* Control Access Protocols:

→ Station seeks info. from one another to find which station has right to send
→ To avoid collision; it allows only one node/user to send at a time.

Types:

(1) Reservation:

→ A station needs 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.

2) Polling:

→ Requires one of the nodes to be designated as Master Node (Primary station)

→ Master Node polls each of the nodes in a round-robin fashion.

→ The master node itself sends an individual message to each node; saying that it can transmit upto some 'x' maximum number of frames.

→ Master Node can determine when a node has finished sending its frames by observing the lack of signal on the channel.

→ Continues until all the nodes have sent the data.

→ Higher efficiency

*→ Drawbacks:

↳ polling delay: the amount of time required to notify a node that it can transmit.

↳ If master node fails; the whole channel becomes inoperative.

*→ Functions:

↳ Poll function: primary wants to receive data; it asks the secondaries if they have anything to send.

↳ Select function: primary wants to send data; it tells secondary to get ready to receive.

→ Efficiency:

T_{poll} = time for polling

T_t = time required for transmission of data.

$$\text{Efficiency} = \frac{T_t}{T_t + T_{poll}}$$

3) Token passing:

→ A station is authorized to send data when it receives a special frame called a token.

→ A small, special purpose frame known as a token is exchanged among the nodes in some fixed order.

→ When a node receives a token, it holds onto the token only if it has some frames to transmit; else it immediately forwards the token.

→ If a node has a frames to transmit, it sends up maximum no. of frames and then forwards the token.

→ Highly efficient.

✱→ Drawbacks:

→ If one node fails; it cannot send or forward the token further resulting in crash of entire channel.
We must invoke some recovery technique; to get back token into circulation.

✱→ Performance:

T_p = propagation delay
T_t = transmission delay
N = No. of stations.

S → throughput

$a \rightarrow \frac{T_p}{T_t}$.

$$S = \frac{1}{1 + \frac{a}{N}} \quad (a < 1)$$

$$S = \frac{1}{a\left(1 + \frac{a}{N}\right)} \quad (a > 1)$$

→ Network Hardware requirements:

✱→ Network cables
✱→ Routers
✱→ Repeaters, Hubs, Switches
✱→ Bridges
✱→ Gateways
✱→ Network Interface Card → Internal / External

→ Routers: connecting device that transfers data packets b/w computer networks.

→ Repeaters: receives & retransmits signals over larger distances.

→ Hub: multiport repeater having several I/Ps and O/Ps; where I/P at any port is available at every other port.

→ Switch: receives data from a port; uses packet switching to resolve the destination device and forwards data to particular device.

→ Bridge: connects 2 Ethernet network segments.

→ Gateways: Connects different networks that work upon diff. protocols.

* Medium Access Control Sub layer: (MAC)

→ MAC is a sub layer of the data link layer of OSI model.

→ Datalink layer consists of 2 sublayers
- → Logical link control layer (LLC)
- → Medium Access control layer (MAC)

*→ Functions of MAC layer:

→ provides abstraction of physical layer to LLC and other upper layers.

→ encapsulates frames so that they are suitable for transmission.

→ resolves addressing of source station as well as destination station.

→ performs collision resolution & initiates retransmission.

→ generates frame sequences.

*→ MAC address is a unique identifier alloted to a NIC card of a device.

* Carrier Sense Multiple Access Protocol:

→ Minimize chance of collision and increase the performance.

→ Principle: Sense before transmit (or) Listen before talk.

→ Carrier busy ⇒ transmission taking place.

Carrier idle ⇒ No transmission.

*→ chance of collision still exists due to propagation delay; if the distance b/w sender & receiver is longer.

→ If propagation delay is high; worse the performance of the protocol.

*→ CSMA/CD: Collision detection.

→ If 2 stations sense the channel; idle and begins the transmission; they will detect a collision immediately.

→ In this protocol; rather than finish transmitting the frames; which irretrievably garbled; the transmission must stop

abruptly as soon as collision is detected.

→ Quicker termination of damaged frames saves time & bandwidth

→ widely used on LANs in MAC sub layer; also used by ethernet.

→ ∴ CSMA/CD model will consist of alternating contention & transmission periods; with idle periods occurring when all stations are quiet.

Efficiency for CSMA/CD $= \frac{1}{1+6.44*a}$ $\quad a = \frac{T_p}{T_t}$

→ If distance ↑ses; efficiency ↓ses.

→ works for LAN rather than WAN

→ length of packet is bigger; efficiency of CSMA ↑ses. (max length ≤1500 bytes)

→ Transmission Time = Round Trip Time of 1 bit

= 2 * Propagation time.

* → CSMA/CA: Collision Avoidance.

→ Nodes attempt to avoid collisions by beginning transmission only after the channel is sensed to be idle.

→ Important for wireless networks where CSMA/CD is not possible due to wireless transmitters desensing their receivers during packet transmission.

→ unreliable due to
hidden node problem
exposed terminal problem.

→ operates in data link layer; also used by WIFI.

Connection Oriented (TCP)	Connection - Less (UDP)
→ Telephone system.	→ Postal system.
→ preferred for long & steady communication	→ preferred for bursty communication
→ feasible & necessary; reliable	→ Not feasible & not necessary. and not reliable.
→ Congestion is not possible.	→ Congestion is possible.
→ Packets follow same route & requires authentication	→ Packets do not follow same route & does not require authentication

* Stop and wait protocol:

→ data link layer protocol for noiseless channels
→ unidirectional flow control & do not deal with error control facilities
→ After transmitting one frame; the sender waits for an acknowledgement before transmitting next frame.

*↔ Sender Side:
- → Send one data packet at a time.
- → Send next packet only after receiving the acknowledgement.

*↔ Receiver Side:
- → Receive and consume data packet.
- → After consuming the packet; ACK need to be sent.

*↔ Drawbacks:
- → Data is lost due to which both Sender and receiver wait for infinite amount of time.
- → ACK sent by receiver is lost due to which sender waits for a long time.
- → Timeout problem due to delay of ACK/data.

* Stop and wait ARQ protocol

→ If the acknowledgement does not arrive ~~after~~ after a certain period of time; the sender times out and retransmits the original frame.

⇒ Stop & Wait ARQ = Stop & Wait + Timeout Timer + Sequence Number of frame.

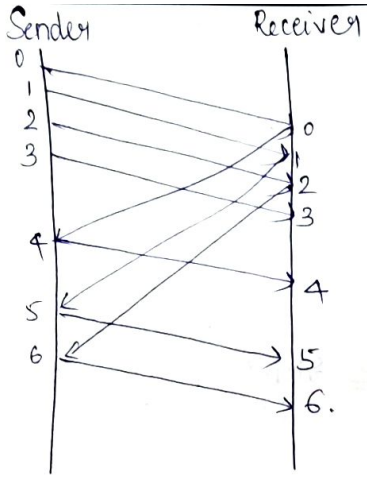
*↔ Drawbacks:
- → One frame at a time
- → poor utilization of bandwidth
- → poor performance.

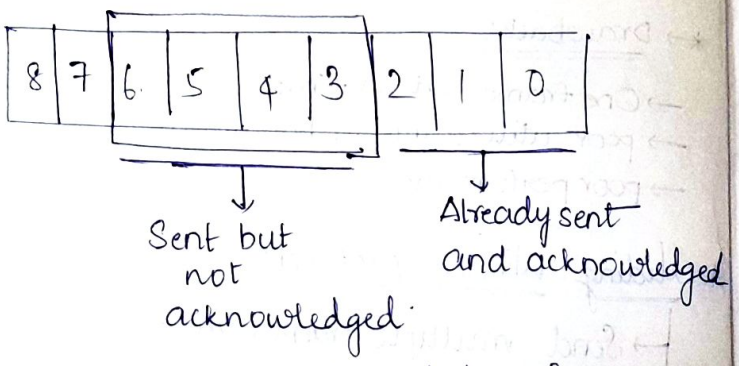
* Sliding Window protocol:
- → Send multiple frames at a time.
- → No. of frames to be sent depends on window size.
- → Each frame is numbered: Sequence number.

8	7	6	5	4	3	2	1	0

Window size: 4

Before receiving an ACK; the sender can send 4 packets.



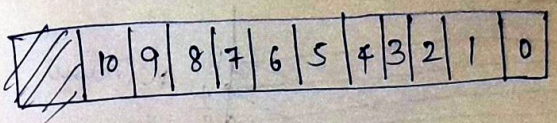


* Go Back - N: → Sender window size

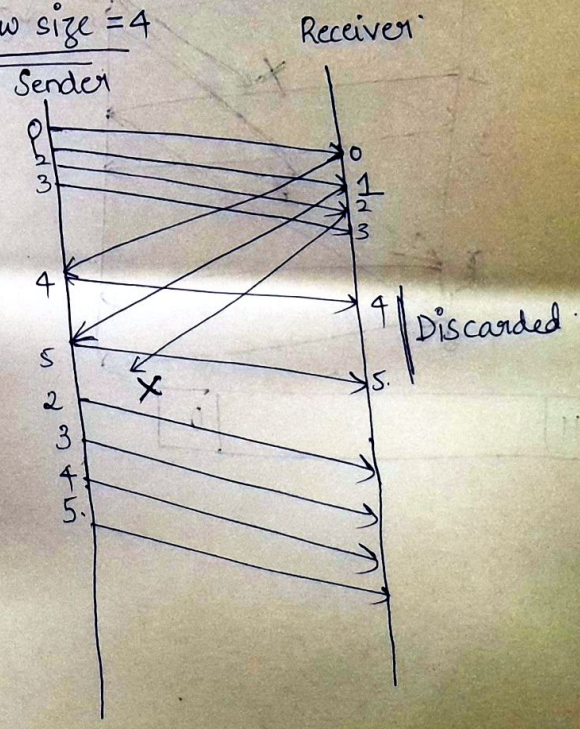
- Sender can send multiple frames before receiving the acknowledgement for the first frame.
- Frames are numbered in sequential order.
- No. of frames that can be sent depends on sender window size.

- If acknowledgement of a frame is not received within an agreed upon time period, all frames in the current window are transmitted.

* Ex: N - Sender's window size.
  N = 4 (2^2)
  ⇒ Sequence Nos shall be
  0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, ...



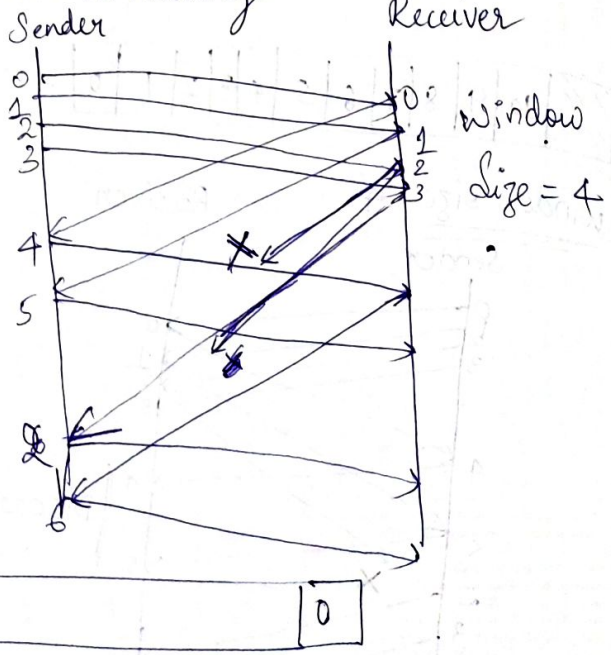
window size = 4



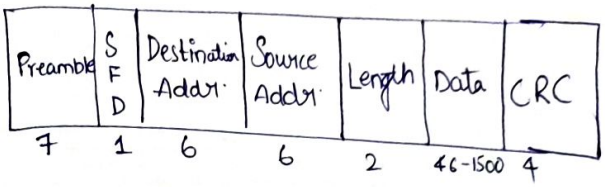
* Selective Repeat ARQ:

→ In selective repeat ARQ, only lost frames are retransmitted; while correct frames are received.

→ Keeps track of Sequence numbers and sends -ve ACK for only one frame which is missing.



* Ethernet Frame format:



Preamble: pattern of alternative 0's & 1's which indicate starting of frame and allow sender & receiver to establish bit synchronization.

SFD → Start of frame delimiter