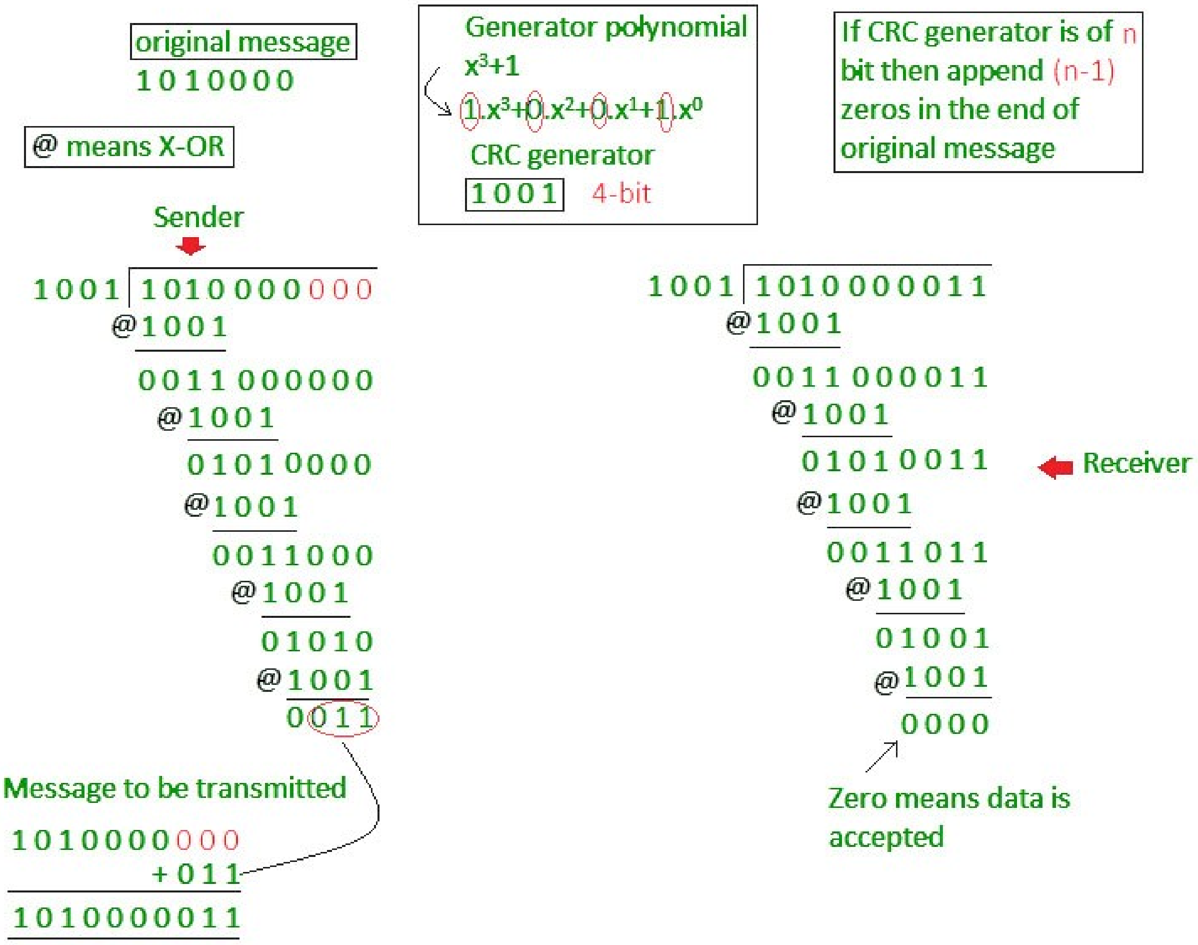
Data unit is composite by number of 0s, which is one less than the divisor. Step 1: Append (k-1) zero's to the original message.

Then it is divided by the predefined divisor using binary division technique. The remainder is called CRC. CRC is appended to the data unit and is sent to the receiver.

Receiver follows following steps.

When data unit arrives followed by the CRC it is divided by the same divisor which was used to find the CRC (remainder).

If the remainder result in this division process is zero then it is error free data, otherwise it is corrupted.



3. Checksum

Checksum error detection is a method used to identify errors in transmitted data. The process involves dividing the data into equally sized segments and using a 1's complement to calculate the sum of these segments. The calculated sum is then sent along with the data to the receiver. At the receiver's end, the same process is repeated and if all zeroes are obtained in the sum, it means that the data is correct.

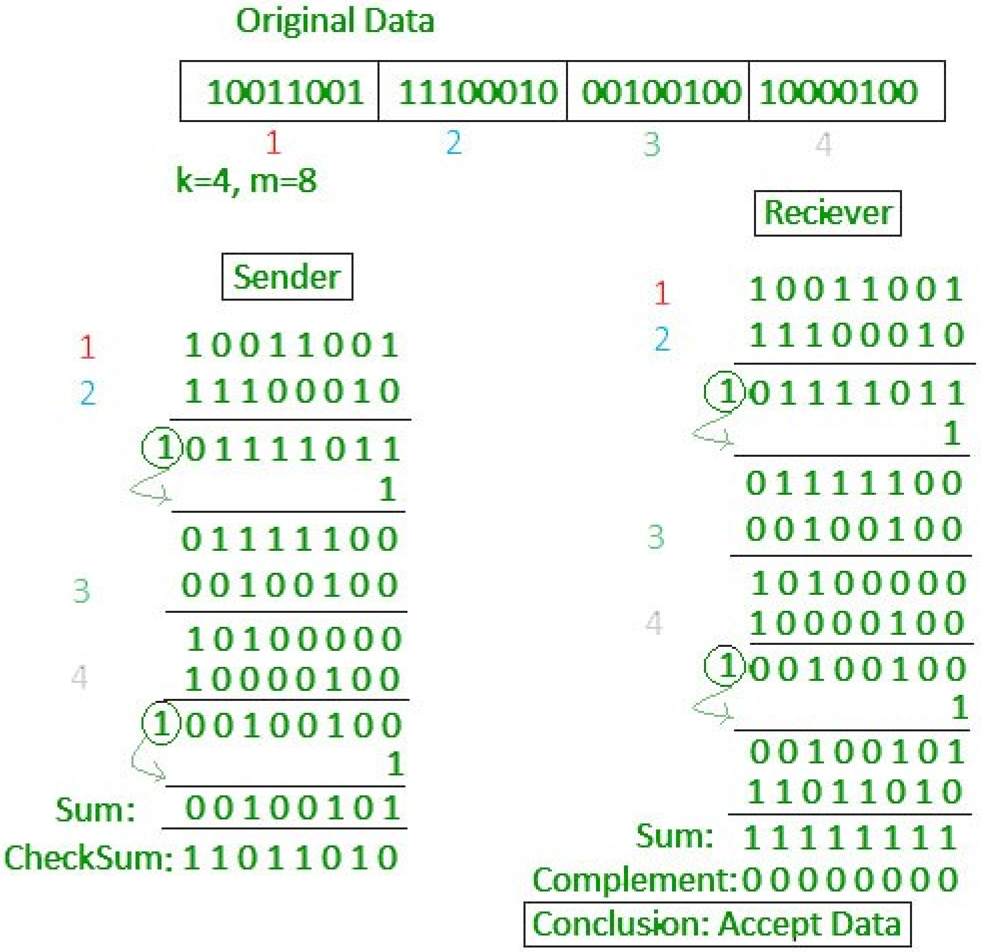
Checksum - Operation at Sender's Side

Firstly, the data is divided into k segments each of m bits.

On the sender’s end, the segments are added using 1’s complement arithmetic to get the sum. The sum is complemented to get the checksum.

The checksum segment is sent along with the data segments.

Checksum - Operation at Receiver's Side



At the receiver’s end, all received segments are added using 1’s complement arithmetic to get the sum. The sum is complemented.

If the result is zero, the received data is accepted; otherwise discarded.

Error Correction

In the digital world, error correction can be done in two ways:

Backward Error Correction: When the receiver detects an error in the data received, it requests back the sender to retransmit the data unit. In many cases, the request is implicit; the receiver sends an acknowledgement (ACK) of correctly received data, and the transmitter re-sends anything not acknowledged within a reasonable period of time. This mechanism is also called Automatic Repeat Request (ARQ).

Forward Error Correction: When the receiver detects some error in the data received, it executes error-correcting code, which helps it to auto-recover and to correct some kinds of errors.

Hamming Code

Hamming code is a set of error-correction codes that can be used to detect and correct bit errors that can occur when computer data is moved or stored.

The hamming code can be applied to data units of any length and uses the relationship between data and redundancy bits.

Calculating the Hamming Code

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:

1. Mark all bit positions that are powers of two as parity bits. (Positions 1, 2, 4, 8, 16, 32, 64, etc.)

2. 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.)

3. 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.

Position 1: check 1 bit, skip 1 bit, check 1 bit, skip 1 bit, etc. (1,3,5,7,9,11,13,15,...)

Position 2: check 2 bits, skip 2 bits, check 2 bits, skip 2 bits, etc (2,3,6,7,10,11,14,15,...)

Position 4: check 4 bits, skip 4 bits, check 4 bits, skip 4 bits, etc. (4,5,6,7,12,13,14,15,20,21,22,23,...)

Position 8: check 8 bits, skip 8 bits, check 8 bits, skip 8 bits, etc. (8-15,24-31,40-47,...)

Position 16: check 16 bits, skip 16 bits, check 16 bits, skip 16 bits, etc. (16-31,48-63,80-95,...)

Position 32: check 32 bits, skip 32 bits, check 32 bits, skip 32 bits, etc. (32-63,96-127,160-191,...)

etc.

4. 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.

Here is an example:

A byte of data: 10011010

Create the data word, leaving spaces for the parity bits: _ _ 1 _ 0 0 1 _ 1 0 1 0

Calculate the parity for each parity bit (a ? represents the bit position being set):

Position 1 checks bits 1,3,5,7,9,11:

? _ 1 _ 0 0 1 _ 1 0 1 0. Even parity so set position 1 to a 0: 0 _ 1 _ 0 0 1 _ 1 0 1 0

Position 2 checks bits 2,3,6,7,10,11:

0 ? 1 _ 0 0 1 _ 1 0 1 0. Odd parity so set position 2 to a 1: 0 1 1 _ 0 0 1 _ 1 0 1 0

Position 4 checks bits 4,5,6,7,12:

0 1 1 ? 0 0 1 _ 1 0 1 0. Odd parity so set position 4 to a 1: 0 1 1 1 0 0 1 _ 1 0 1 0

Position 8 checks bits 8,9,10,11,12:

0 1 1 1 0 0 1 ? 1 0 1 0. Even parity so set position 8 to a 0: 0 1 1 1 0 0 1 0 1 0 1 0

Code word: 011100101010.

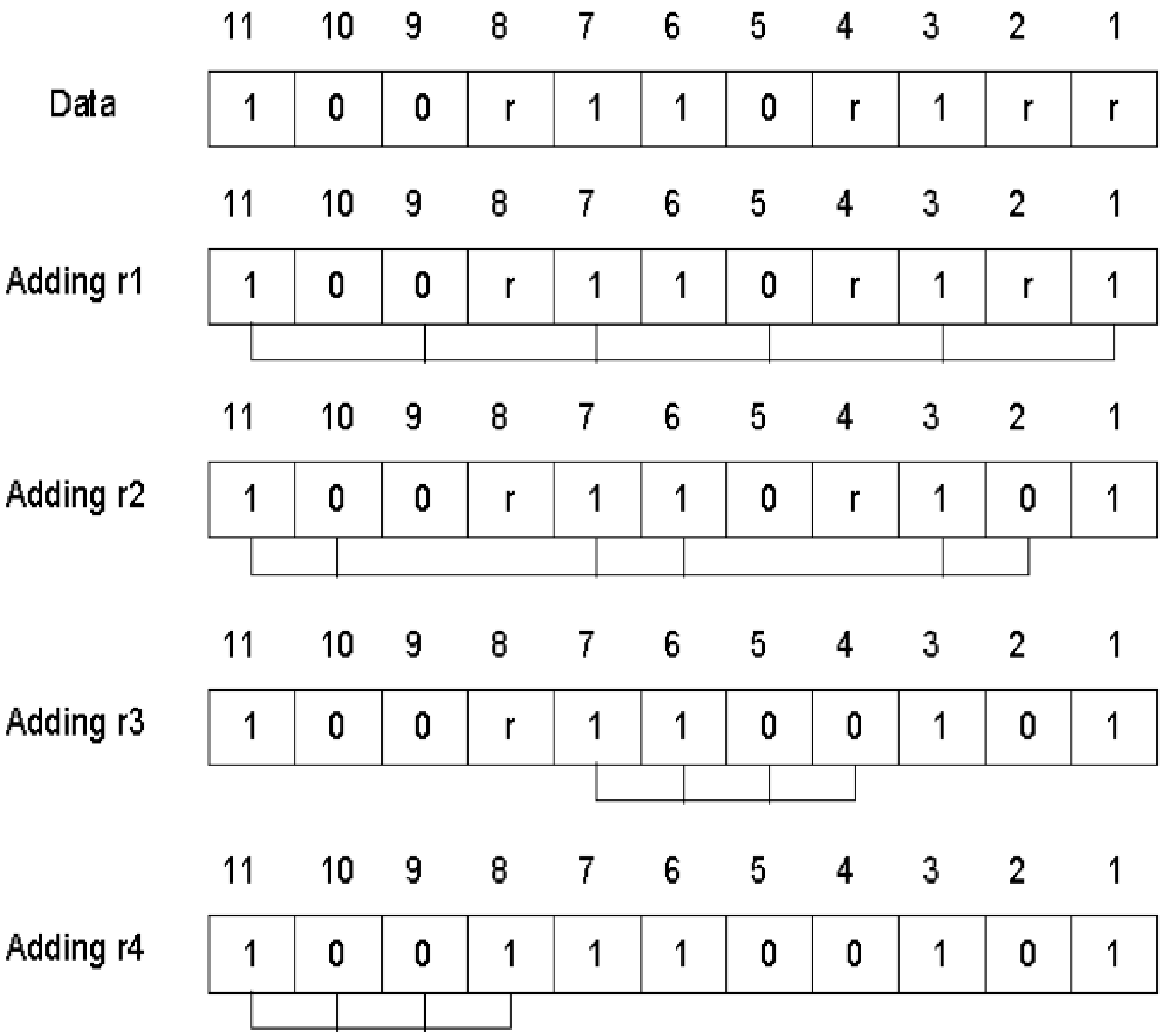
Error Detection and Correction

Example:

At the sender:

Data to be sent: 1001101

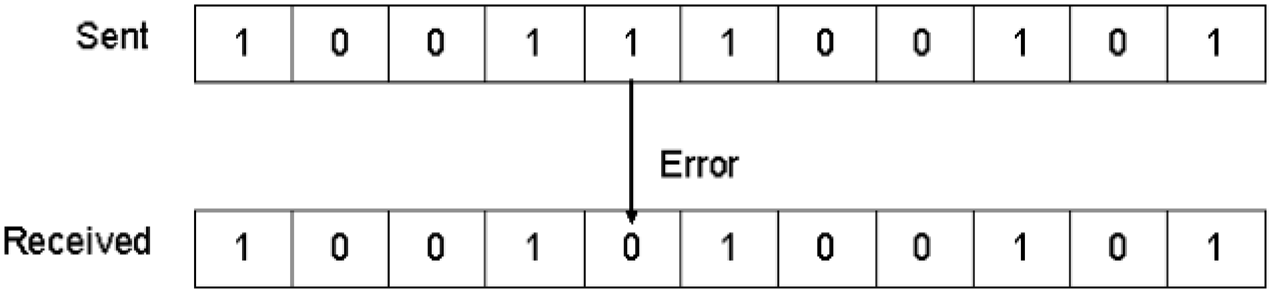
Redundancy bit calculation is shown below.



	11	10	9	8	7	6	5	4	3	2	1
Data	1	0	0	r	1	1	0	r	1	r	r
Adding r1	1	0	0	r	1	1	0	r	1	r	1
Adding r2	1	0	0	r	1	1	0	r	1	0	1
Adding r3	1	0	0	r	1	1	0	0	1	0	1
Adding r4	1	0	0	1	1	1	0	0	1	0	1

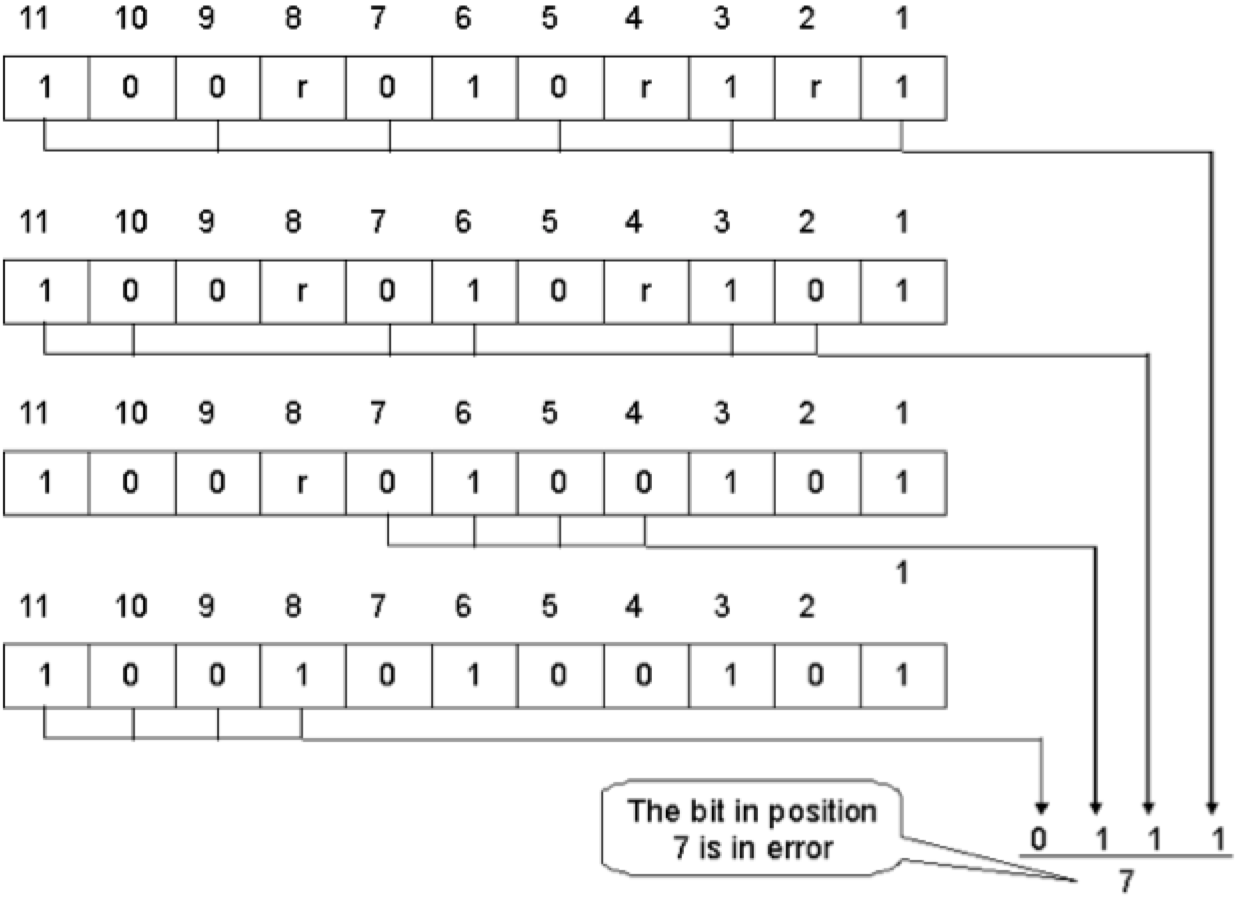
Data sent with redundancy bits: 10011100101

During transmission:



At the receiver:

The receiver takes the transmission and recalculates four new r values using the same set of bits used by the sender plus the relevant parity (r) bit for each set. Then it assembles the new parity values into a binary number in order of r position (r8, r4, r2, r1).
Once the bit is identified, the receiver can reverse its value and correct the error.



4.7 Channel Allocation Techniques (Multiple Access Techniques): Random Access, ALOHA, Pure ALOHA, Slotted ALOHA

Channel allocation techniques are used to allow multiple users to share the same communication medium efficiently.
Multiple Access Protocols are methods used in computer networks to control how data is transmitted when multiple devices are trying to communicate over the same network. These protocols ensure that data packets are sent and received efficiently, without collisions or interference. They help manage the network traffic so that all devices can share the communication channel smoothly and effectively.

1. Random Access Protocol

In this, all stations have same superiority that is no station has more priority than another station. Any station can send data depending on medium's state(idle or busy). It has two features:

There is no fixed time for sending data
There is no fixed sequence of stations sending data

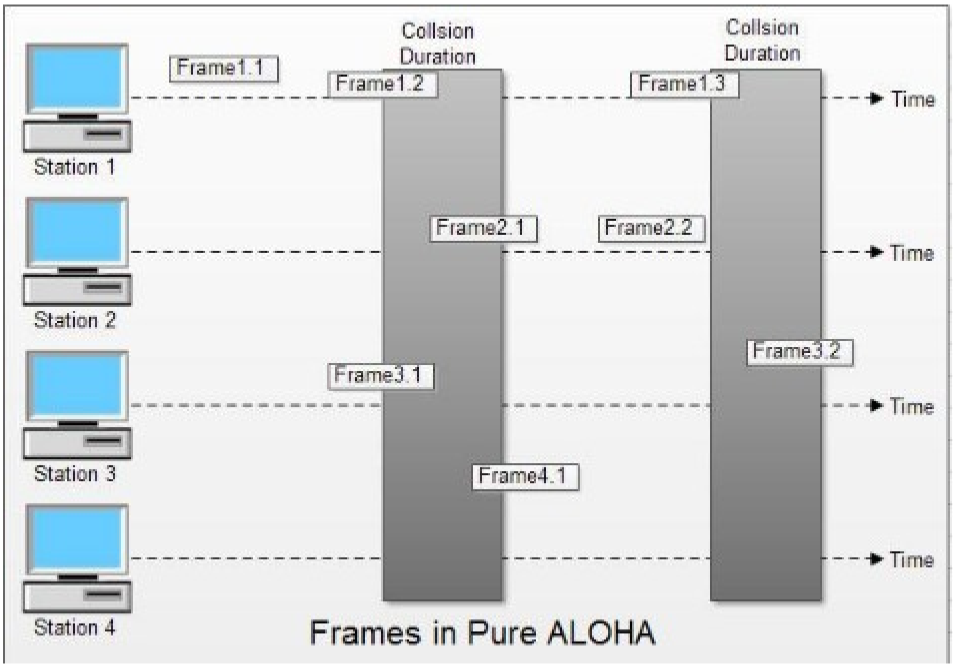
The Random access protocols are further subdivided as:

ALOHA

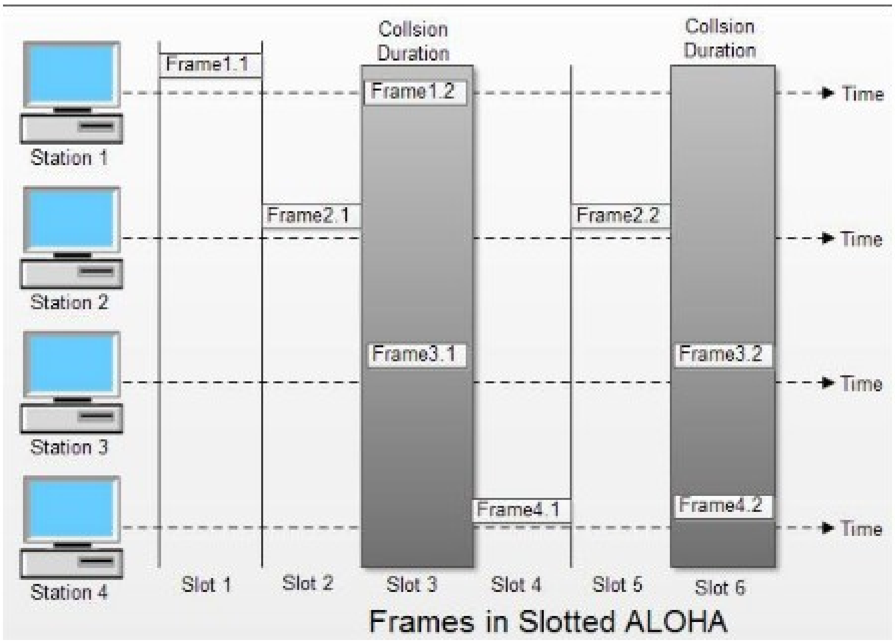
A shared communication system like ALOHA requires a method of handling collisions that occur when two or more systems attempt to transmit on the channel at the same time. In this, multiple stations can transmit data at the same time and can hence lead to collision and data being garbled. Aloha is a multiple access protocol at the data link layer and proposes how multiple terminals access the medium without interference or collision.

Pure ALOHA

When a station sends data it waits for an acknowledgement. If the acknowledgement doesn't come within the allotted time then the station waits for a random amount of time called back-off time (Tb) and re-sends the data. Since different stations wait for different amount of time, the probability of further collision decreases.



Slotted ALOHA



It is similar to pure aloha, except that we divide time into slots and sending of data is allowed only at the beginning of these slots. If a station misses out the allowed time, it must wait for the next slot. This reduces the probability of collision.

4.8 Carrier Sense Multiple Access (CSMA): CSMA/CD and CSMA/CA

Carrier Sense Multiple Access (CSMA) is a method used in computer networks to help devices share a communication channel without interfering with each other.

1. Carrier Sense Multiple Access/Collision Detection (CSMA/CD)

CSMA/CD is a Media Access Control (MAC) protocol. It defines how network devices respond when two devices attempt to use a data channel simultaneously and encounter a data collision. The CSMA/CD rules define how long the device should wait if a collision occurs.
CSMA/CD (carrier sense multiple access/collision detection) CD (collision detection) defines what happens when two devices sense a clear channel, then attempt to transmit at the same time. A collision occurs, and both devices stop transmission, wait for a random amount of time, and then retransmit. This is the technique used to access the 802.3 Ethernet network channel.

2. Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

CSMA/CA is a technique used mainly in wireless networks to avoid collisions when multiple devices try to send data. Unlike wired networks where a device can detect collisions easily, in wireless networks, most of the energy goes into transmitting, so a device cannot sense if a collision has happened during transmission. To solve this, CSMA/CA prevents collisions instead of detecting them, making it ideal for technologies like Wi-Fi.

4.9 VLAN

VLAN (Virtual Local Area Network)

A VLAN is a logical subgroup within a Local Area Network (LAN) that combines a group of devices from different physical locations into a single broadcast domain.

- 🔹 Definition:

VLAN is a network configuration that allows logical segmentation of a LAN into smaller, isolated broadcast domains, regardless of physical location.

- 🔹 Role of VLAN in Logical Segmentation of LAN:

Logical Grouping:

VLANs allow users or devices to be grouped logically (e.g., by department) rather than physically.

Reduces Broadcast Traffic:

Broadcasts in one VLAN do not reach devices in another VLAN, reducing unnecessary traffic.

Improves Security:

Devices in different VLANs cannot communicate directly unless through a router or Layer 3 switch, improving isolation and security.

Better Network Management:

Makes it easier to manage and troubleshoot networks by organizing users logically.

Flexibility:

Devices can be moved physically without changing their VLAN configuration — just plug them into any switch port and assign the same VLAN.

🔹 Example Use Case:

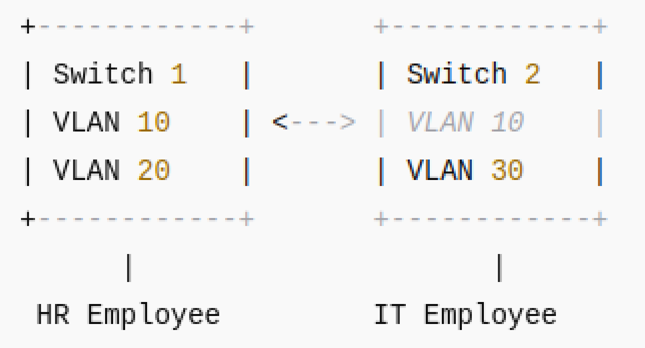
In a company:

VLAN 10: HR Department

VLAN 20: Finance Department

VLAN 30: IT Department

Even if users are on different floors or buildings, VLAN ensures they operate in their own isolated broadcast domains.



```
+------------+       +------------+
| Switch 1   |       | Switch 2   |
| VLAN 10    | <---> | VLAN 10    |
| VLAN 20    |       | VLAN 30    |
+------------+       +------------+
      |                     |
 HR Employee          IT Employee
```