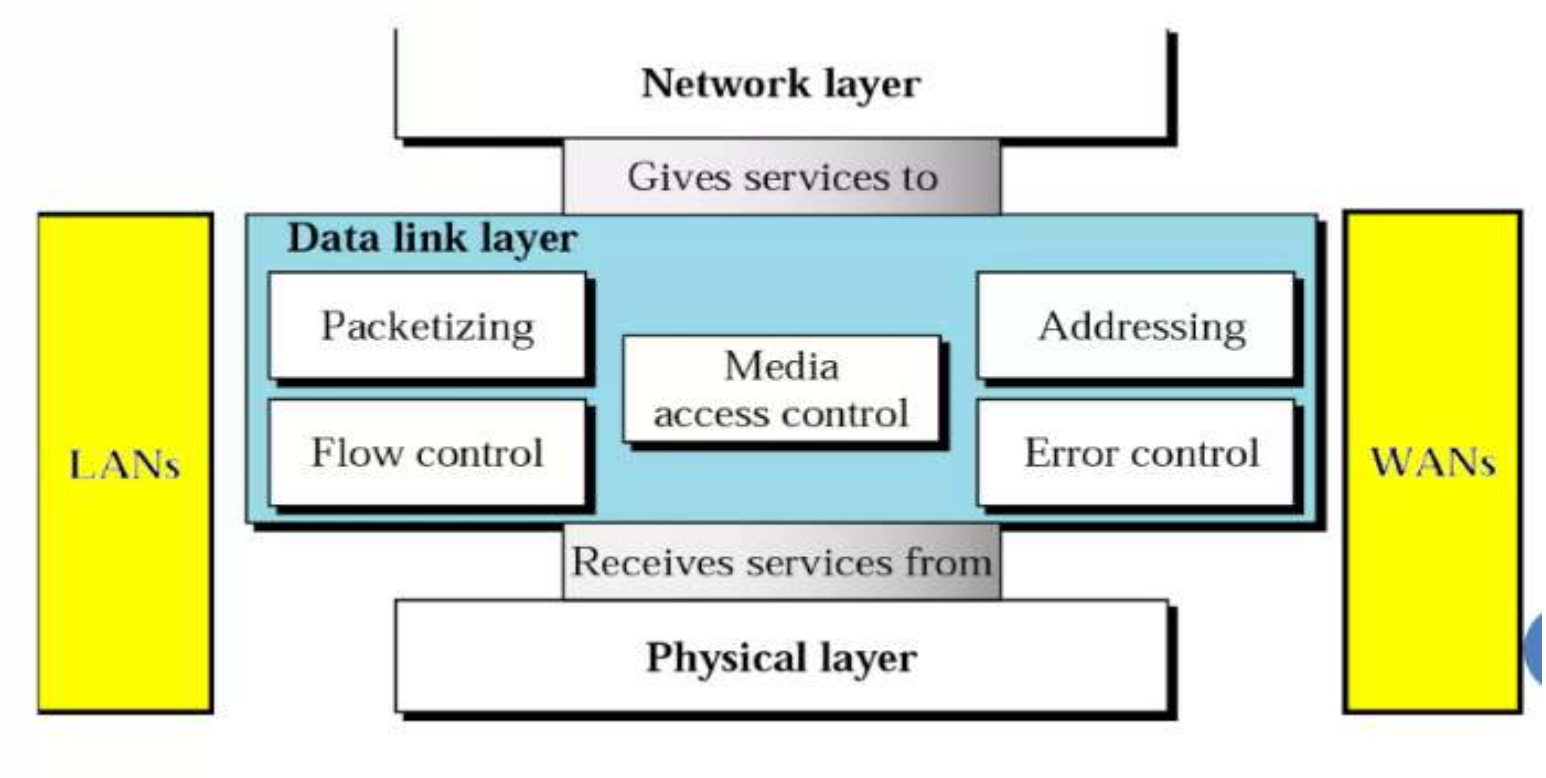
DATA LINK LAYER

The Data-link layer is the **second layer from the bottom in the OSI (Open System Interconnection) network architecture model**. It is responsible for the node-to-node delivery of data. Its major role is to ensure error-free transmission of information.



The data link layer in the OSI (Open System Interconnections) Model, is in between the physical layer and the network layer.

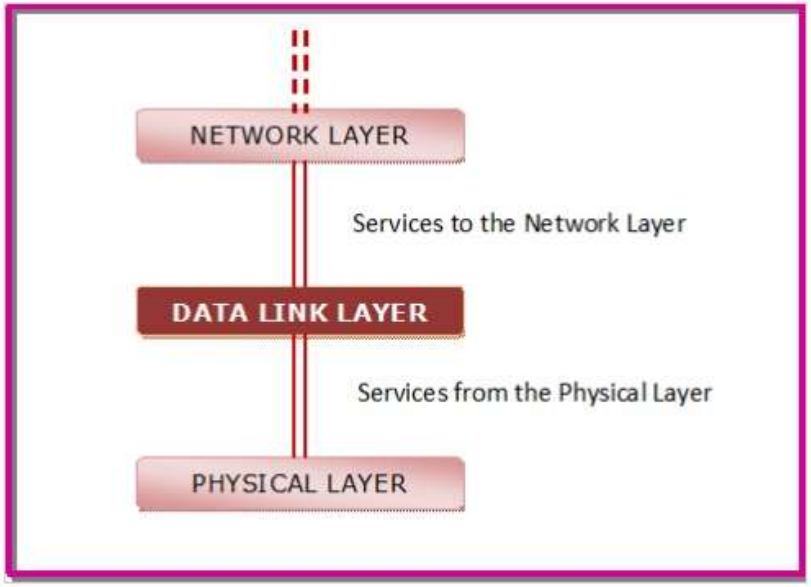
The main functions and the design issues of this layer are

- Providing services to the network layer
- Framing
- Error Control
- Flow Control

Services to the Network Layer

In the OSI Model, each layer uses the services of the layer below it and provides services to the layer above it.

The data link layer uses the services offered by the physical layer. The primary function of this layer is to provide a well defined service interface to network laver above it.



The types of services provided can be of three types −

- Unacknowledged connectionless service
- Acknowledged connectionless service
- Acknowledged connection - oriented service

1) Unacknowledged connectionless service

- As the name suggests, it is unacknowledged form of transmission. Here the source machine sends the data to the destination machine without any acknowledgement. For this, no connection is either established or released. If the data is lost due to noise or interference, the lost data is not even recovered by the layer.

2) Acknowledged connectionless service

- In acknowledged connectionless service each data frame is acknowledged by the destination machine. If any data frame is lost or not arrived in time the same can be transmitted again. In this service no connection are used.

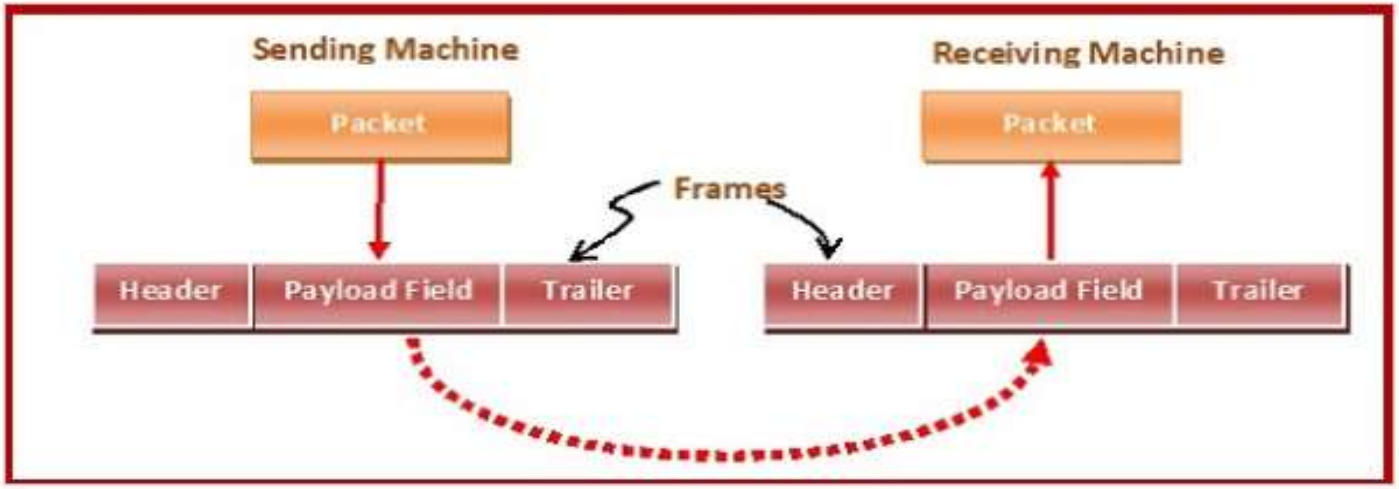
3) Acknowledged connection-oriented service

- Acknowledged connection service establishes a connection prior to data transmission. Each frame is numbered before transmission and corresponding acknowledgement is also received. The transmission is carried out in distinct phases.

Framing

Framing in the data link layer separates a message from one source to a destination by adding a sender address & destination address.

- A frame has three parts, namely −
- Frame Header
- Payload field that contains the data packet from network layer
- Trailer



Error Control

- To ensure the proper sequencing and safe delivery of frames at the destination, an acknowledgement should be sent by the destination network. The receiver sends back special control frames bearing positive or negative acknowledgements about the incoming frames.
- If the sender receives a positive acknowledgement it means the frame has arrived safely.
- If a negative acknowledgement arrives that means, something has gone wrong and the frame is to be retransmitted.
- A timer at sender's and receiver's end is introduced. Also sequence numbers to the outgoing frames are maintained so that the receiver can distinguish retransmissions from originals. This is one of the most important part of data link layer duties.

Flow Control

The data link layer regulates flow control so that a fast sender does not drown a slow receiver. When the sender sends frames at very high speeds, a slow receiver may not be able to handle it. There will be frame losses even if the transmission is error-free. The two common approaches for flow control are −

- Feedback based flow control
- Rate based flow control

Feedback-based Flow Control

- Receiver sends back information to the sender giving it permission to send more data, or Telling sender how receiver is doing.

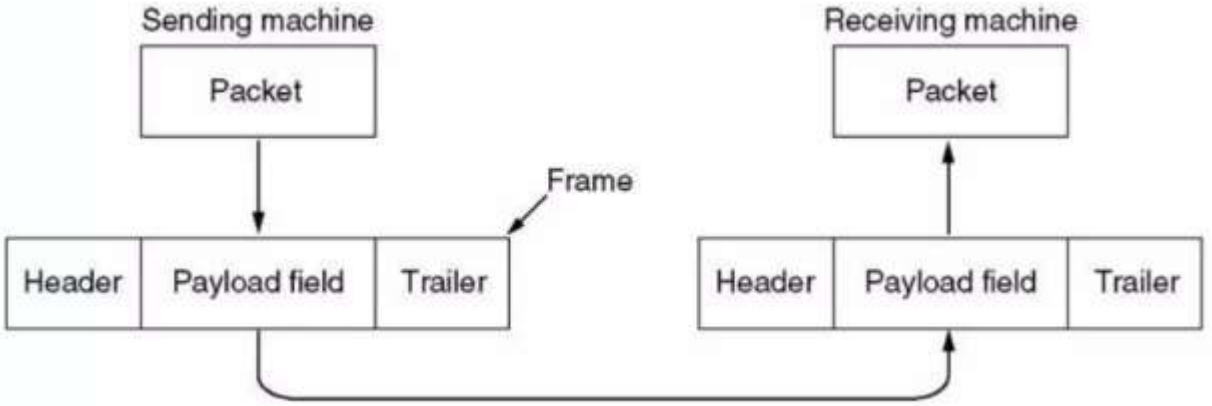
Rate-based Flow Control

- Built in mechanism that limits the rate at which sender may transmit data, without the need for feedback from the receiver.

DLL Designing Issues

- Providing a well-defined service interface to the network layer.
- Dealing with transmission errors.
- Regulating the flow of data so that slow receivers are not swamped by fast senders

For this, the data link layer takes the packets it gets from the network layer and encapsulates them into frames for transmission. Each frame contains a frame header, a payload field for holding the packet, and a frame trailer



Error

- Error is a condition when the output information does not match with the input information.
- During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from one system to other. That means a 0 bit may change to 1 or a 1 bit may change to 0.

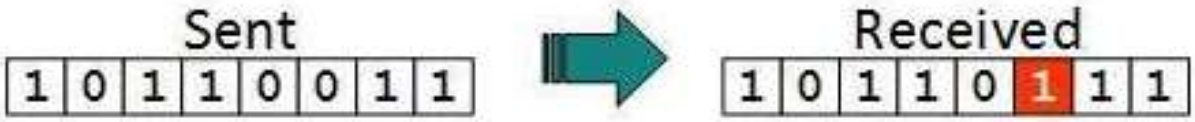
Errors Occur due to

- Noise
- electro magnetic interference
- Atmospheric conditions

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.

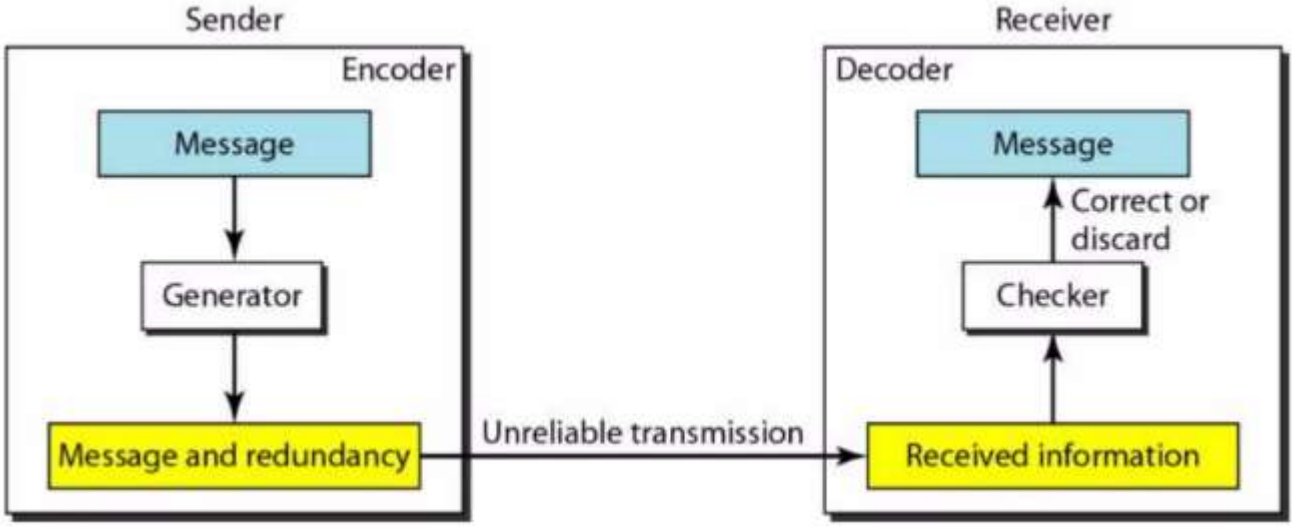
Error Correction and Detection

- It is physically impossible for any data recording or transmission medium to be 100% perfect 100% of the time over its entire expected useful life.
  - In data communication, line noise is a fact of life (e.g., signal attenuation, natural phenomenon such as lightning, and the telephone repairman).
- As more bits are packed onto a square centimeter of disk storage, as communications transmission speeds increase, the likelihood of error increases-- sometimes geometrically.
- Thus, error detection and correction is critical to accurate data transmission, storage and retrieval.
- Detecting and correcting errors requires *redundancy* -- sending additional information along with the data.

https://www.youtube.com/watch?v=EMrY-8m8D1E

ERROR DETECTION Vs ERROR CORRECTION

- There are two types of attacks against errors:
- **Error Detecting Codes:** Include enough redundancy bits to *detect* errors and use ACKs and retransmissions to recover from the errors.
- **Error Correcting Codes:** Include enough redundancy to detect *and* correct errors. The use of error-correcting codes is often referred to as forward error correction.



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.

Cyclic Redundancy Checks (CRCs)

The Cyclic Redundancy Checks (CRC) is the most powerful method for Error-Detection and Correction. It is given as a k-bit 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.

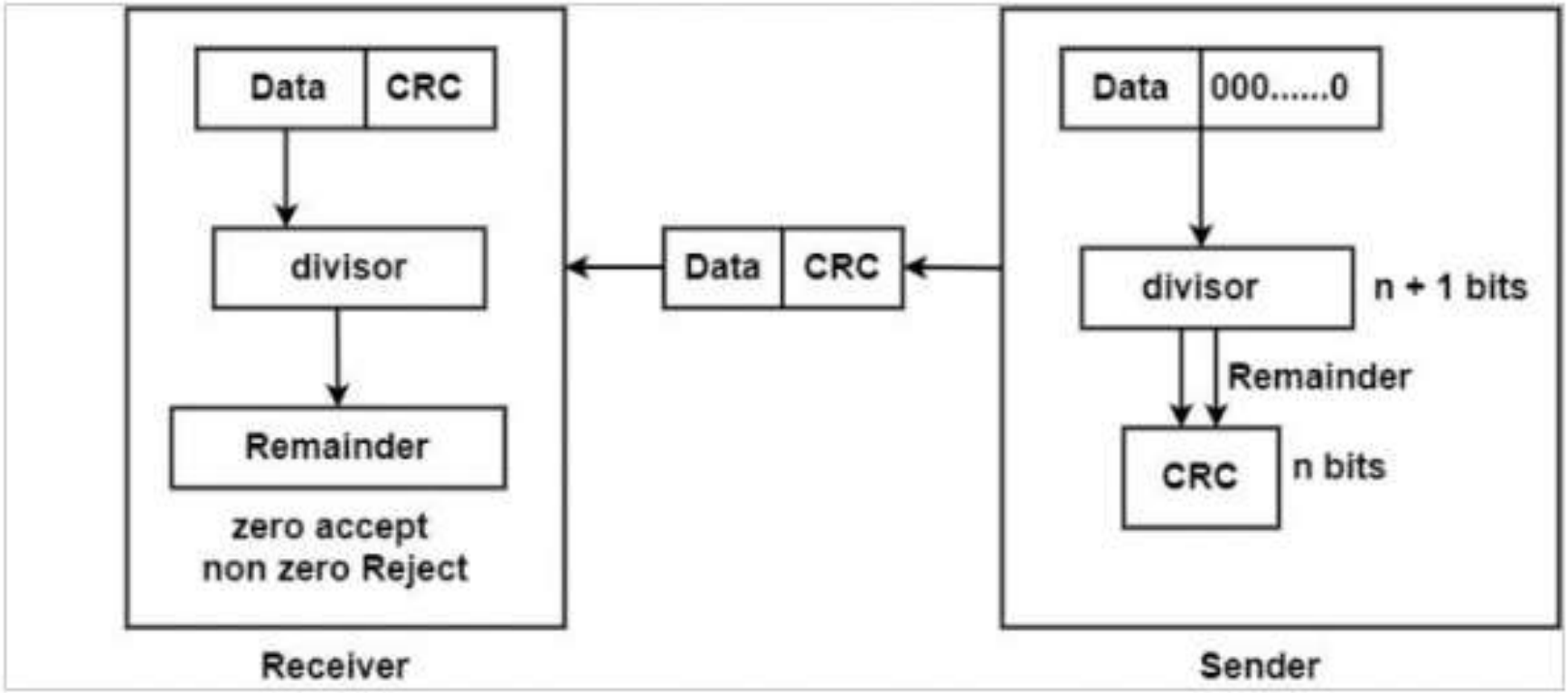
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Redundancy means **duplicity.** The redun l 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.
- https://www.youtube.com/watch?v=A9g6rTMblz4&list=PLBlnK6fEyqRhstjOChz8zuHiFoKGPMr9v&index=15

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.

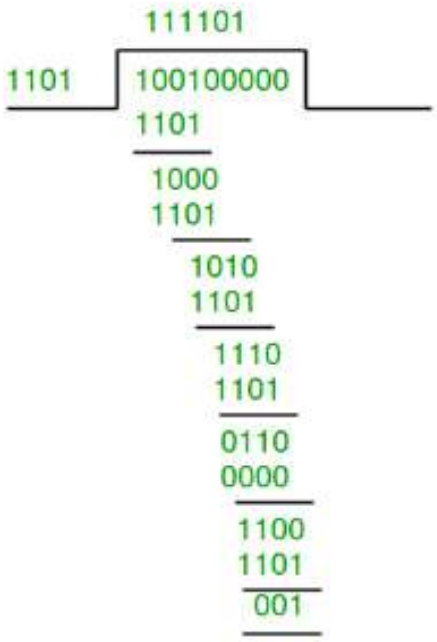
The CRC of n bits interpreted in phase 2 restores the added 0s at the end of the data unit.

Example 1 (No error in transmission):

Data word to be sent - 100100

Key - 1101 [Or generator polynomial $x^3 + x^2 + 1$]

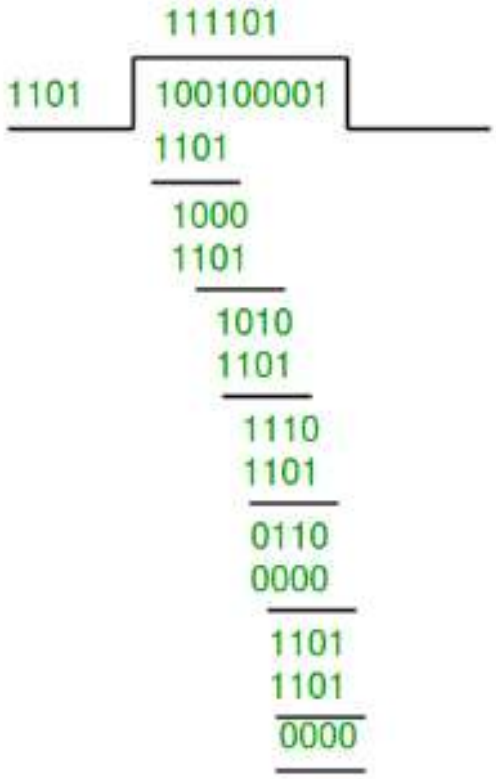
Sender Side:



Therefore, the remainder is 001 and hence the encoded data sent is 100100001.

Receiver Side:

Code word received at the receiver side 100100001

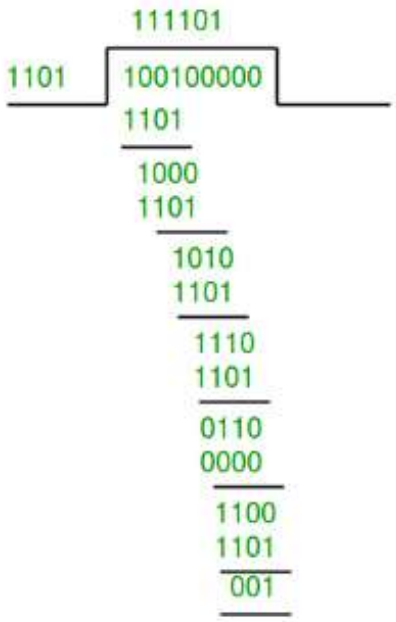


Therefore, the remainder is all zeros. Hence, the data received has no error.

Example 2: (Error in transmission)

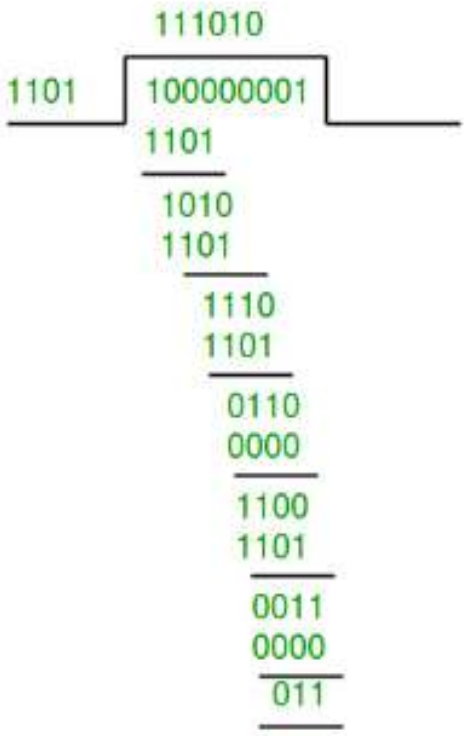
```
Data word to be sent - 100100
Key - 1101

Sender Side:
```




```
Therefore, the remainder is 001 and hence the
code word sent is 100100001.
```

```
Receiver Side
Let there be an error in transmission media
Code word received at the receiver side - 100000001
```

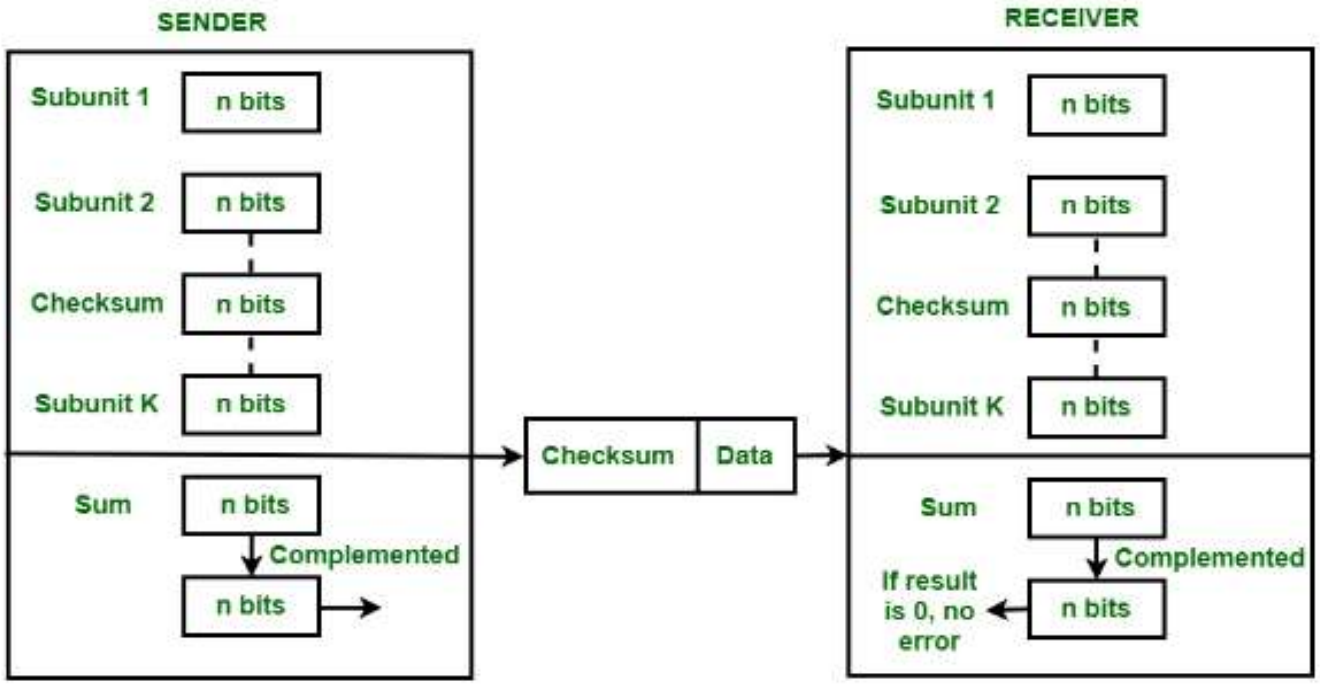


Since the remainder is not all zeroes, the error is detected at the receiver side.

Checksum

Checksum is the error detection method used by upper layer protocols and is considered to be more reliable than LRC, VRC and CRC. This method makes the use of **Checksum Generator** on Sender side and **Checksum Checker** on Receiver side.

At the Sender side, the data is divided into equal subunits of n bit length by the checksum generator. This bit is generally of 16-bit length. These subunits are then added together using one's complement method. This sum is of n bits. The resultant bit is then complemented. This complemented sum which is called checksum is appended to the end of original data unit and is then transmitted to Receiver.



The Receiver after receiving data + checksum passes it to checksum checker. Checksum checker divides this data unit into various subunits of equal length and adds all these subunits. These subunits also contain checksum as one of the subunits. The resultant bit is then complemented. If the complemented result is zero, it means the data is error-free. If the result is non-zero it means the data contains an error and Receiver rejects it.

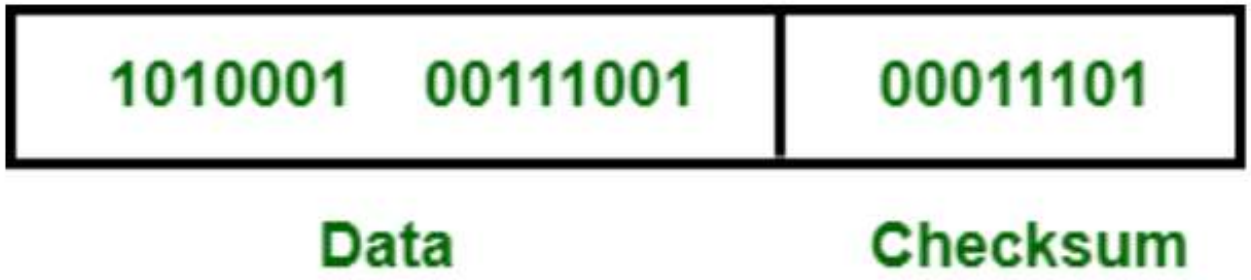
Example –

If the data unit to be transmitted is 10101001 00111001, the following procedure is used at Sender site and Receiver site.

Sender Site :

```
10101001        subunit 1
00111001        subunit 2
11100010        sum (using 1s complement)
00011101        checksum (complement of sum)
```

Data transmitted to Receiver is –



Receiver Site :

```
10101001        subunit 1
00111001        subunit 2
00011101        checksum
11111111        sum
00000000        sum's complement

Result is zero, it means no error.
```

Advantage :

The checksum detects all the errors involving an odd number of bits as well as the error involving an even number of bits.

Disadvantage :

The main problem is that the error goes undetected if one or more bits of a subunit is damaged and the corresponding bit or bits of a subunit are damaged and the corresponding bit or bits of opposite value in second subunit are also damaged. This is because the sum of those columns remains unchanged.

Example –

If the data transmitted along with checksum is 10101001 00111001 00011101. But the data received at destination is **0**0101001 **1**0111001 00011101.

Receiver Site :

```
00101001            1st bit of subunit 1 is damaged
10111001            1st bit of subunit 2 is damaged
00011101            checksum
11111111            sum
00000000            Ok 1's complement
```

Although data is corrupted, the error is undetected.

CHECKSUM VS CRC

- CRC is more thorough as opposed to Checksum in checking for errors and reporting.
- Checksum is the older of the two programs.
- CRC has a more complex computation as opposed to checksum.
- Checksum mainly detects single-bit changes in data while CRC can check and detect double-digit errors.
- CRC can detect more errors than checksum due to its more complex function.
- A checksum is mainly employed in data validation when implementing software.
- A CRC is mainly used for data evaluation in analogue data transmission.

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.
- **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.

The first one, Backward Error Correction, is simple and can only be efficiently used where retransmitting is not expensive. For example, fiber optics. But in case of wireless transmission retransmitting may cost too much. In the latter case, Forward Error Correction is used.

To correct the error in data frame, the receiver must know exactly which bit in the frame is corrupted. To locate the bit in error, redundant bits are used as parity bits for error detection.For example, we take ASCII words (7 bits data), then there could be 8 kind of information we need: first seven bits to tell us which bit is error and one more bit to tell that there is no error.

For m data bits, r redundant bits are used. r bits can provide 2r combinations of information. In m+r bit codeword, there is possibility that the r bits themselves may get corrupted. So the number of r bits used must inform about m+r bit locations plus no-error information, i.e. m+r+1.

$$2^r >= m+r+1$$

Hamming Code

In Computer Networks, Hamming code is used for the set of error-correction codes which may occur when the data is moved from the sender to the receiver. The hamming method corrects the error by finding the state at which the error has occurred.

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 redundancy bits are placed at certain calculated positions to eliminate the errors and the distance between the two redundancy bits is called "Hamming Distance".

Error Correction Code − This is the relationship between data bits and redundancy bits to correct a single-bit error. A-frame consists of M data bits and R redundant bits. Suppose the total length of the frame be N (N=M+R). An N-bit unit containing data and the check bit is often referred to as an N-bit codeword.

The following formula is used to find the number of redundant bits.

Number of single-bit errors = M + R

Number of states for no error = 1

So, the number of redundant bits (R) that represent all states (M+R+1) must satisfy −

$$2^R \geq M + R + 1$$

where R = Redundant bit, and M = data bit.

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.

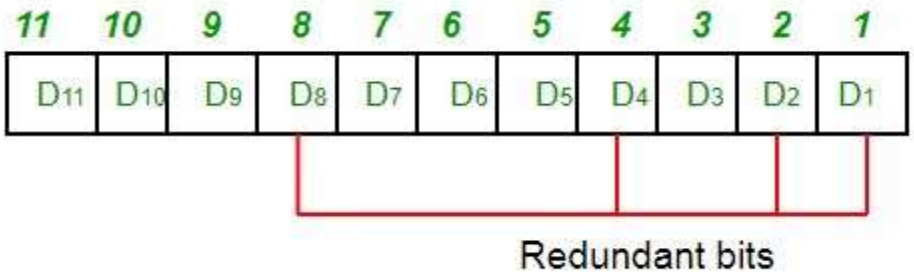
As in the above example:

•The number of data bits = 7

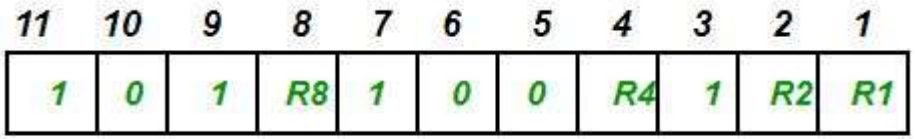
•The number of redundant bits = 4

•The total number of bits = 11

•The redundant bits are placed at positions corresponding to power of 2 i.e.1, 2, 4, and 8



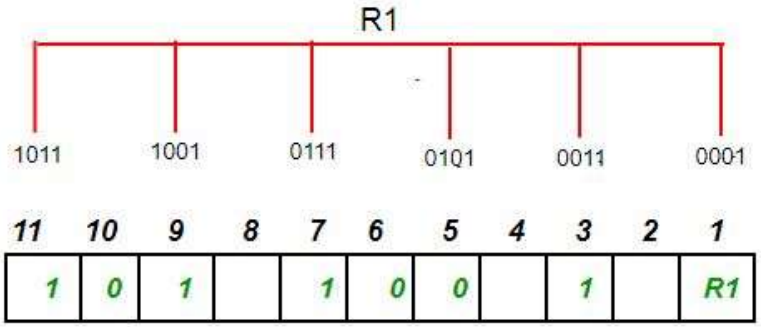
•Suppose the data to be transmitted is 1011001, the bits will be placed as follows:



Determining the Parity bits:

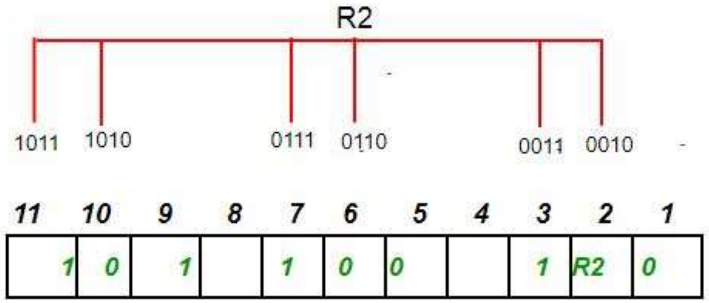
R1 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the least significant position.

R1: bits 1, 3, 5, 7,



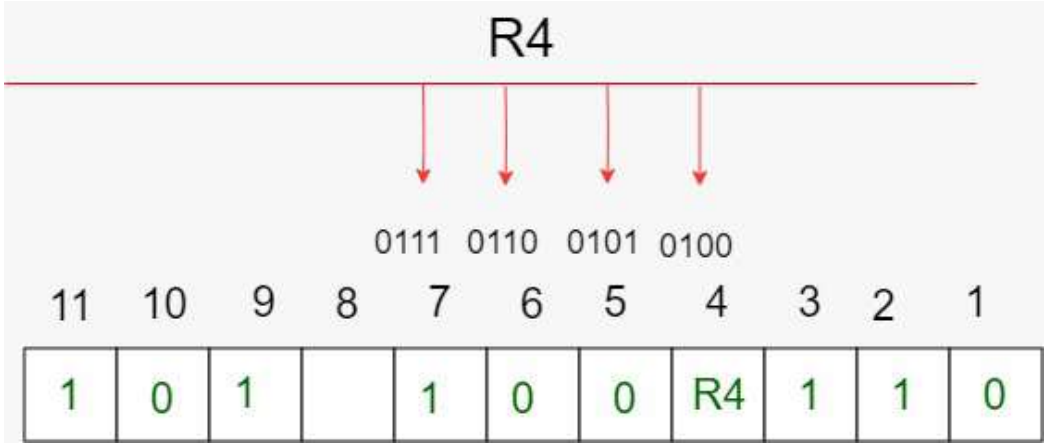
To find the redundant bit R1, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R1 is an even number the value of R1 (parity bit's value) = 0

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



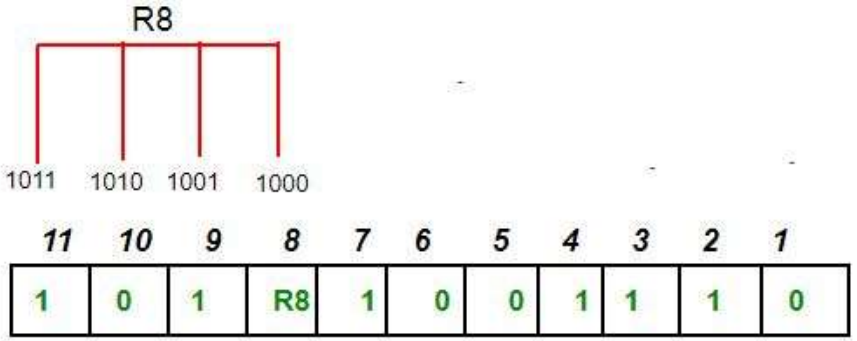
To find the redundant bit R2, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R2 is odd the value of R2(parity bit's value)=1

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



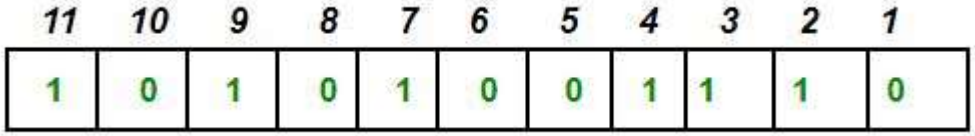
To find the redundant bit R4, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R4 is odd the value of R4(parity bit's value) = 1

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



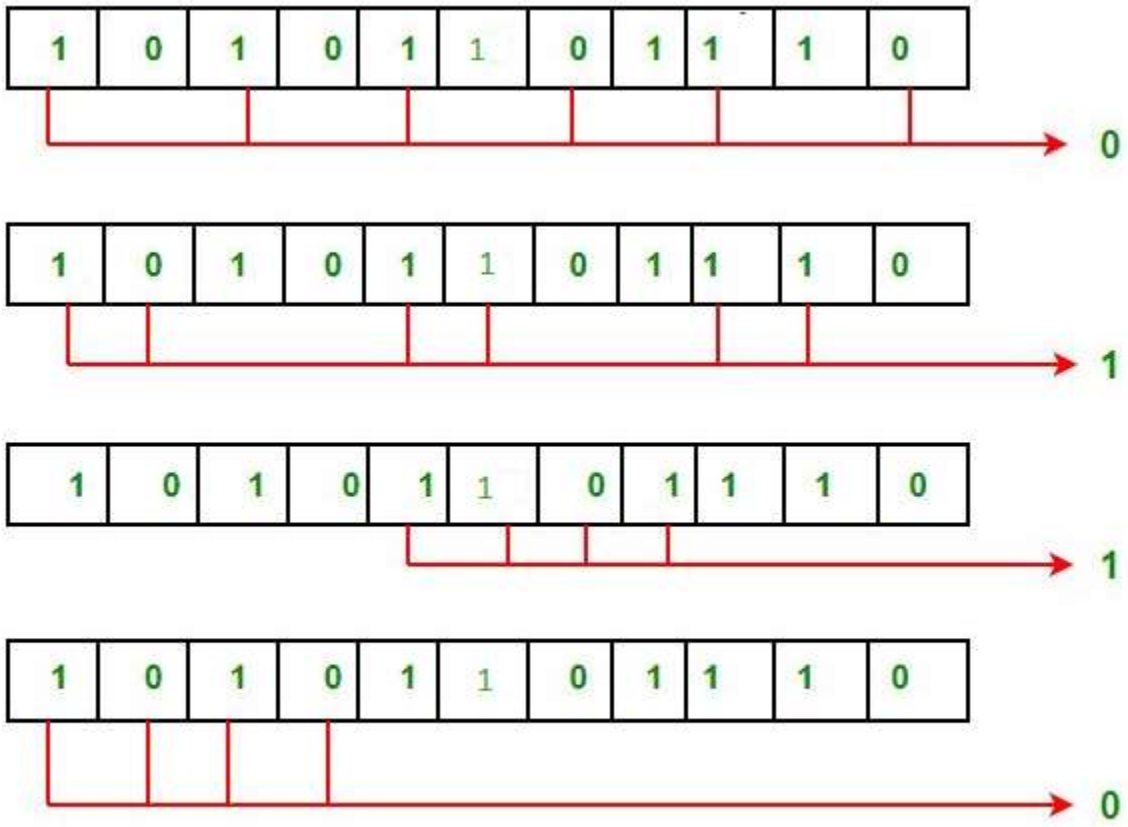
To find the redundant bit R8, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R8 is an even number the value of R8(parity bit's value)=0.

Thus, the data transferred is:



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:



For all the parity bits we will check the number of 1's in their respective bit positions. For R1: bits 1, 3, 5, 7, 9, 11. We can see that the number of 1's in these bit positions are 4 and that's even so we get a 0 for this.

For R2: bits 2,3,6,7,10,11 . We can see that the number of 1's in these bit positions are 5 and that's odd so we get a 1 for this.

For R4: bits 4, 5, 6, 7 . We can see that the number of 1's in these bit positions are 3 and that's odd so we get a 1 for this.

For R8: bit 8,9,10,11 . We can see that the number of 1's in these bit positions are 2 and that's even so we get a 0 for this.

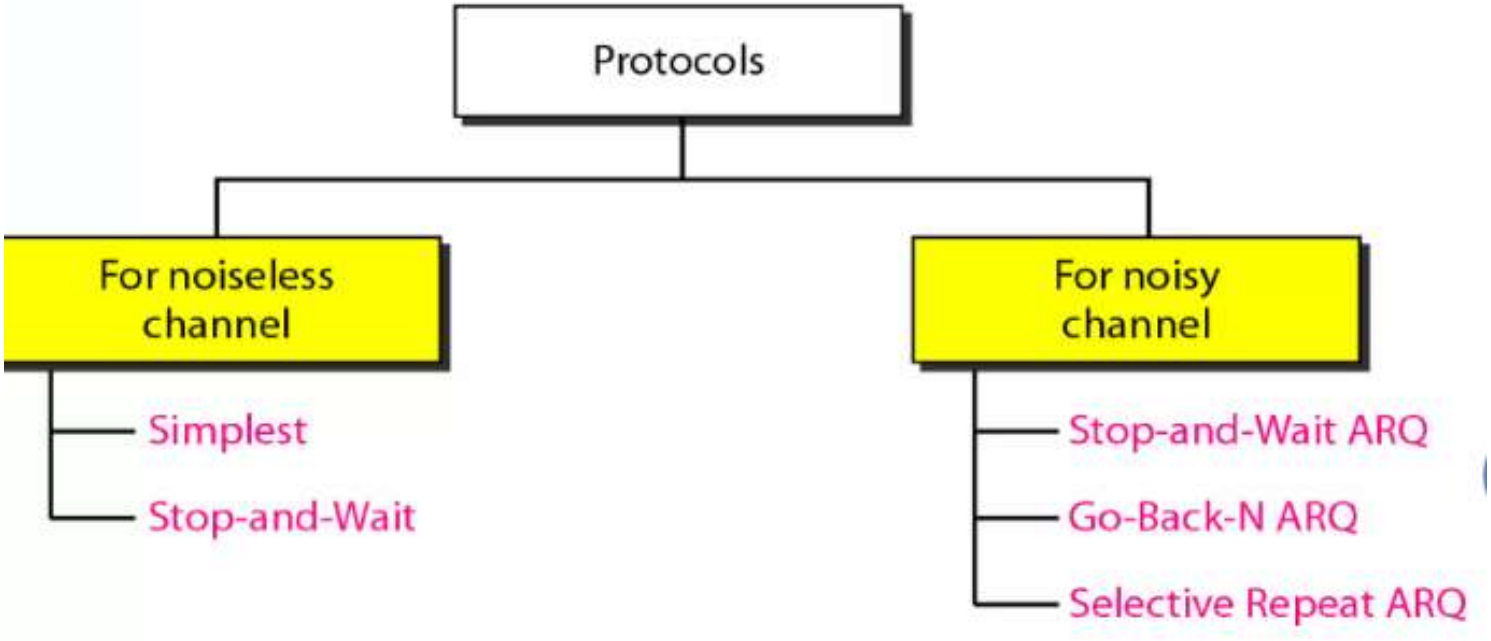
The bits give the binary number 0110 whose decimal representation is 6. Thus, bit 6 contains an error.

To correct the error the 6th bit is changed from 1 to 0.

ELEMENTARY DATA LINK PROTOCOLS

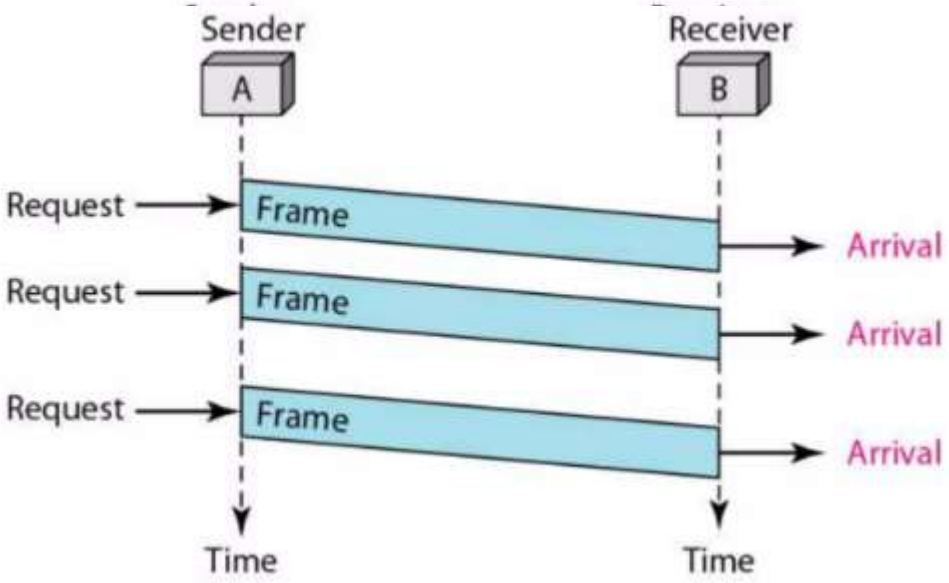
The protocols are normally implemented in software by using one of the common programming languages.

- An Unrestricted Simplex Protocol
- A Simplex Stop-and-Wait Protocol
- A Simplex Protocol for a Noisy Channel



AN UNRESTRICTED SIMPLEX PROTOCOL

- In order to appreciate the step by step development of efficient and complex protocols we will begin with a simple but unrealistic protocol. In this protocol: Data are transmitted in one direction only
- The transmitting (Tx) and receiving (Rx) hosts are always ready
- Processing time can be ignored
- Infinite buffer space is available
- No errors occur; i.e. no damaged frames and no lost frames (perfect channel)



A SIMPLEX STOP-AND-WAIT PROTOCOL

- In this protocol we assume that Data are transmitted in one direction only
- No errors occur (perfect channel)
- The receiver can only process the received information at a finite rate
- These assumptions imply that the transmitter cannot send frames at a rate faster than the receiver can process them.

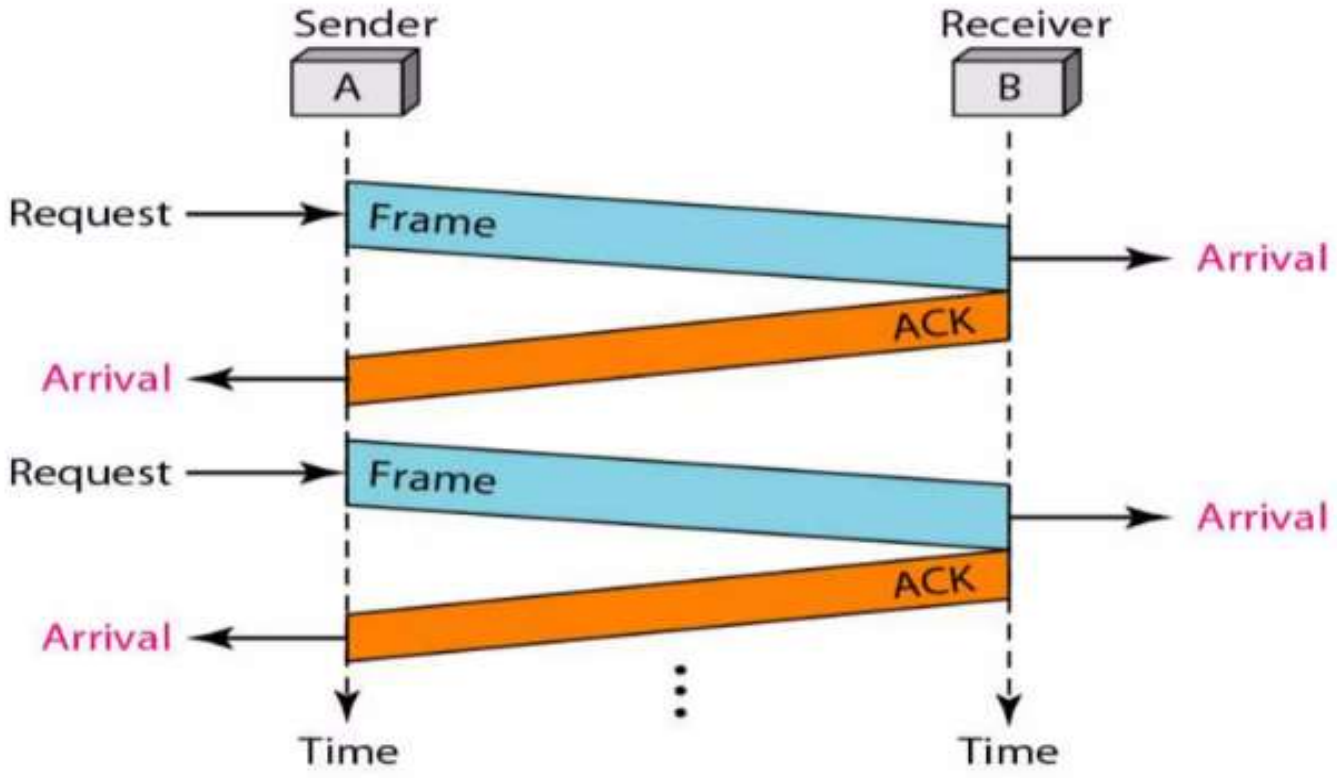
The problem here is how to prevent the sender from flooding the receiver.

- A general solution to this problem is to have the receiver provide some sort of feedback to the sender. The process could be as follows: The receiver send an acknowledge frame back to the sender telling the sender that the last received frame has been processed and passed to the host; permission to send the next frame is granted. The sender, after having sent a frame, must wait for the acknowledge frame from the receiver before sending another frame.

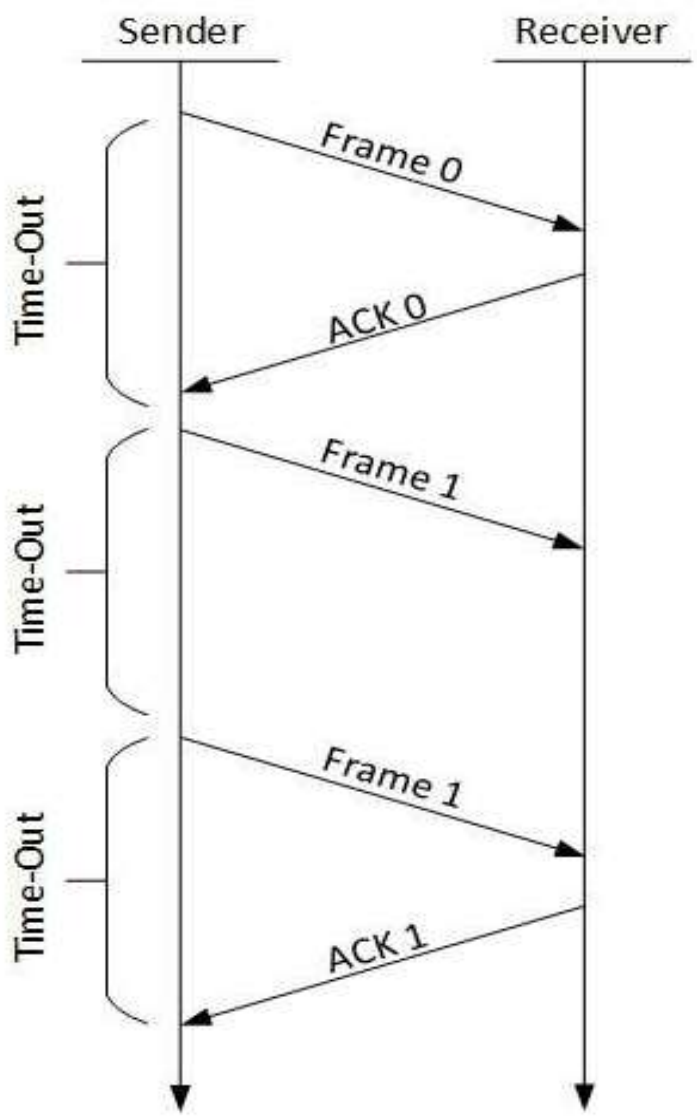
This protocol is known as *stop-and-wait*.

STOP & WAIT PROTOCOL

The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame



- Stop-and-wait ARQ

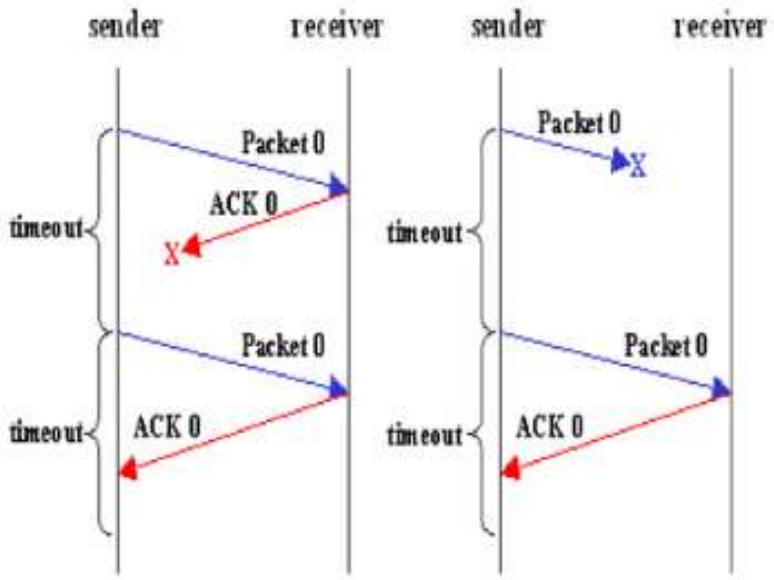


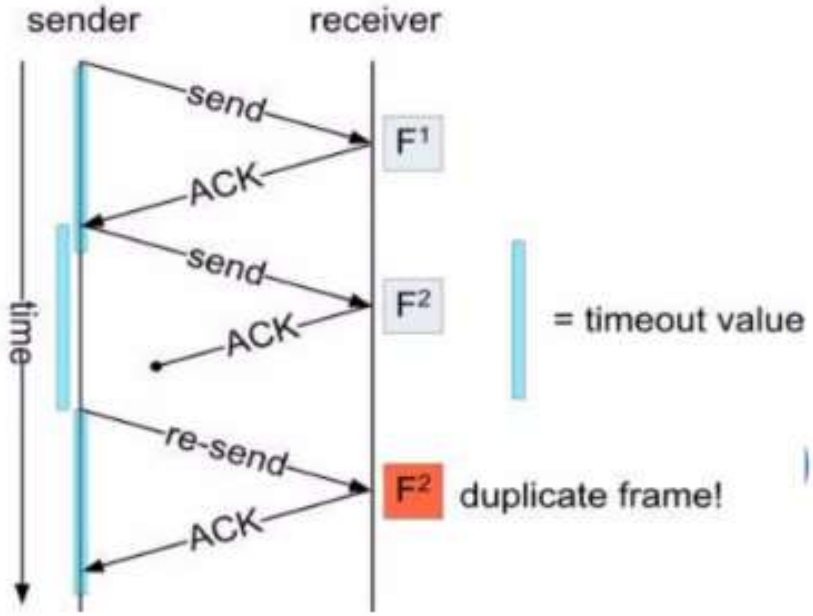
The following transition may occur in Stop-and-Wait ARQ:

- 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.

A SIMPLEX PROTOCOL FOR A NOISY CHANNEL

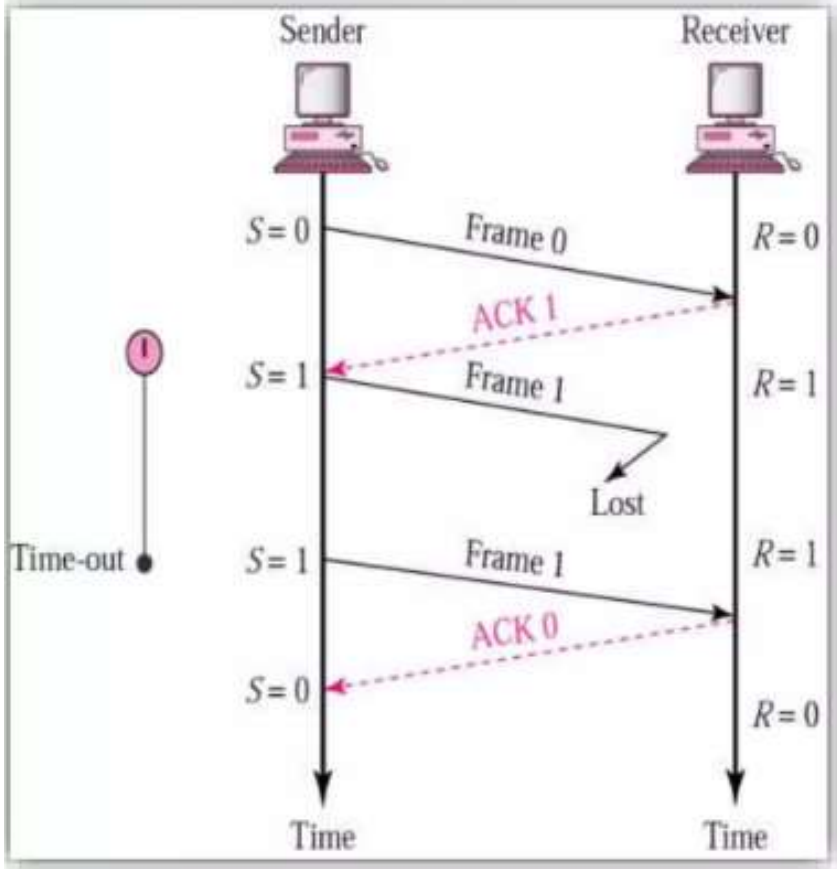
- In this protocol the unreal "error free" assumption in protocol 2 is dropped. Frames may be either damaged or lost completely. We assume that transmission errors in the frame are detected by the hardware checksum. One suggestion is that the sender would send a frame, the receiver would send an ACK frame only if the frame is received correctly. If the frame is in error the receiver simply ignores it; the transmitter would time out and would retransmit it.
- One fatal flaw with the above scheme is that if the ACK frame is lost or damaged, duplicate frames are accepted at the receiver without the receiver knowing it.



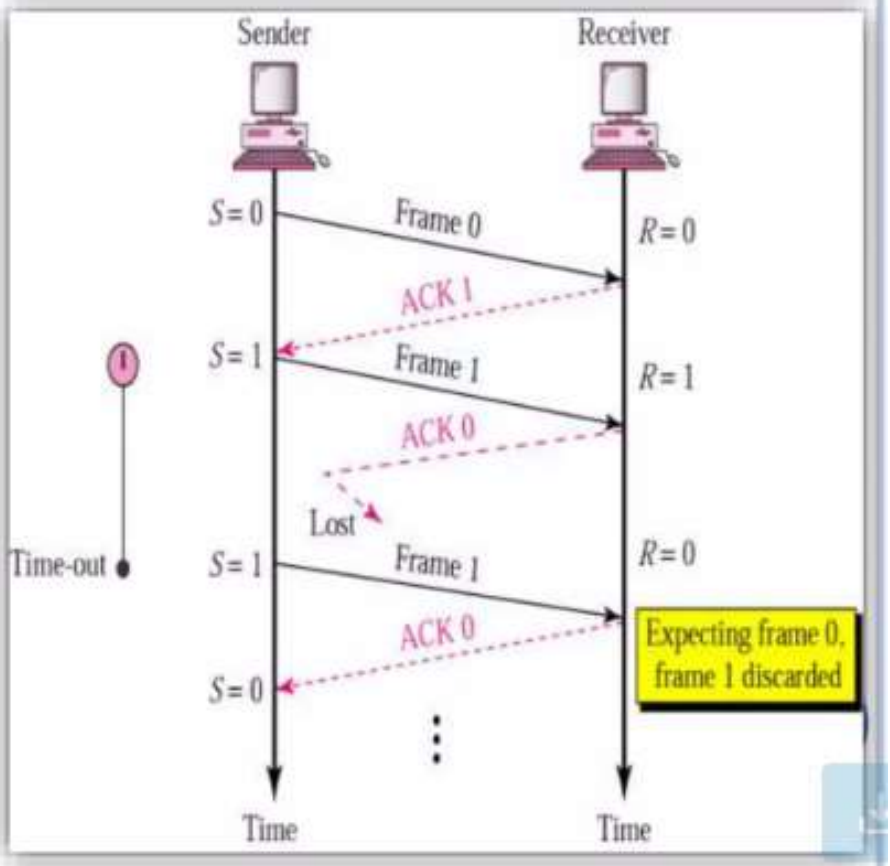


- Imagine a situation where the receiver has just sent an ACK frame back to the sender saying that it correctly received and already passed a frame to its host. However, the ACK frame gets lost completely, the sender times out and retransmits the frame. There is no way for the receiver to tell whether this frame is a retransmitted frame or a new frame, so the receiver accepts this duplicate happily and transfers it to the host. The protocol thus fails in this aspect.

STOP-AND-WAIT, LOST FRAME



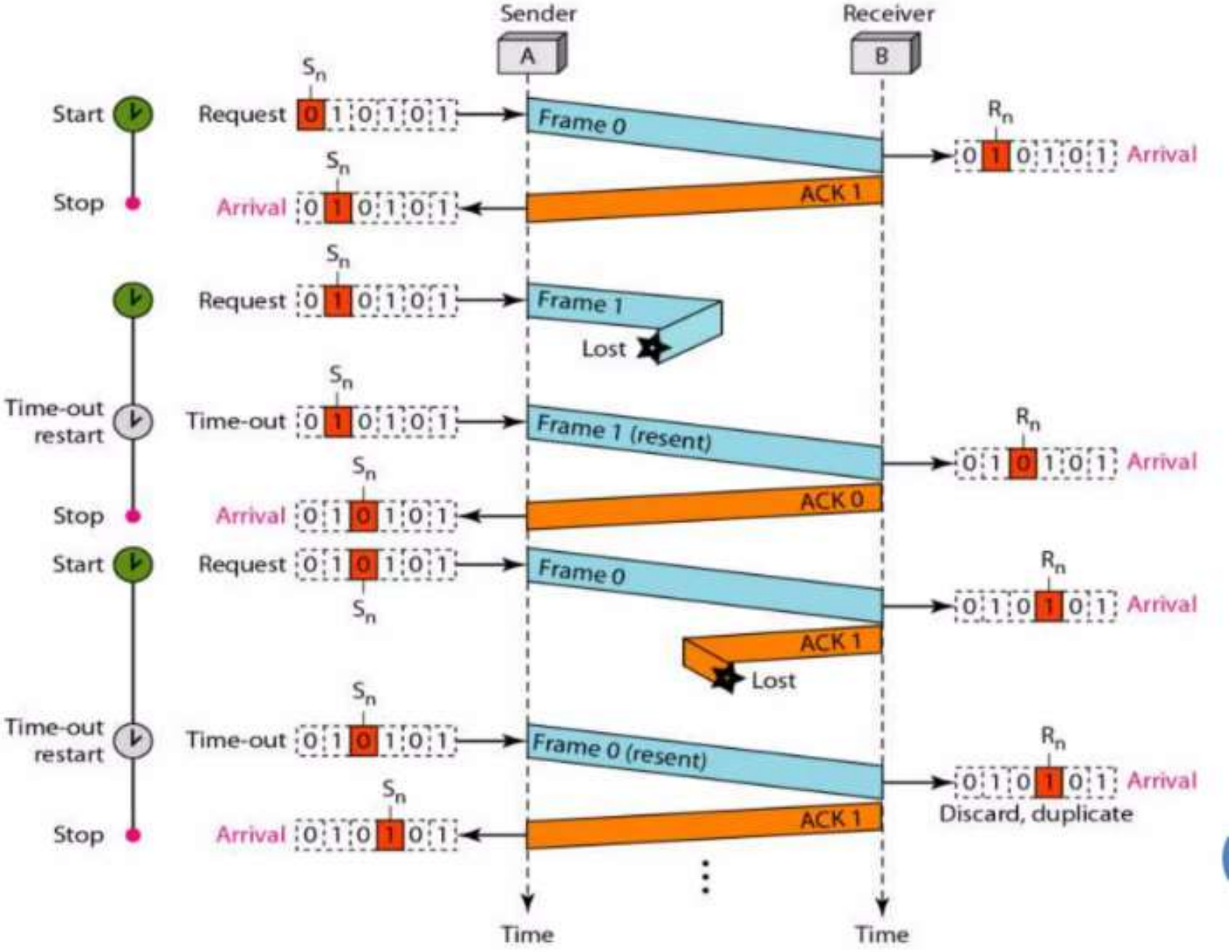
STOP-AND-WAIT, LOST ACK FRAME



- To overcome this problem it is required that the receiver be able to distinguish a frame that it is seeing for the first time from a retransmission. One way to achieve this is to have the sender put a **sequence number** in the header of each frame it sends. The receiver then can check the sequence number of each arriving frame to see if it is a new frame or a duplicate to be discarded.

- The receiver needs to distinguish only 2 possibilities: a new frame or a duplicate; a **1-bit sequence number** is sufficient. At any instant the receiver expects a particular sequence number. Any wrong sequence numbered frame arriving at the receiver is rejected as a duplicate. A correctly numbered frame arriving at the receiver is accepted, passed to the host, and the expected sequence number is incremented by 1 (modulo 2).

FLOW DIAGRAM OF A STOP & WAIT PROTOCOL



- After transmitting a frame and starting the timer, the sender waits for something exciting to happen.
  - Only three possibilities exist: an acknowledgement frame arrives undamaged, a damaged acknowledgement frame staggers in, or the timer expires.
- If a valid acknowledgement comes in, the sender fetches the next packet from its network layer and puts it in the buffer, overwriting the previous packet. It also advances the sequence number. If a damaged frame arrives or no frame at all arrives, neither the buffer nor the sequence number is changed so that a duplicate can be sent.
- When a valid frame arrives at the receiver, its sequence number is checked to see if it is a duplicate. If not, it is accepted, passed to the network layer, and an acknowledgement is generated. Duplicates and damaged frames are not passed to the network layer.

Sliding Window Protocols

Sliding Window Protocol

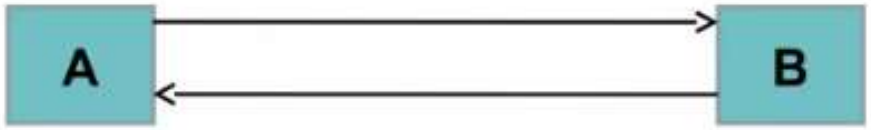
- Sliding window algorithms are a method of flow control for network data transfers.

- Data Link Layer uses a sliding window algorithm, which allows a sender to have more than one unacknowledged packet "in flight" at a time, which improves network throughput.

https://www.youtube.com/watch?v=LnbvhoxHn8M
https://www.youtube.com/watch?v=QD3oCelHJ20
https://www.youtube.com/watch?v=WfIhQ3o2xow

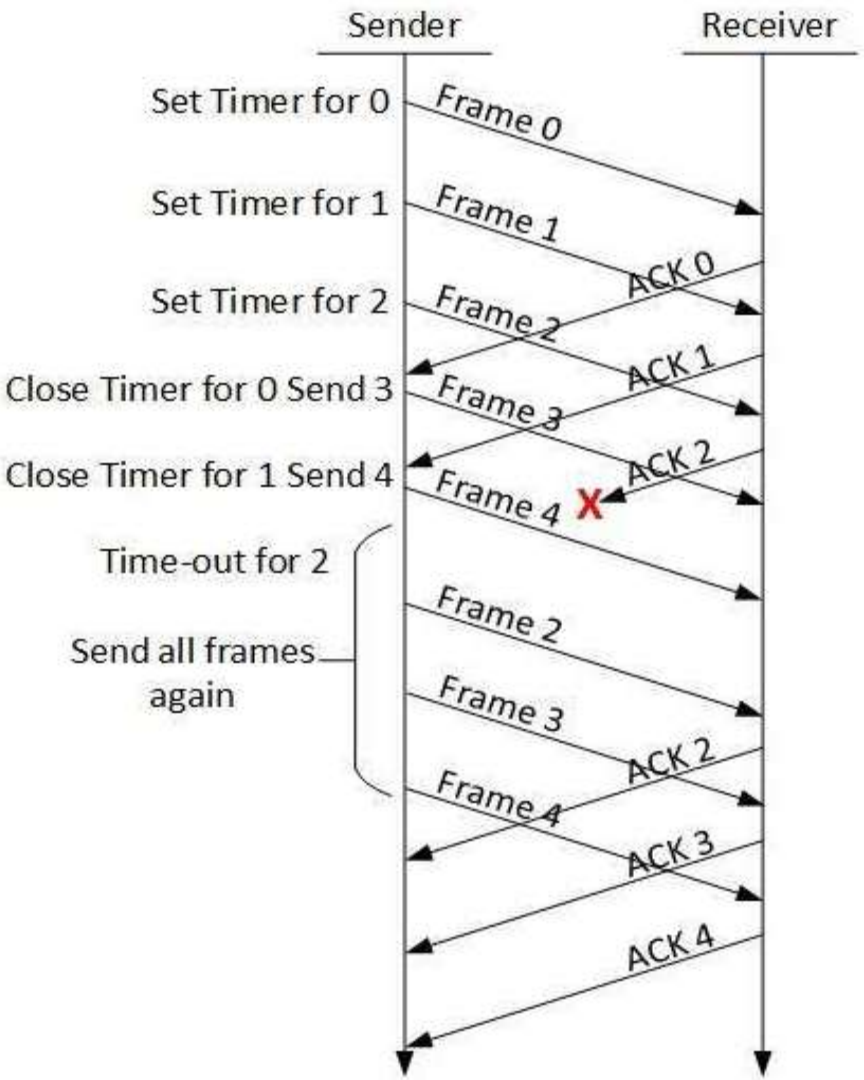
Key concepts of the Sliding Window

- Both the sender and receiver maintain a finite size buffer to hold outgoing and incoming packets from the other side.
- Every packet sent by the **sender**, must be acknowledged by the receiver. The sender maintains a timer for every packet sent, and any packet unacknowledged in a certain time, is **resent**.
- The sender may send a whole window of packets before receiving an acknowledgement for the first packet in the window. This results in higher transfer rates, as the sender may send multiple packets without waiting for each packet's acknowledgement.
- The Receiver advertises a window size that tells the sender how much data it can receive, in order for the sender not to fill up the receivers buffers.
- Efficiency can also be improved by making use of the full-duplex line



Go-Back-N ARQ

Stop and wait ARQ mechanism does not utilize the resources at their best.When the acknowledgement is received, the sender sits idle and does nothing. In Go-Back-N ARQ method, both sender and receiver maintain a window.

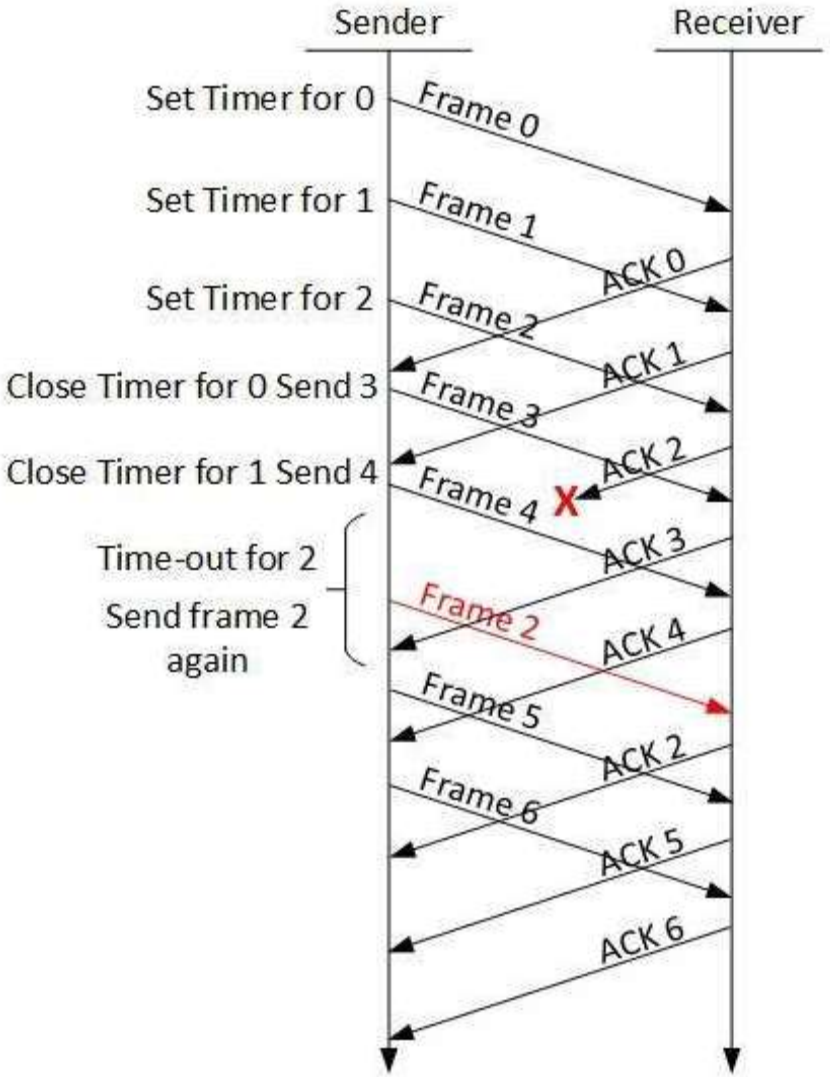


The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame's sequence number.

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 receive any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

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.

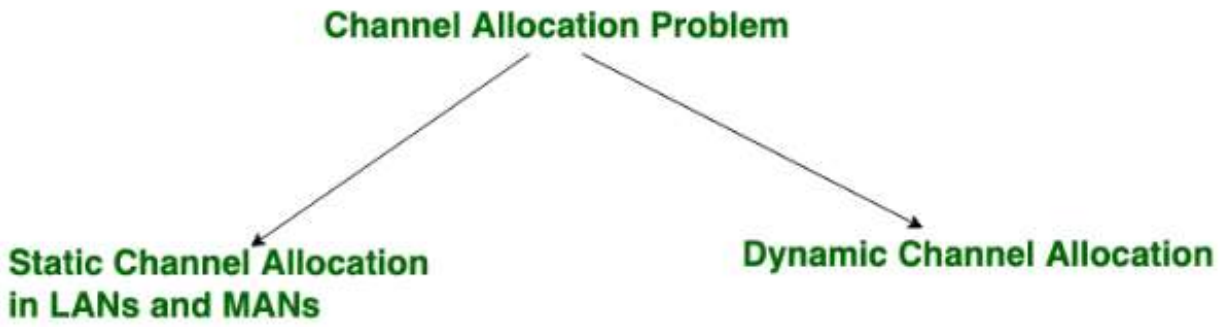
The Medium Access Control Sublayer

Channel Allocation Problem

Channel allocation is a process in which a single channel is divided and allotted to multiple users in order to carry user specific tasks.

There are user's quantity may vary every time the process takes place. If there are N number of users and channel is divided into N equal-sized sub channels, Each user is assigned one portion. If the number of users are small and don't vary at times, then Frequency Division Multiplexing can be used as it is a simple and efficient channel bandwidth allocating technique.

Channel allocation problem can be solved by two schemes: Static Channel Allocation in LANs and MANs, and Dynamic Channel Allocation.



Static Channel Allocation:

In this scheme a **Frequency Division Multiplexing** (FDM) is used for allocating a single channel among competing users.

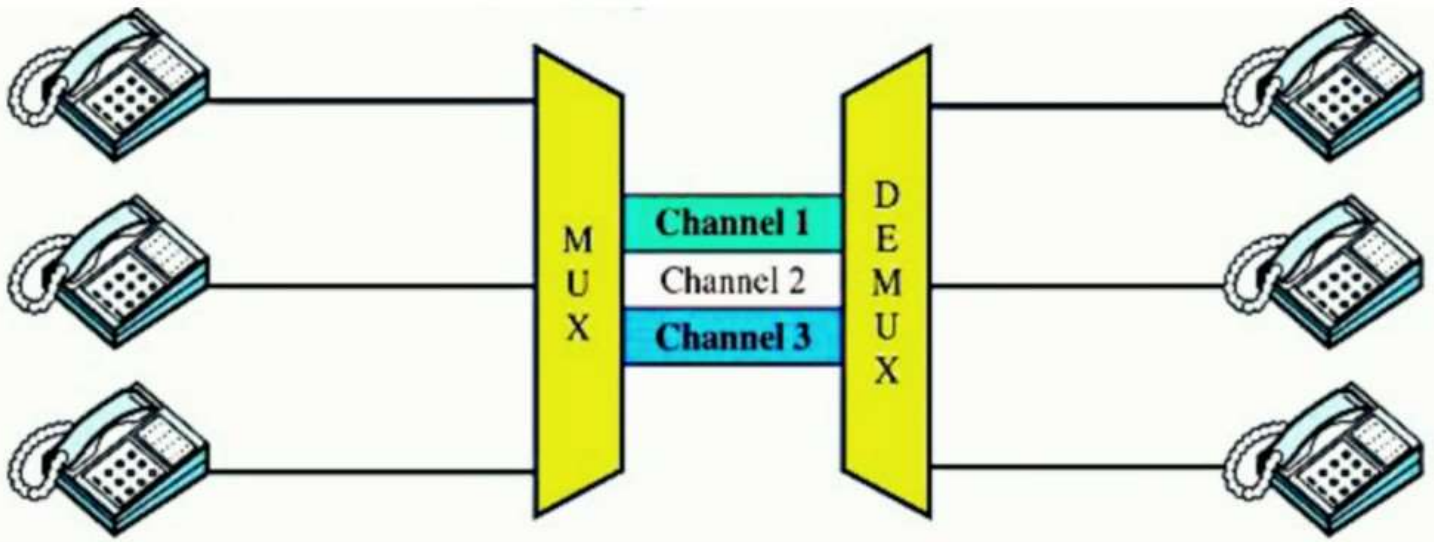
Example

if we have N users, the bandwidth will be divided into N equal-size portions.

++ FDM is a **simple and efficient** allocation mechanism.

- - **Waste of resources** when the traffic is bursty, or the channel is lightly loaded.

Frequency Division Multiplexing



2. Dynamic Channel Allocation:

Possible assumptions include:

1. **Station Model:**
   Assumes that each of N stations independently produce frames. The probability of producing a packet in the interval IDt where I is the constant arrival rate of new frames.

2. **Single Channel Assumption:**
   In this allocation all stations are equivalent and can send and receive on that channel.

3. **Collision Assumption:**
   If two frames overlap in time-wise, then that's collision. Any collision is an error, and both frames must re transmitted. Collisions are only possible error.

4. **Time** can be divided into Slotted or Continuous.

5. **Stations** can sense a channel is busy before they try it.

Protocol Assumption

- N independent stations.
- A station is blocked until its generated frame is transmitted.
- probability of a frame being generated in a period of length Dt is IDt where I is the arrival rate of frames.
- Only a single Channel available.
- Time can be either: Continuous or slotted.
- **Carrier Sense:** A station can sense if a channel is already busy before transmission.
- **No Carrier Sense:** Time out used to sense loss data.

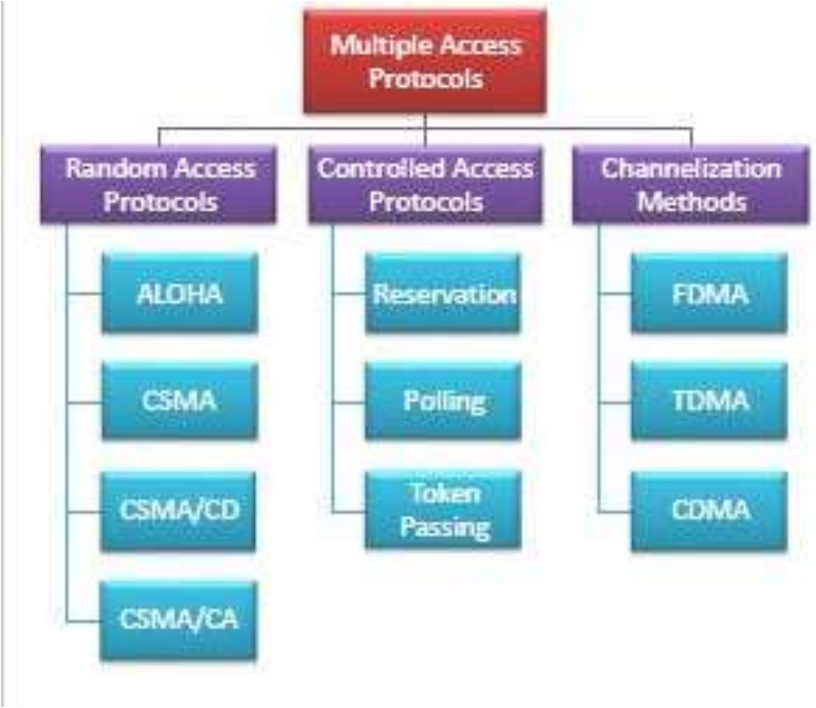
Multiple Access Protocols

Multiple access protocols are a set of protocols operating in the Medium Access Control sublayer (MAC sublayer) of the Open Systems Interconnection (OSI) model. These protocols allow a number of nodes or users to access a shared network channel. Several data streams originating from several nodes are transferred through the multi-point transmission channel.

The objectives of multiple access protocols are optimization of transmission time, minimization of collisions and avoidance of crosstalks.

Categories of Multiple Access Protocols

Multiple access protocols can be broadly classified into three categories - random access protocols, controlled access protocols and channelization protocols.



Random Access Protocols

Random access protocols assign uniform priority to all connected nodes. Any node can send data if the transmission channel is idle. No fixed time or fixed sequence is given for data transmission.

The four random access protocols are−

- ALOHA
- Carrier sense multiple access (CMSA)
- Carrier sense multiple access with collision detection (CMSA/CD)
- Carrier sense multiple access with collision avoidance (CMSA/CA)

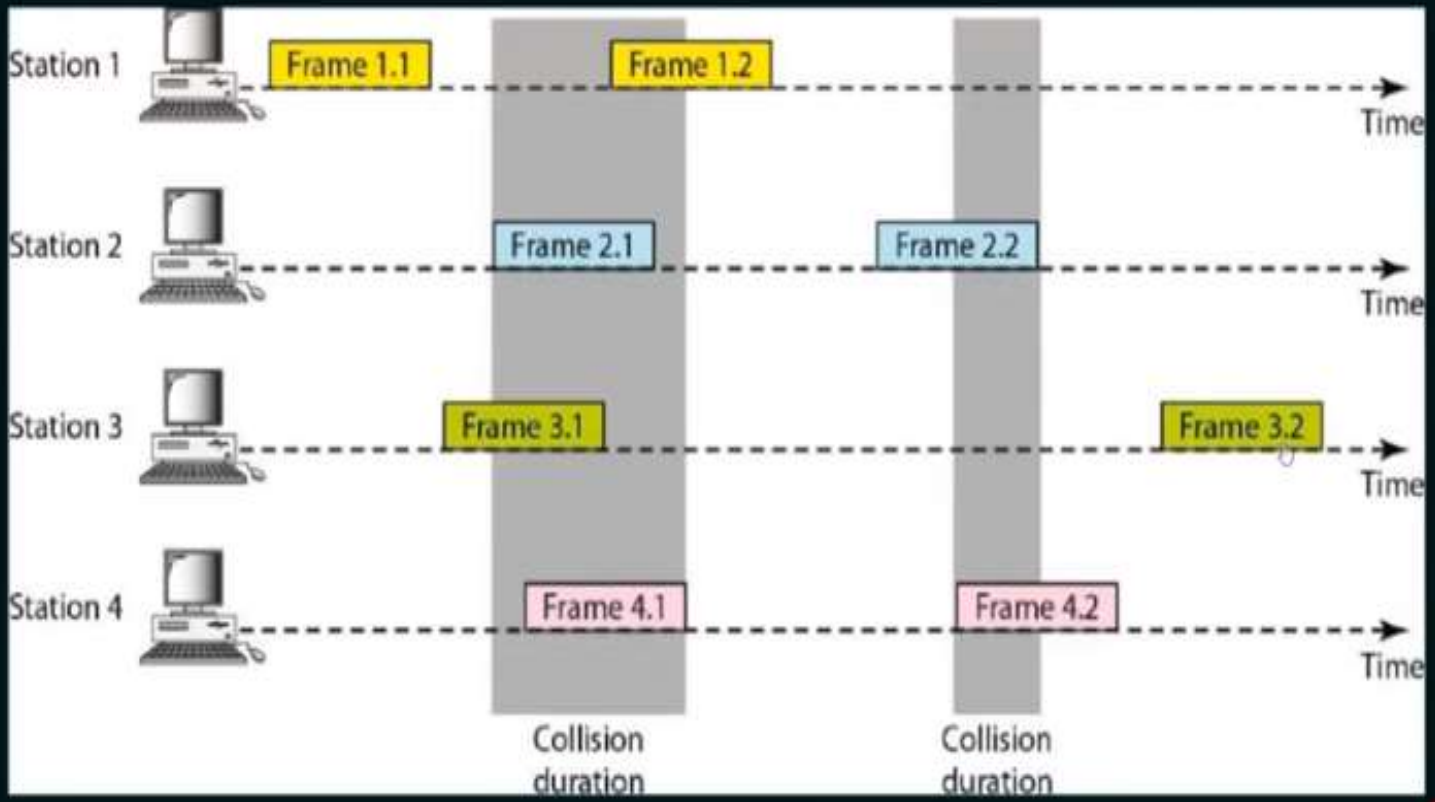
ALOHA

- ★ Aloha is a random access protocol.
- ★ It was actually designed for WLAN but it is also applicable for shared medium.
- ★ In this, multiple stations can transmit data at the same time and can hence lead to collision and data being garbled.

Types:

- ★ Pure Aloha
- ★ Slotted Aloha

Pure ALOHA



Pure ALOHA

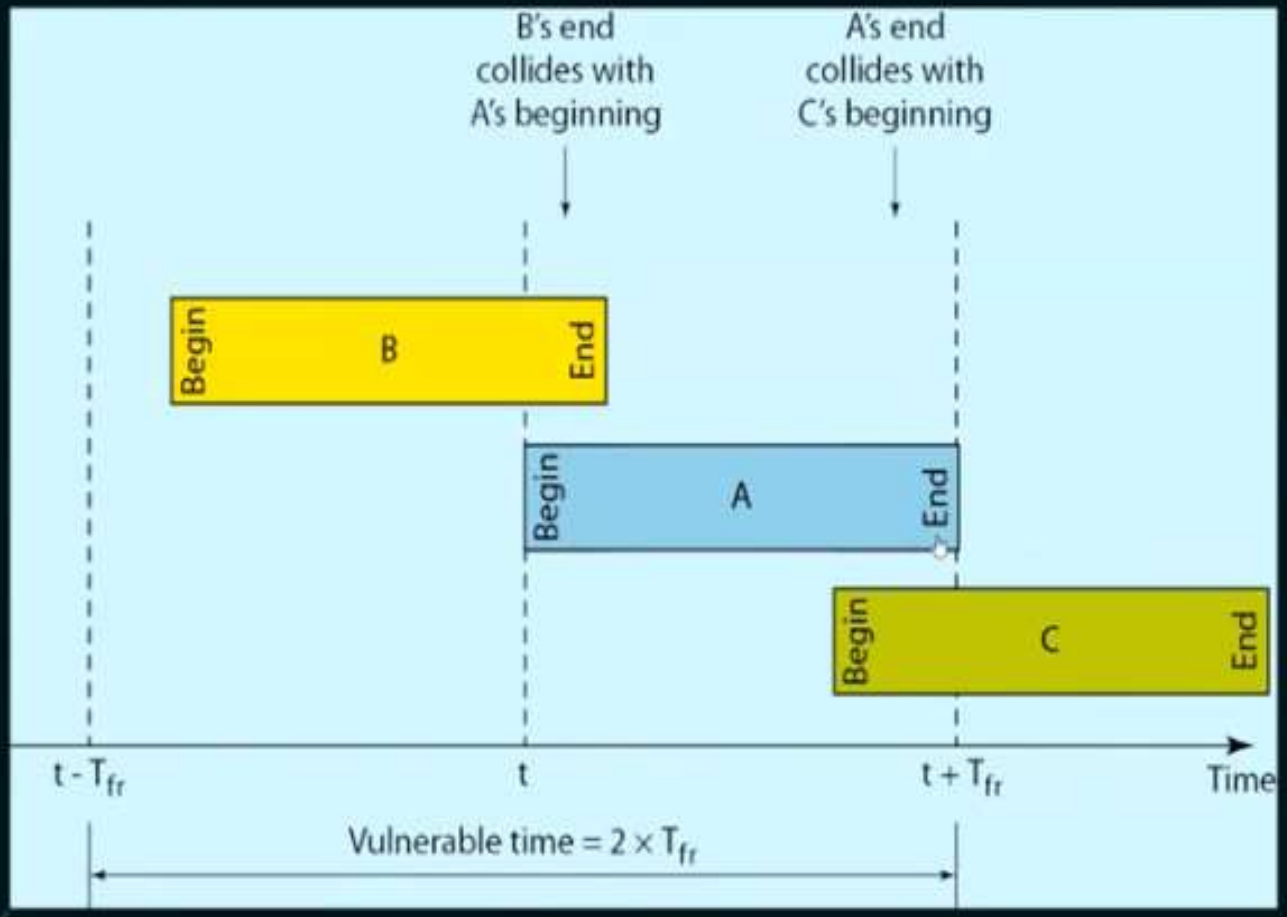
- ★ Pure ALOHA allows stations to transmit whenever they have data to be sent.
- ★ When a station sends data it waits for an acknowledgement.
- ★ If the acknowledgement doesn't come within the allotted time then the station waits for a random amount of time called back-off time (Tb) and re-sends the data.
- ★ Since different stations wait for different amount of time, the probability of further collision decreases.
- ★ The throughput of pure aloha is maximized when frames are of uniform length.

PURE ALOHA

- ★ Whenever two frames try to occupy the channel at the same time, there will be a collision and both will be garbled.
- ★ If the first bit of a new frame overlaps with just the last bit of a frame almost finished, both frames will be totally destroyed and both will have to be retransmitted later.

Vulnerable Time = $2*T_{fr}$

Pure aloha



Example 12.2

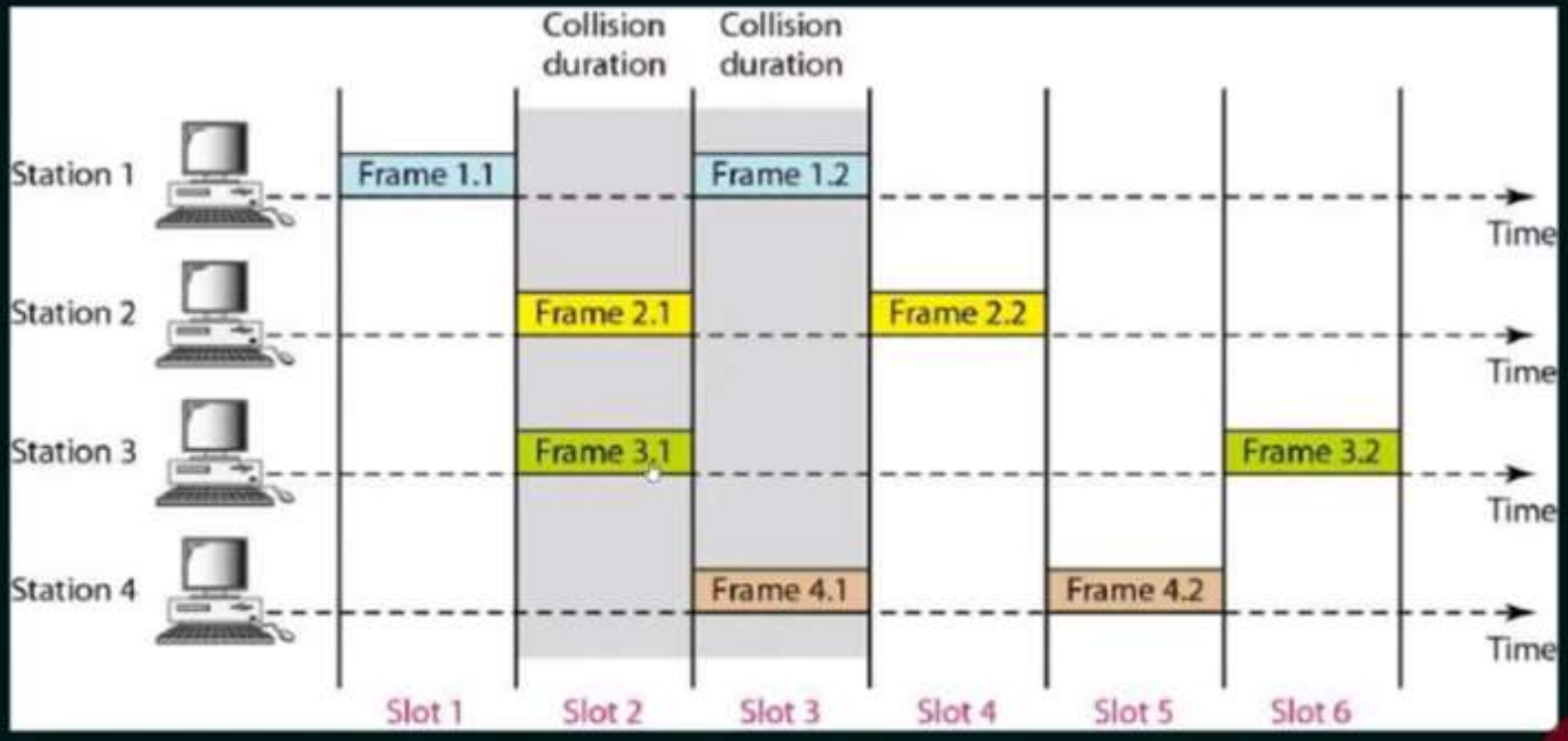
A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

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

https://www.youtube.com/watch?v=aqWTNk90zRA

Slotted ALOHA

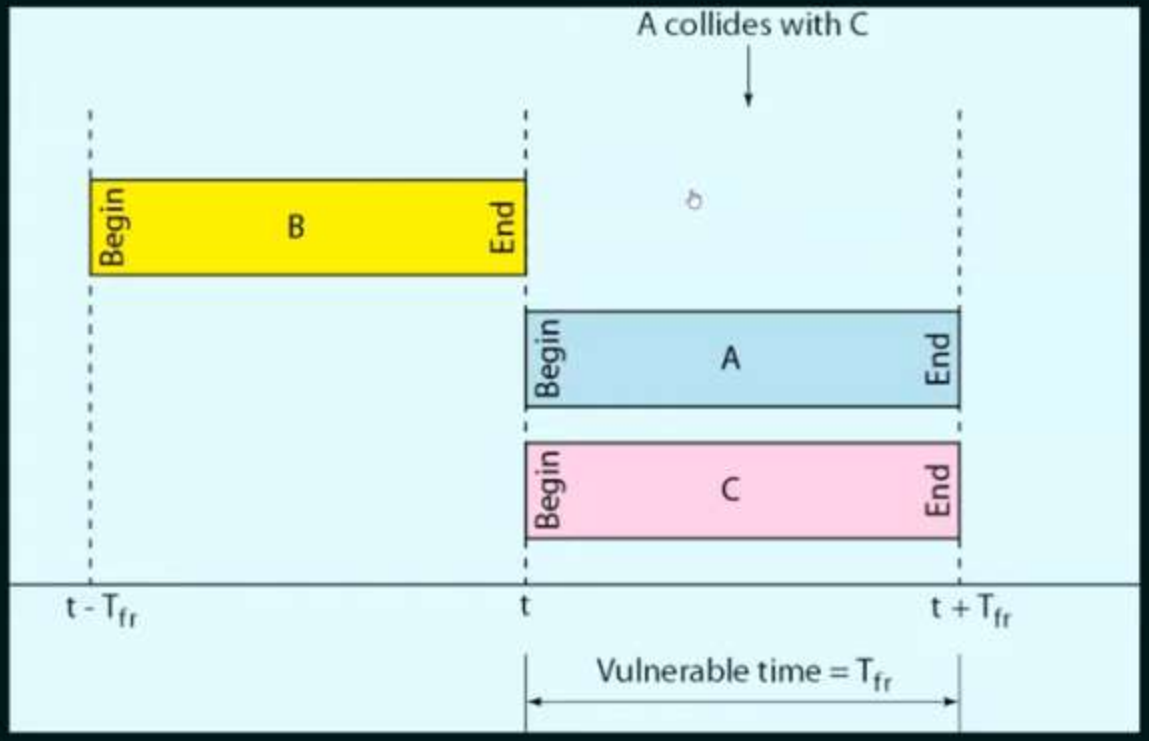


Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to T_{fr}. Figure 12.7 shows the situation.

Slotted aloha

- ★ It was developed just to improve the efficiency of pure aloha as the chances for collision in pure aloha are high.
- ★ The time of the shared channel is divided into discrete time intervals called slots.
- ★ 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.

SLOTTED ALOHA



Pure aloha vs Slotted aloha

Pure Aloha	Slotted Aloha
Any station can transmit the data at any time.	Any station can transmit the data at the beginning of any time slot.
The time is continuous and not globally synchronized.	The time is discrete and globally synchronized.
Vulnerable time in which collision may occur $= 2 \times T_{Fr}$	Vulnerable time in which collision may occur $= T_{Fr}$
Probability of successful transmission of data packet= $G \times e^{-2G}$	Probability of successful transmission of data packet= $G \times e^{-G}$
Maximum efficiency = 18.4% (Occurs at G = 1/2)	Maximum efficiency = 36.8% (Occurs at G = 1)
Main advantage: Simplicity in implementation.	Main advantage: It reduces the number of collisions to half and doubles the efficiency of pure aloha.

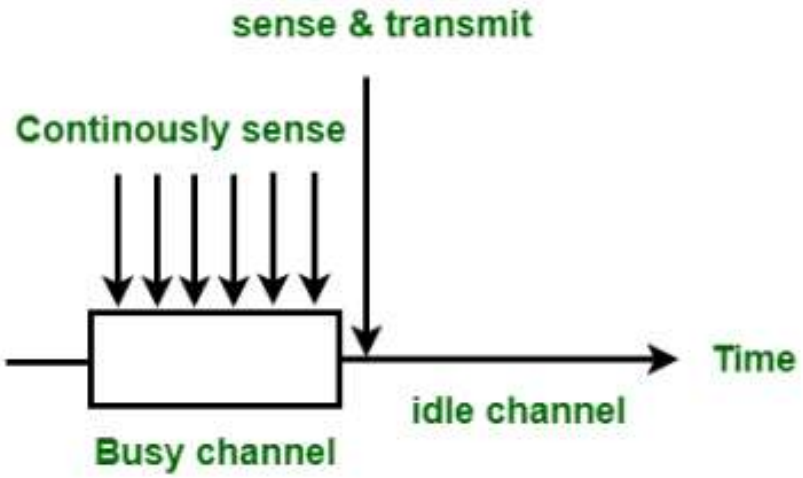
CSMA PROTOCOL

- ★ Carrier Sense Protocol.
- ★ To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed.
- ★ Principle of CSMA: "sense before transmit" or "listen before talk."
- ★ Carrier busy = Transmission is taking place.
- ★ Carrier idle = No transmission currently taking place.
- ★ The possibility of collision still exists because of propagation delay; a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

- Before sending the data, the station 1st listens to the channel to see if anyone else is transmitting the data at that moment.

- If the channel is idle,the station transmits a frame.

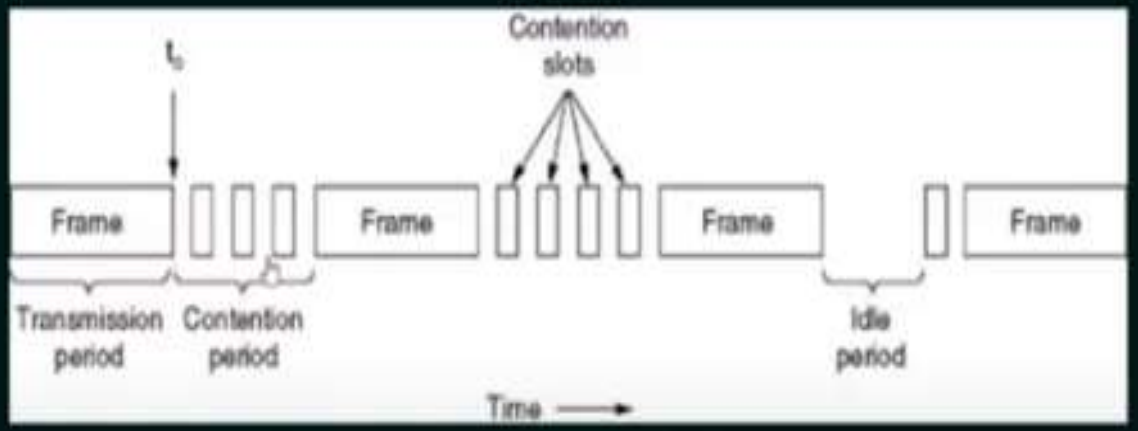
- If busy,then it senses the transmission medium continuously until it becomes idle.



CSMA/CD

- ★ If two stations sense the channel to be idle and begin transmitting simultaneously, they will both detect the collision almost immediately.
- ★ Rather than finish transmitting their frames, which are irretrievably garbled anyway, they should suddenly stop transmitting as soon as the collision is detected.
- ★ Quickly terminating damaged frames saves time and bandwidth.
- ★ This protocol, known as CSMA/CD (CSMA with Collision Detection) is widely used on LANs in the MAC sublayer.
- ★ Access method used by Ethernet: CSMA/CD.

CSMA/CD


Contention slots
t_0
Frame
Frame
Frame
Frame
Transmission period
Contention period
Idle period
Time

- Contention Period:a minimum time where any host or stations will check the collision is happen or not.

CSMA/CA

- ★ Carrier-sense multiple access with collision avoidance (CSMA/CA) is a network multiple access method in which carrier sensing is used, but nodes attempt to avoid collisions by beginning transmission only after the channel is sensed to be "idle".
- ★ It is particularly important for wireless networks, where the collision detection of the alternative CSMA/CD is not possible due to wireless transmitters desensing their receivers during packet transmission.
- ★ The Access method used by IEEE 802.11 Wi-Fi is CSMA/CA.

- Collisions are avoided by using three methods.
  a. Inter-frame space b. Contention window c. Acknowledgments
- Fig. 5.6.12 shows the all three method of CSMA/CA

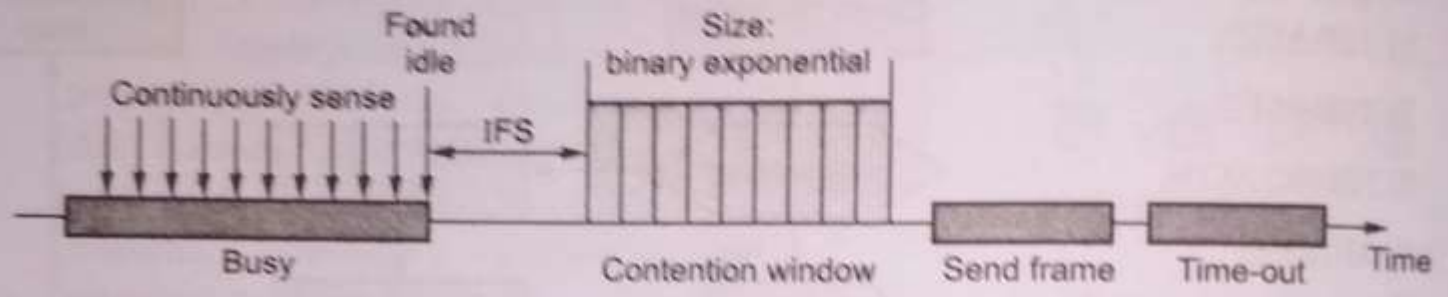


Fig. 5.6.12 CSMA/CA methods

Inter-frame space

- Collisions are avoided by deferring transmission even if the channel is found idle.
- When an idle channel is found, the station does not send immediately. It waits for a period of time called the Inter-Frame Space (IFS).
- In CSMA/CA, the IFS can also be used to define the priority of a station of a frame. A station that is assigned shorter IFS has a higher priority.

Contention window

- Contention windows are an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time.
- Station set one slot for the first time and then double each time the station cannot detect an idle channel after the IFS time.
- In this method, the station needs to sense the channel after each time slot
- If the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle.
- This method gives the priority to the station with the longest waiting time.

Acknowledgments

- The data may be corrupted during the transmission. The positive acknowledgment and the time out can help guarantee that the receiver has received the frame.

Controlled Access Protocols

Controlled access protocols allow only one node to send data at a given time. Before initiating transmission, a node seeks information from other nodes to determine which station has the right to send. This avoids collision of messages on the shared channel.

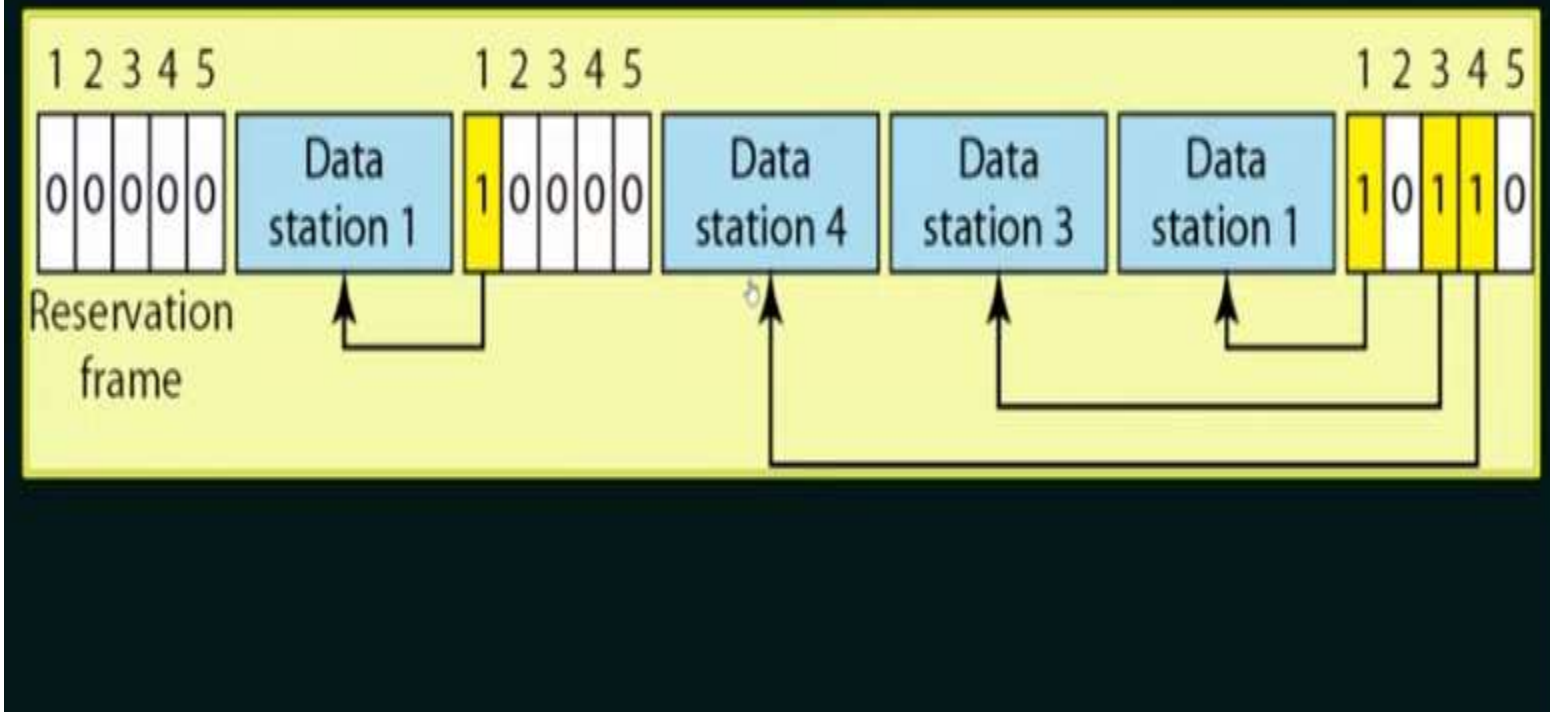
The station can be assigned the right to send by the following three methods−

- Reservation
- Polling
- Token Passing

Reservation

- ★ A station need to make a reservation before sending data.
- ★ In each interval, a reservation frame precedes the data frames sent in that interval.
- ★ If there are N stations in the system, there are exactly N reservation minislots in the reservation frame.
- ★ Each minislot belongs to a station.
- ★ When a station needs to send a data frame, it makes a reservation in its own minislot.
- ★ The stations that have made reservations can send their data frames after the reservation frame.

RESERVATION



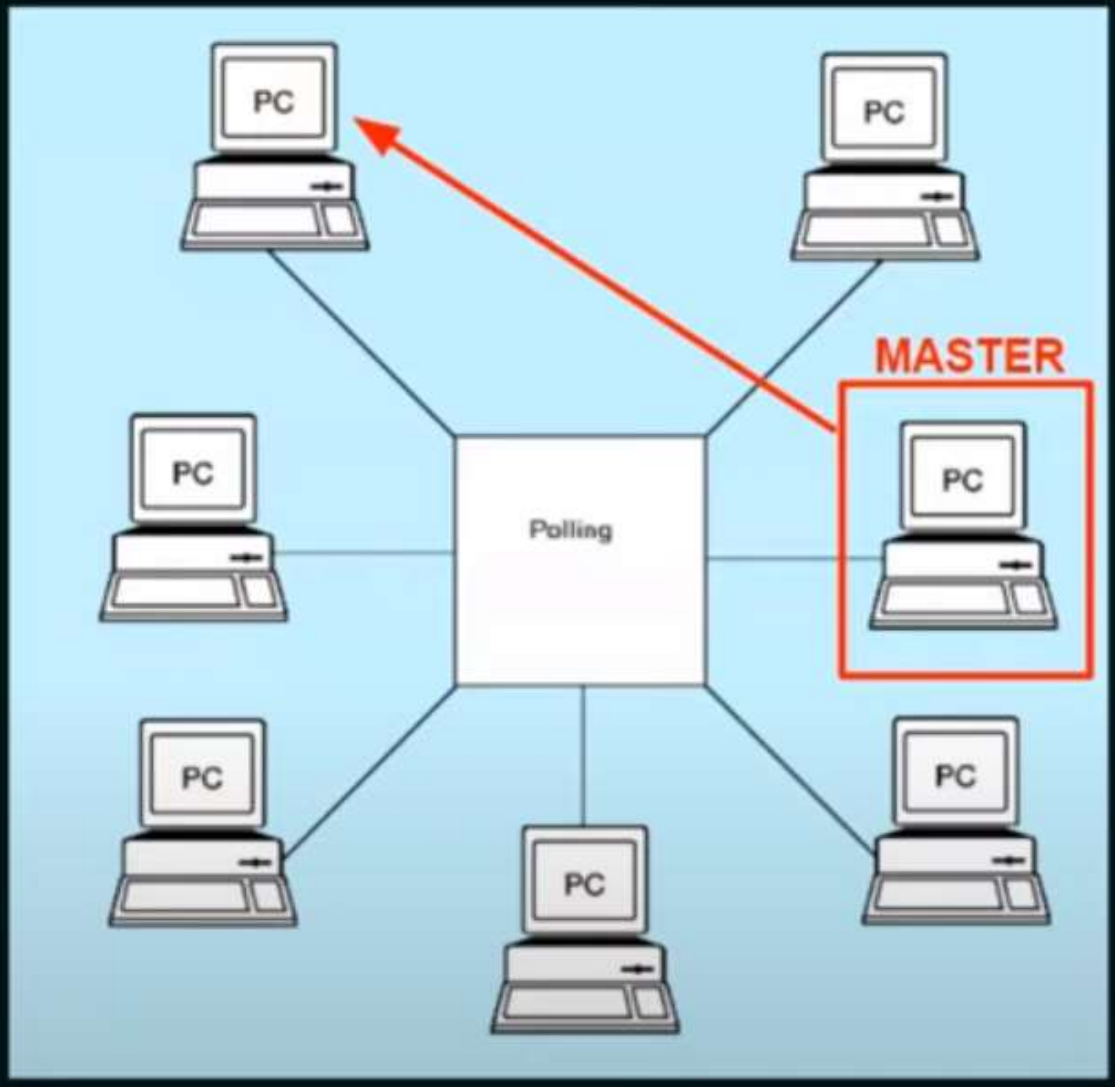
Polling

- ★ The polling protocol requires one of the nodes to be designated as a Master node (Primary station).
- ★ The master node polls each of the nodes in a round-robin fashion.
- ★ In particular, the master node first sends a message to node 1, saying that it (node 1) can transmit up to some maximum number of frames.
- ★ After node 1 transmits some frames, the master node tells node 2 it (node 2) can transmit up to the maximum number of frames.
- ★ The master node can determine when a node has finished sending its frames by observing the lack of a signal on the channel.

POLLING

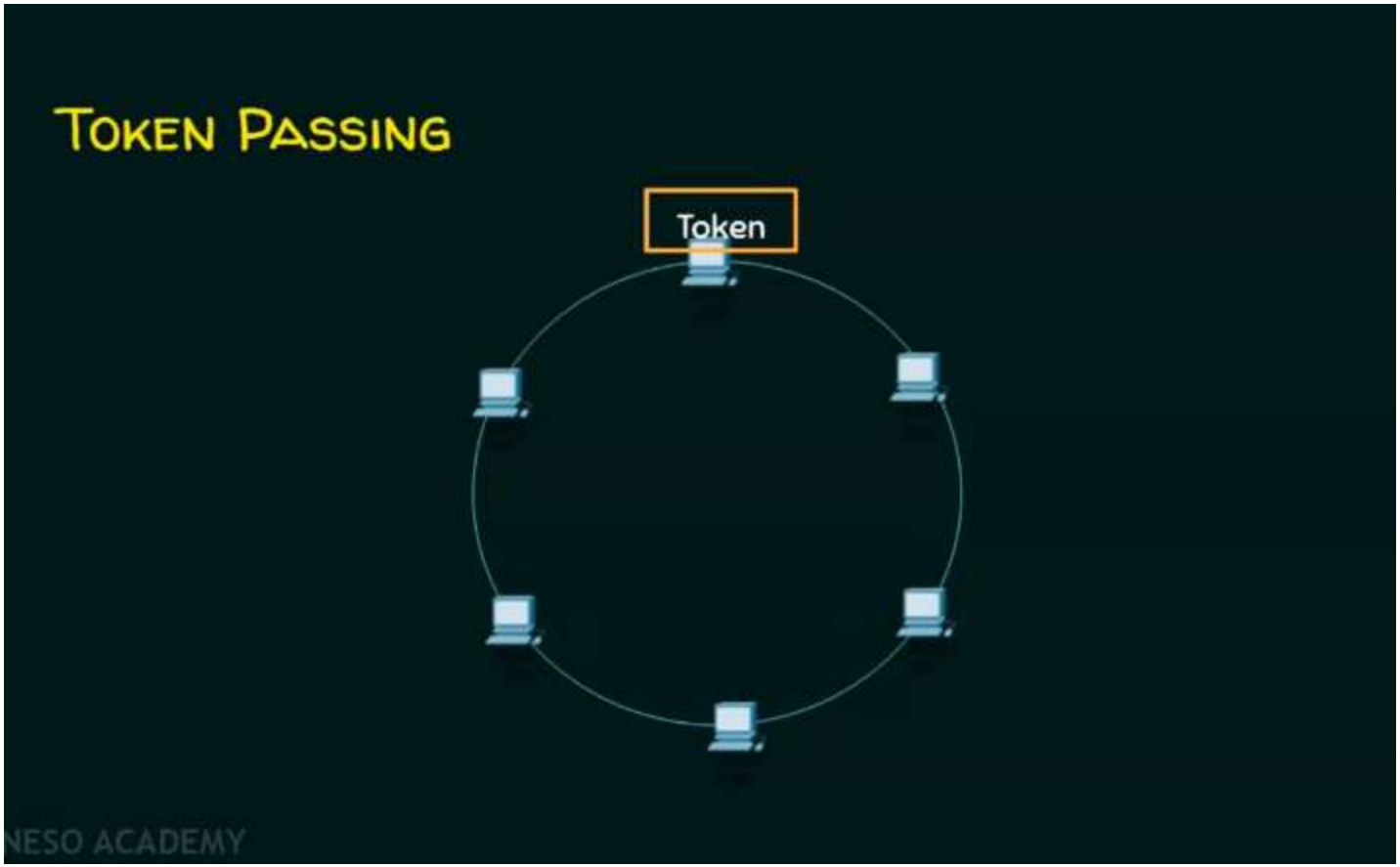
- ★ The procedure continues in this manner, with the master node polling each of the nodes in a cyclic manner.
- ★ The polling protocol eliminates the collision.
- ★ This allows polling to achieve a much higher efficiency.
- ★ The first drawback is that the protocol introduces a polling delay–the amount of time required to notify a node that it can transmit.
- ★ The second drawback, which is potentially more serious, is that if the master node fails, the entire channel becomes inoperative.

POLLING



Token passing

- ★ If a node does have frames to transmit when it receives the token, it sends up to a maximum number of frames and then forwards the token to the next node.
- ★ Token passing is decentralized and highly efficient. But it has problems as well.
- ★ For example, the failure of one node can crash the entire channel. Or if a node accidentally neglects to release the token, then some recovery procedure must be invoked to get the token back in circulation.


Token Passing
Token
NESO ACADEMY

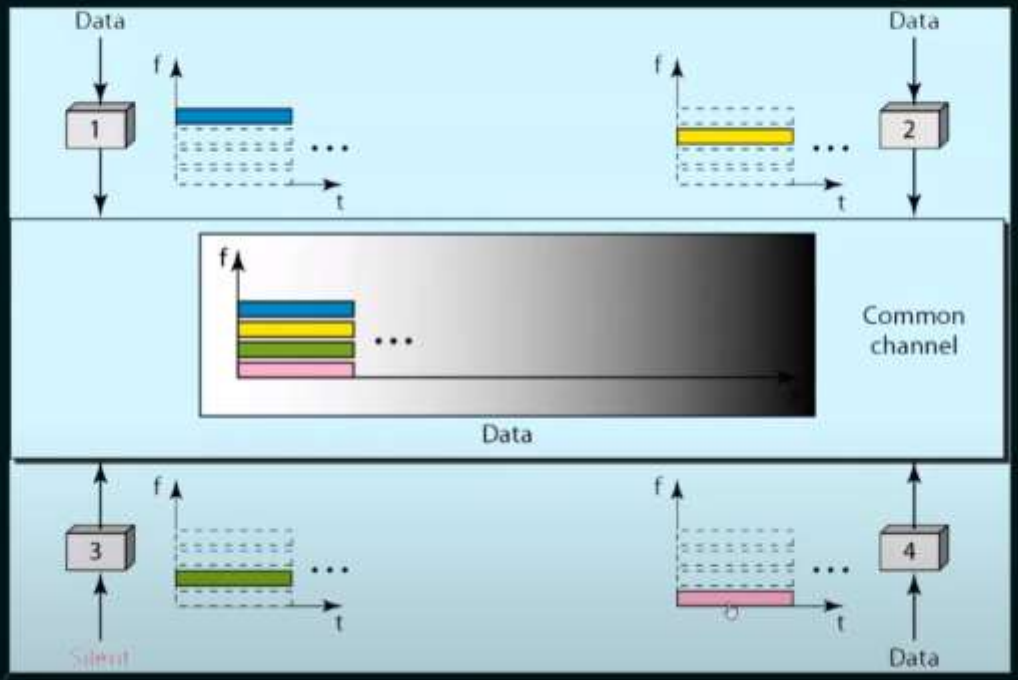
Channelization

Channelization are a set of methods by which the available bandwidth is divided among the different nodes for simultaneous data transfer.

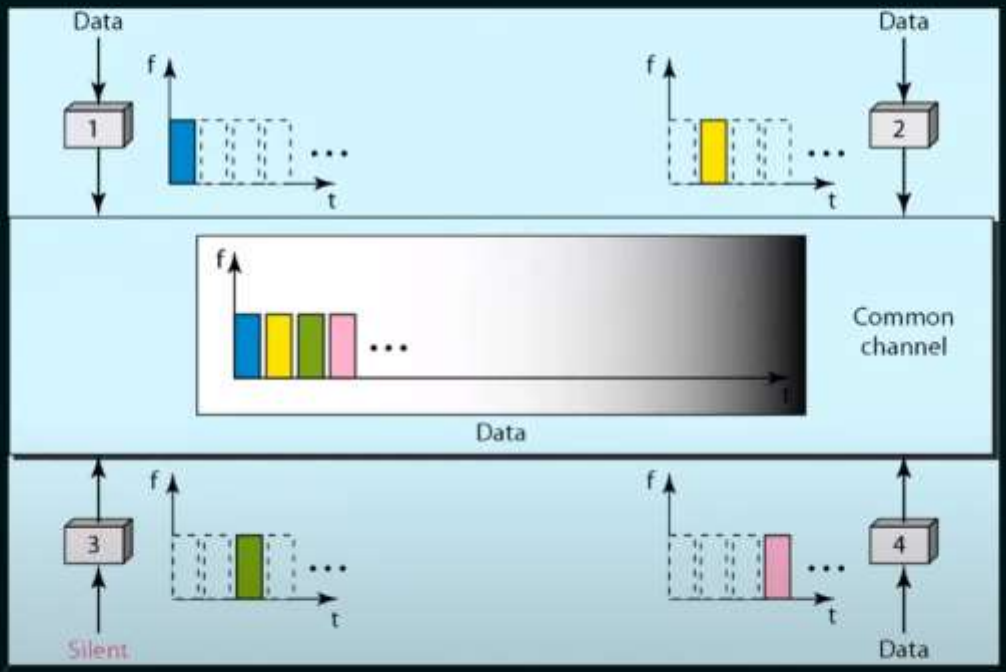
The three channelization methods are−

- Frequency division multiple access (FDMA)
- Time division multiple access (TDMA)
- Code division multiple access (CDMA)

FDMA

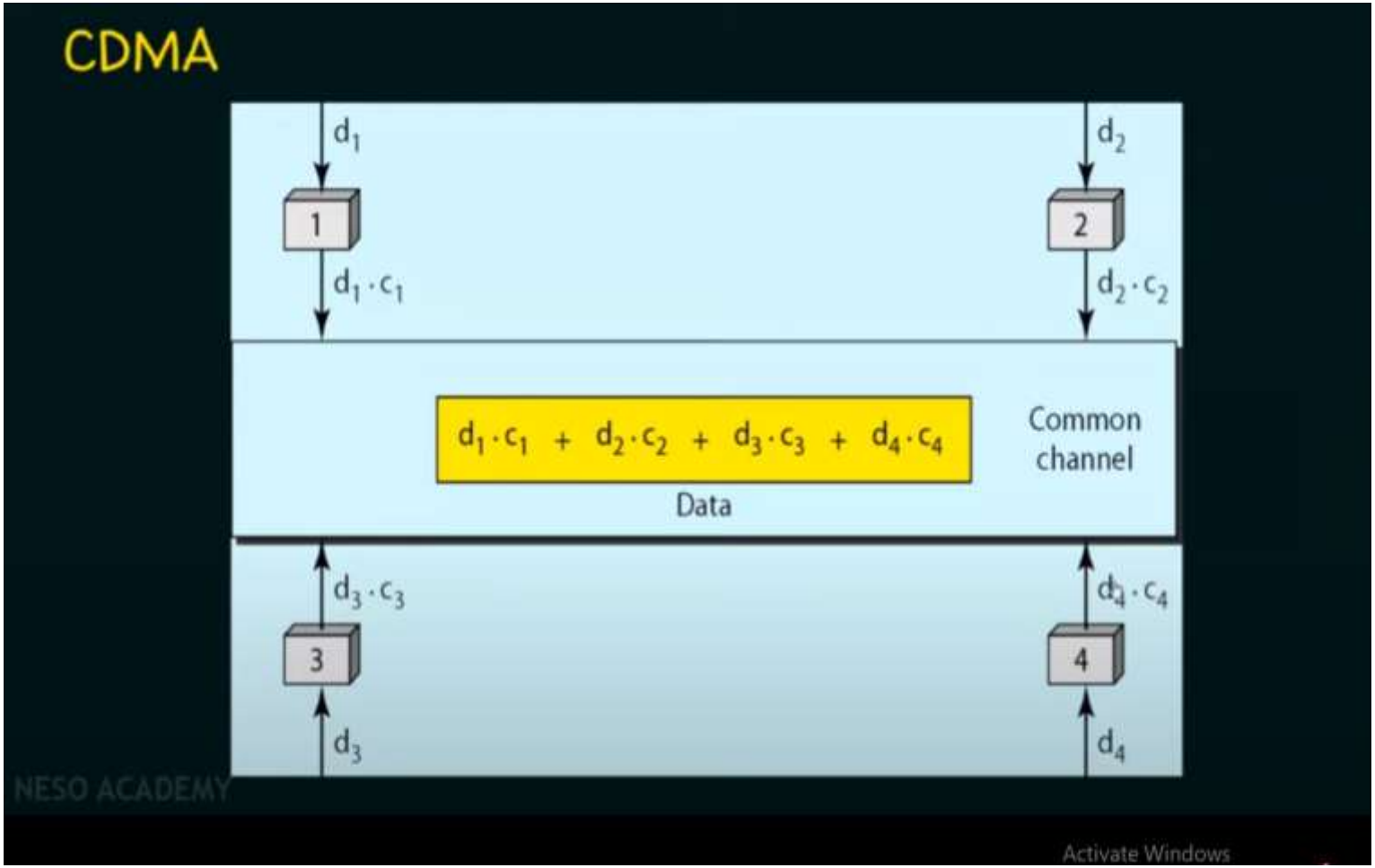


TDMA



CDMA

- ★ In CDMA, one channel carries all transmissions simultaneously.
- ★ CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link.
- ★ It differs from TDMA because all stations can send data simultaneously; there is no time sharing.


CDMA
d_1
d_2
1
2
$d_1 \cdot c_1$
$d_2 \cdot c_2$
$d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4$
Data
Common channel
$d_3 \cdot c_3$
$d_4 \cdot c_4$
3
4
d_3
d_4
NESO ACADEMY
Activate Windows

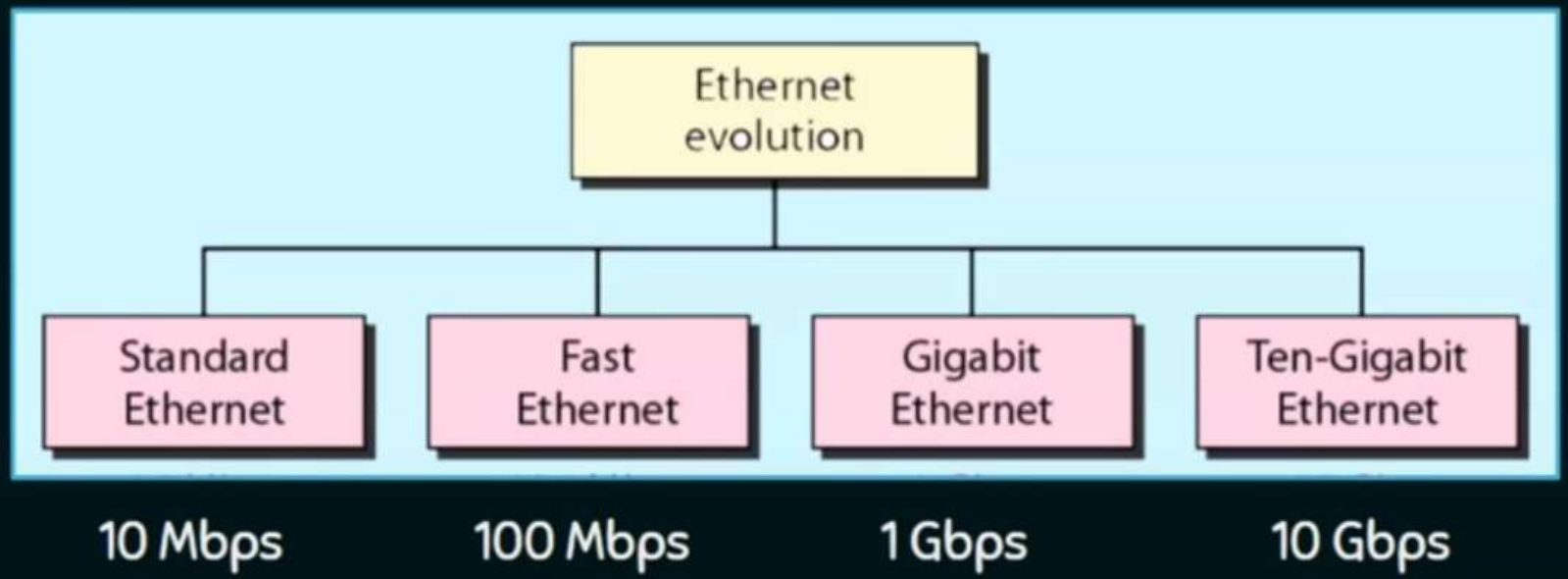
Ethernet

- ★ One of the most widely used Wired LAN technologies.
- ★ Operates in the data link layer and the physical layer.
- ★ Family of networking technologies that are defined in the IEEE 802.2 and 802.3 standards.
- ★ Supports data bandwidths of 10, 100, 1000, 10,000, 40,000, and 100,000 Mbps (100 Gbps).

Ethernet Standards

- ★ Define Layer 2 protocols and Layer 1 technologies
- ★ Two separate sublayers of the data link layer to operate – Logical link control (LLC) and the MAC sublayers.

Evolution of Ethernet



Wireless LANs

Wireless communication is one of the fastest growing technologies. The demand for connecting devices without the use of cables is increasing everywhere. Wireless LANs can be found on college campuses, in office buildings, and in many public areas.

There are two promising wireless technologies for LANs:

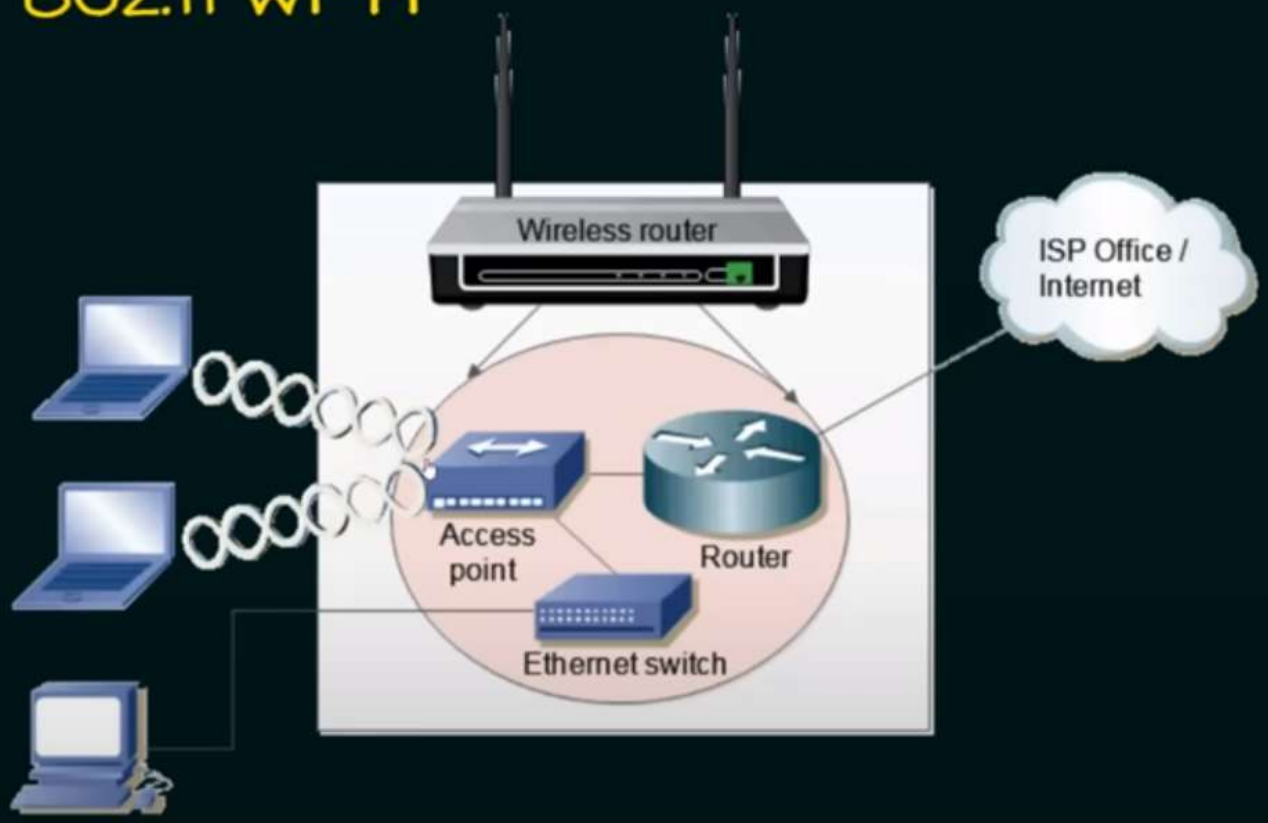
- IEEE 802.11 wireless LANs, sometimes called wireless Ethernet, and
- Bluetooth, a technology for small wireless LANs.

IEEE 802.11

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

IEEE 802.11 standard, popularly known as WiFi, lays down the architecture and specifications of wireless LANs (WLANs). WiFi or WLAN uses high-frequency radio waves instead of cables for connecting the devices in LAN. Users connected by WLANs can move around within the area of network coverage.

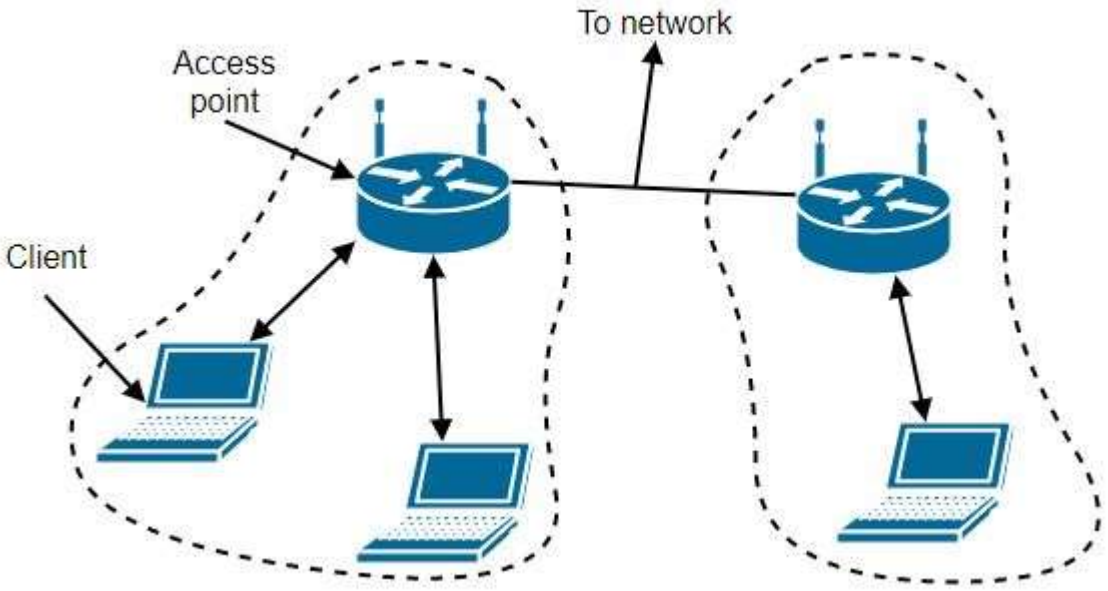
IEEE 802.11 Wi-Fi



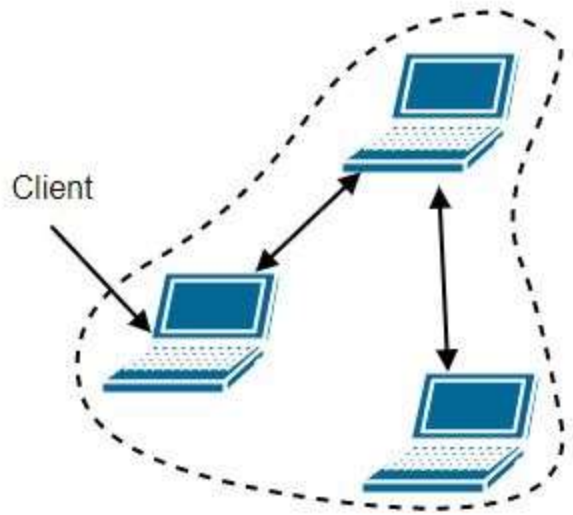
802.11 Architecture

- 802.11 networks can be used in two modes:
- Infrastructure mode
- ad-hoc mode

Infrastructure mode uses an AP (Access Point) that is connected to the network. Clients send and receive packets via the AP. Several APs can be connected to form an extended network



Ad-hoc mode is a collection of computers connected to each other so that they can send frames to each other. There's no AP



- **Advantages of wireless local area network (WLAN) :**
- It’s a reliable sort of communication.
- As WLAN reduces physical wires so it’s a versatile way of communication.
- It’s easier to feature or remove workstation.
- It provides high rate thanks to small area coverage.
- You’ll also move workstation while maintaining the connectivity.

- **Disadvantages of wireless local area network (WLAN) :**
- WLAN requires license.

- It’s a limited area to hide.

- If the amount of connected devices increases then data transfer rate decreases.

- WLAN uses frequency which may interfere with other devices which use frequency.
- Communication isn’t secure and may be accessed by unauthorized users.

Applications of WLAN

- Office/Campus Environment
- Factory Shop Floor
- Homes
- Workgroup Environment
- Heritage Buildings
- Public Places
- War/Defense Sites

- **What is broadband?**
- Broadband refers to telecommunications in which a wide band of frequencies is available to transmit information.
- Because a wide band of frequencies is available, information can be multiplexed and sent on many different frequencies or channels within the band concurrently.
- Multiplexing enables more information to be transmitted in a given time

- These types of broadband internet services use one of several different technologies, such as the following:
- Copper twisted pair.
- Ethernet via fiber optic cable.
- Coaxial cable.
- DSL from the local telephone company.Digital subscriberLine.
- 4G or 5G mobile communication technology.
- Satellite links via a fixed earth station.

Bluetooth

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.

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.

Bluetooth Architecture

Bluetooth defines two types of networks: piconet and scatternet.

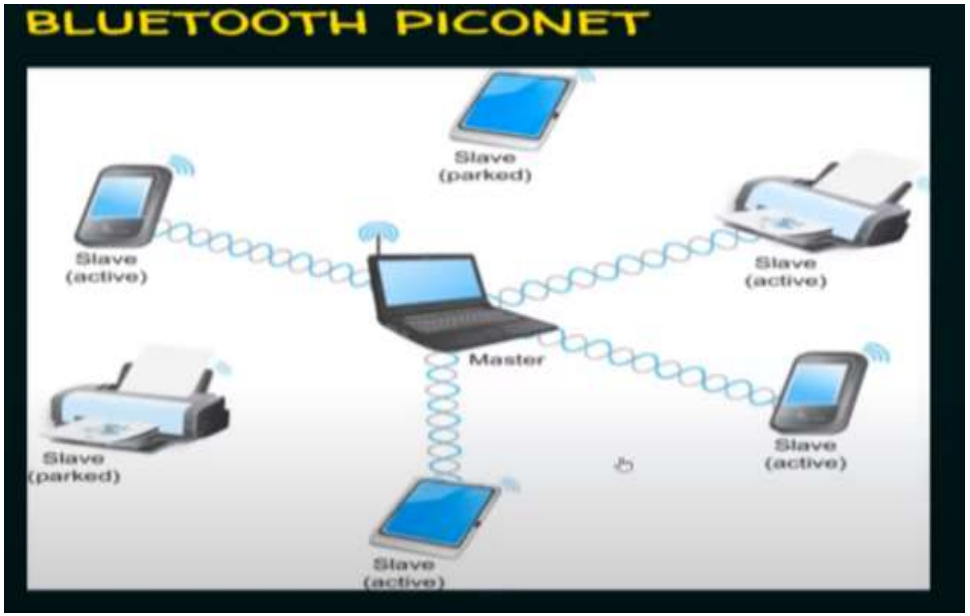
Piconets

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.

A piconet can have only one primary station.

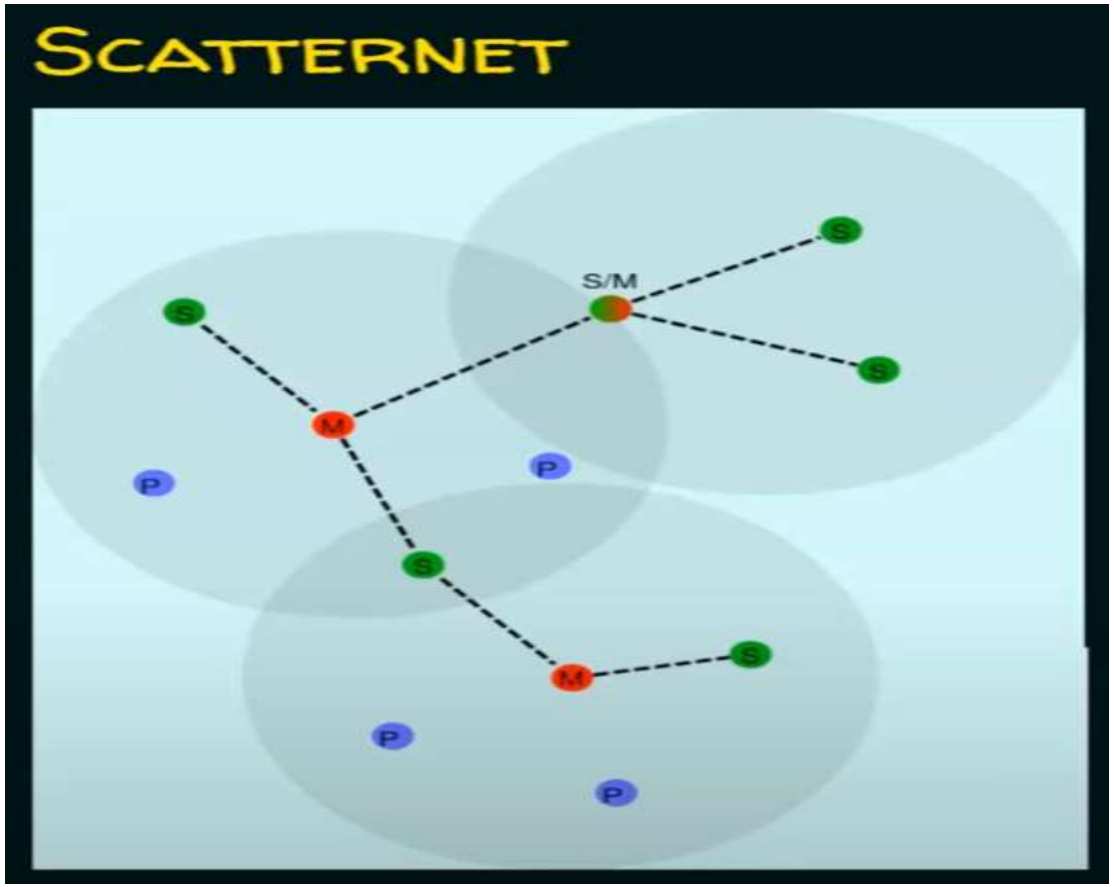
The communication between the primary and the secondary can be one-to-one or one-to-many. Figure 14.19 shows a piconet.



Scatternet

Piconets can be combined to form what is called a scatternet. A secondary station in one piconet can be the primary in another piconet. This station can receive messages from the primary in the first piconet (as a secondary) and, acting as a primary, deliver them to secondaries in the second piconet.

Figure 14.20 illustrates a scatternet.



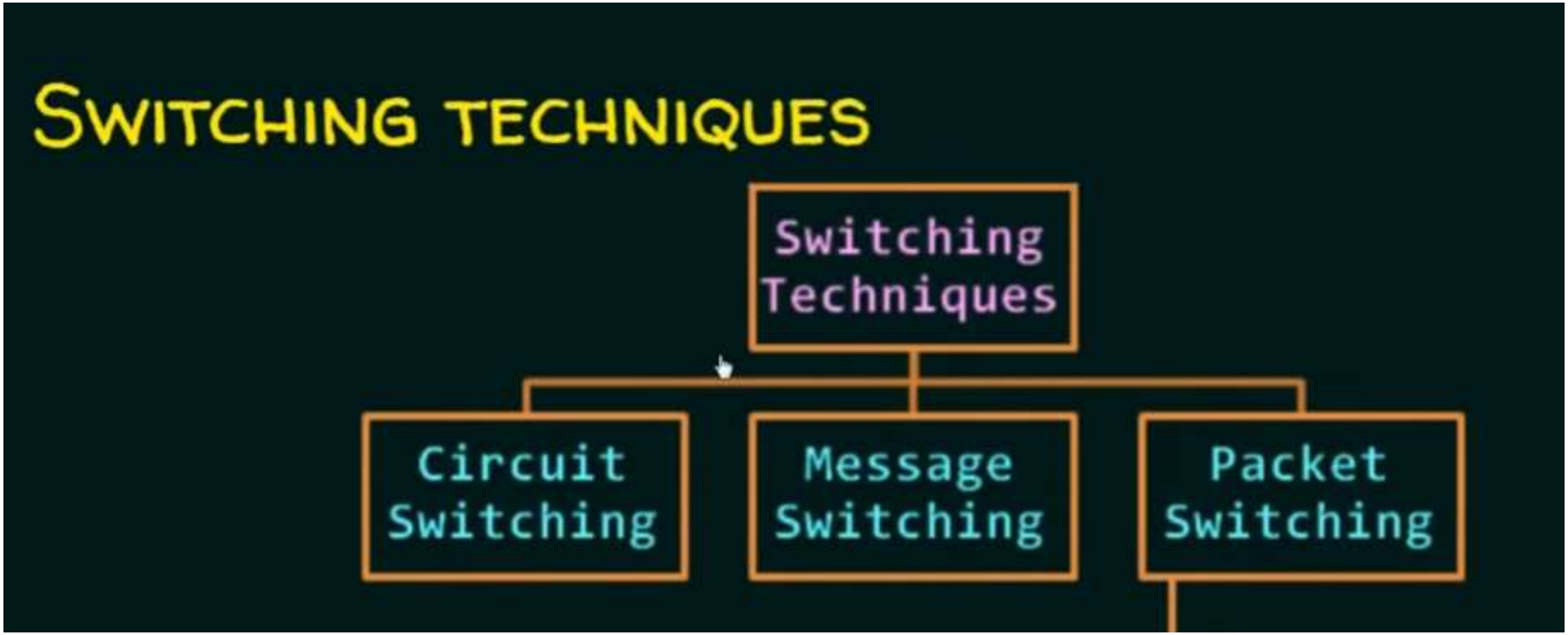
Pros

- ★ Low cost.
- ★ Easy to use.
- ★ It can also penetrate through walls.
- ★ It creates an ad-hoc connection immediately without any wires.
- ★ It is used for voice and data transfer.

Cons

- ★ It can be hacked and hence, less secure.
- ★ It has slow data transfer rate.
- ★ It has small range: 10 meters.

- **Switching** is the process of transferring data packets from one device to another in a network, or from one network to another, using specific devices called **switches**.



CIRCUIT SWITCHING

- ★ A dedicated path is established between the sender and receiver.
- ★ Before data transfer, connection will be established first.
- ★ Example: Telephone network.

3 phases in circuit switching:

1. Connection establishment.
2. Data transfer
3. Connection Disconnection.

Message switching

- ★ Store and forward mechanism.
- ★ Message is transferred as a complete unit and forwarded using store and forward mechanism at the intermediary node.
- ★ Not suited for streaming media and real-time applications.

Packet switching

- ★ The internet is a packet switched network.
- ★ Message is broken into individual chunks called as packets.
- ★ Each packet is sent individually.
- ★ Each packet will have source and destination IP address with sequence number.
- ★ Sequence numbers will help the receiver to
  - Reorder the packets.
  - Detect missing packets and
  - Send acknowledgments.