# Computer Networks 1

A computer network is a collection of various computing devices. Main function is to share data. If there is a connection between sender and receiver and proper protocols are being followed then it is called **proper communication**.

Inter-process communication is when sender and receiver both in same system. Example: Pressing a key on keyboard to display something on monitor. This is handled by operating system, not computer networks. Computer network only handles cases where client and server are on separate systems.

**Functionalities of CN:**

1. **Mandatory**:
    1. Error Control
    2. Flow Control
    3. Multiplexing and demultiplexing

    and so on...
1. **Optional**
    1. Encryption and Decryption
    2. Checkpoint

    and so on...

| OSI Model | 4-layer TCP/IP Model | 5-layer TCP/IP Model |
|---|---|---|
| Open System Interconnect | Transmission Control Protocol/Internet Protocol | Transmission Control Protocol/Internet Protocol |
| Developed by ISO | Developed by ARPANET | Developed by ARPANET |
| Application Layer | Application Layer(Process to Process) | Application Layer |
| Presentation Layer | " " | " " |
| Session Layer | " " | " " |
| Transport Layer | Transport Layer (Host to Host) | Transport Layer |
| Network Layer | Internet Layer (Source to Destination | Network Layer |
| Data-link Layer | Network Access Layer (Node to Node) | Data-link Layer |
| Physical Layer | " " | Physical Layer |

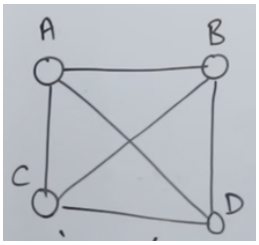
| | | |
|---|---|---|
| Physical Layer | | Physical Layer |

**Physical Layer and its functionalities:**

1. Cables and Connections
2. Physical Topology
3. Hardwares (Repeaters, Hubs)
4. Transmission Mode
5. Multiplexing
6. Encoding

**Physical Topologies:**

1. Mesh: All devices connected to all others.




n=no. of nodes

**No. of Cables:** (n*(n-1))/2 or ${}^{n}C_{2}$

**No. of Ports(No of connections on each node):** (n-1) ports on each node so total, n*(n-1) ports

**Reliability(If there is one failure, does system breakdown?):** Very high reliability because even if one connection fails there are more connections to each node

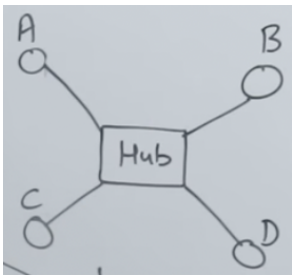
**Cost:** Cost is high because many cables needed for each node

**Security:** High security because suppose A and D are talking to each other, since they have direct connections B and C won't be able to know anything.

**Maintenance**: High because way too many cables to handle

**Point to Point:** Mesh supports point to point connection/dedicated communication. This is good for small connections like LAN but not where there are many nodes.

2. Star/Hub: It has one centralised device called hub(Multi-port device)/central node and all other nodes are connected to it.




n=no. of nodes

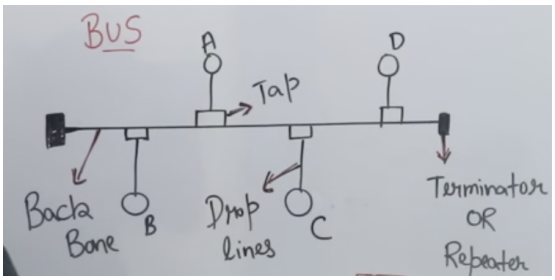
**No. of Cables:** n

**No. of Ports(No of connections on each node):** 1 port on each node so total, n ports

**Reliability(If there is one failure, does system breakdown?):** Very less reliability because if hub fails, the whole system fails and no connection anywhere

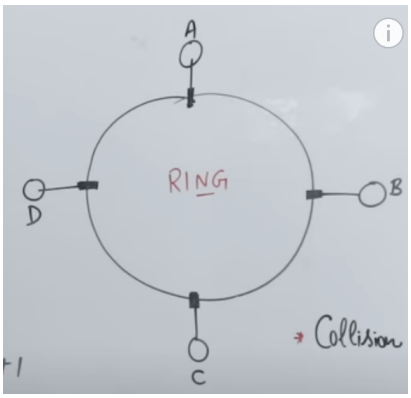
**Cost:** Cost is medium as no. of cables are less but cost of hub is a bit high
**Security:** Medium security because only hub broadcasts message to everyone
**Point to Point:** Star supports point to point connection/dedicated communication. This is good for small connections like LAN but not where there are many nodes.

3. Bus: all devices are connected by one central RJ-45 network cable or coaxial cable also referred to as the bus, backbone, or trunk.




n=no. of nodes
**No. of Cables:** n+1(1 for bus)
**No. of Ports(No of connections on each node):** 1 port on each node so total, n ports
**Reliability(If there is one failure, does system breakdown?):** Very less reliability because if bus backbone fails, the whole system fails and no connection anywhere (Single point of failure)
**Cost:** Cheaper than mesh as number of cables less
**Security:** No security because everyone has same communication channel.
**Collision:** It is a multi-point network as multiple devices are connected to one wire so collision can be very high. So like if all nodes start transmission at same time all signals can overlap.

4. Ring: nodes create a circular data path (basically join both ends of bus topology). It is unidirectional.

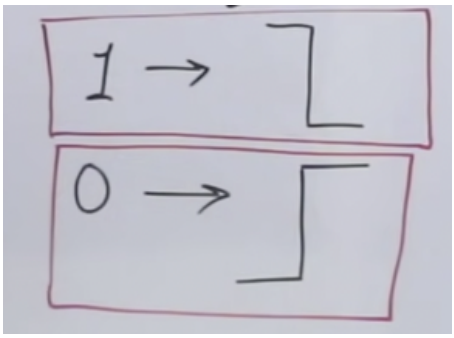


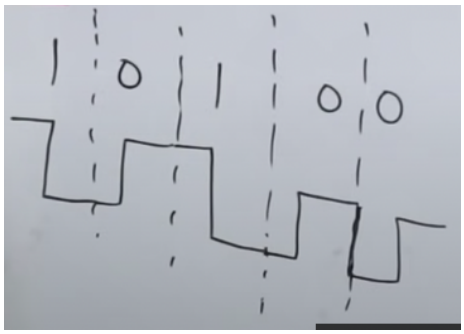

n=no. of nodes
**No. of Cables:** n+1(1 for circular ring)
**No. of Ports(No of connections on each node):** 1 port on each node so total, n ports
**Reliability(If there is one failure, does system breakdown?):** Very less reliability because if ring fails, the whole system fails and no connection anywhere (Single point of failure)
**Cost:** Cheaper than mesh as number of cables less
**Security:** No security because everyone has same communication channel.
**Collision:** It is a multi-point network as multiple devices are connected to one wire so collision can be very high. So like if all nodes start transmission at same time all signals can overlap.

**Manchester encoding and differential manchester encoding:** They are digital-to-digital encoding

methods that convert binary 1s and 0s into a digital signal.

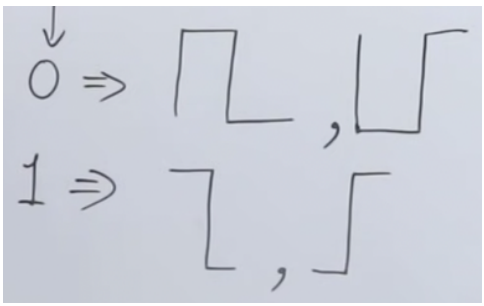
Manchester encoding:

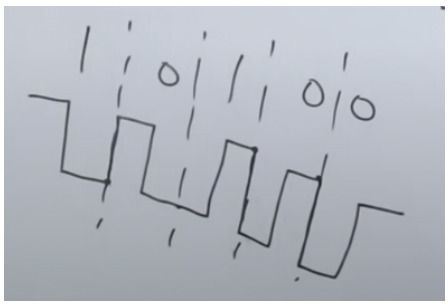



This is Dr. Thomas notation and IEEE 802.3 notation is complete opposite of this.

Differential manchester encoding:




0 always has starting with edge. Depending on where the last point you can choose how to draw 1 or 0 from the above options

**Various Devices In Computer Networks:**

1. Cable: Pure Hardware
2. Repeaters: Pure Hardware
3. Hubs: Pure Hardware
4. Bridges: Hardware+Software
5. Switches: Hardware+Software
6. Routers: Hardware+Software
7. Gateway
8. IDS: Security
9. Firewall: Security
10. Modem

**Cables:** (Physical Layer)

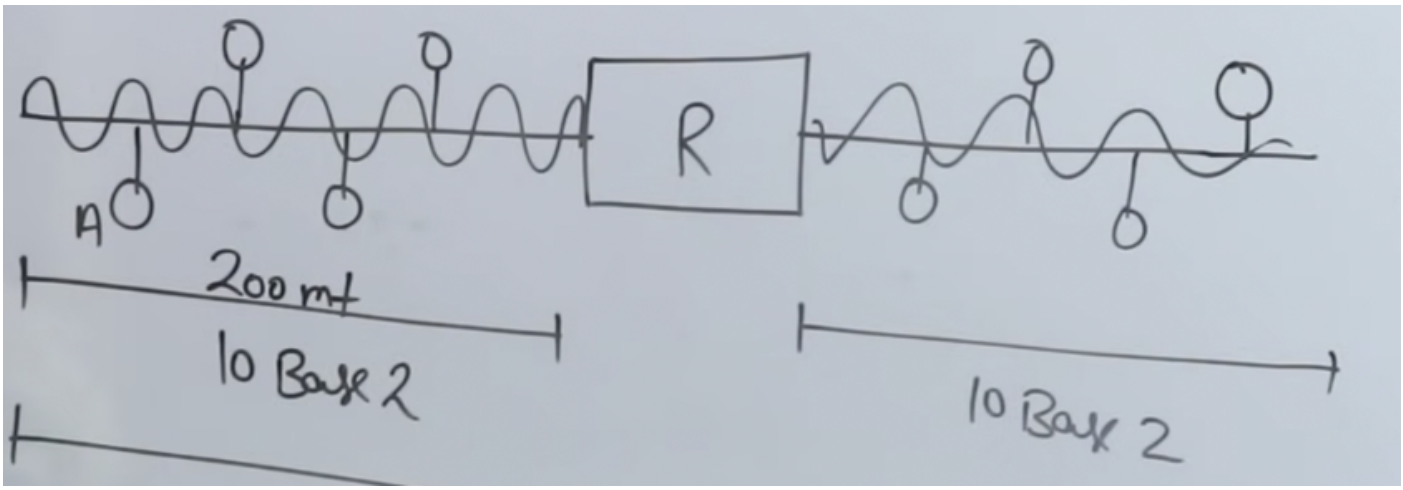
1. Unshielded Twisted Pair cable(UTP): UTP cable is a 100 ohm copper cable that consists of 2 to 1800 unshielded twisted pairs surrounded by an outer jacket. They are represented like 10 Base T, 100 Base T. 10 represents 10 Mbps(Megabits/sec), Base means only one signal can be transmitted at a time, T means 100meters i.e. signals can move upto 100m properly after that attenuation will occur(signal weakens). UTP cables are mostly used in Ethernet, LANs.
2. Coaxial cable, or coax is a type of electrical cable consisting of an inner conductor surrounded by a

concentric conducting shield, with the two separated by a dielectric; many coaxial cables also have a protective outer sheath or jacket. It is represented as 10 Base 2, 10 Base 5 etc

3. An optical fiber is a flexible, transparent fiber made by drawing glass or plastic to a diameter slightly thicker than that of a human hair. It is represented as 100 Base Fx. Fx means fibre cable and is approximately 2km.

**Repeater:** (Physical Layer)

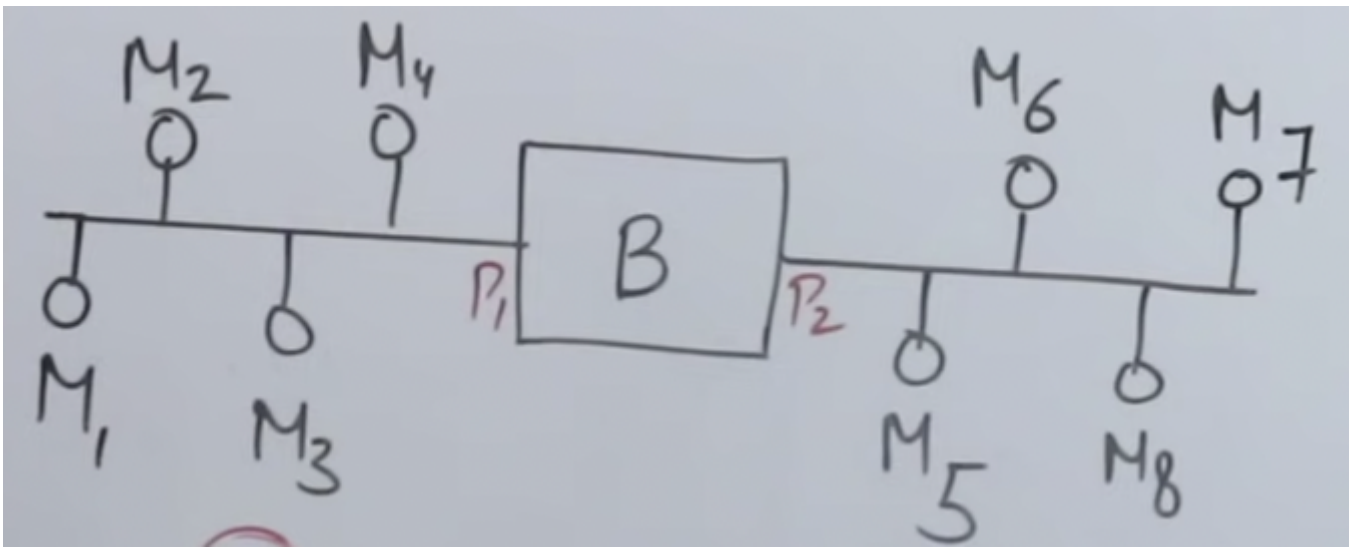
- Repeaters are network devices that amplify or regenerate an incoming signal before retransmitting it.
- It is a 2-port device.
- Repeater forwards the signal.
- Repeater can not filter signals because it is completely a hardware signal and doesn't know if any signal has to be filtered because there is no software.
- Collision domain is 'n' because there is no buffer, if all nodes start communicating together, all signals will overlap.
- Difference between repeater and amplifier: Amplifier increases the amplitude. When attenuation happens and the signal weakens, the repeated regenerates the strength of the signal to its original value.




**Hub:** (Physical Layer)

- Hub is a multiport repeater
- Forwarding happens in hub
- No filteration so traffic is high. Broadcasts all messages to everyone
- Collision domain is n
- Has extra features like tells you which port is working


**Bridges:** (Physical Layer and Data Link layer)




- can connect two different LANs (like one with ring topology and one with bus topology)
- does forwarding but can also stop when needed, so if the message is from M1->M3, it won't forward, but if it's from M2->M6, it will forward
- It can filter according to whether the message needs to be send further or not
- When message is sent, it also sends Source MAC Address and Destination MAC Address
- For filtering and forwarding, there are two ways to know what to do
    - Static: The network administrator manually creates a table in the bridge with the MAC Address and which port each MAC Address is connected to.
    - Dynamic: Initially the table in the bridge is empty. When messages are sent and the entry for that particular destination is missing in the table, the source just broadcasts it and notes down the entry once the message is received. This way the bridge learns and forms the table.

| MAC Address | Port |
|---|---|
| M1 | P1 |
| M2 | P1 |
| M3 | P1 |
| M4 | P1 |
| M5 | P2 |

| | |
|---|---|
| M6 | P2 |
| M7 | P2 |
| M8 | P2 |

- Collision is very low because bridge uses "store and forward" i.e. it has a buffer which stores the signals and then one by one the message is forwarded.
- Bridge Data Unit Protocol is used to remove loops. If I message is stuck in a loop, spanning tree is created to know which way to forward the message

**Switch:** (Data Link layer)

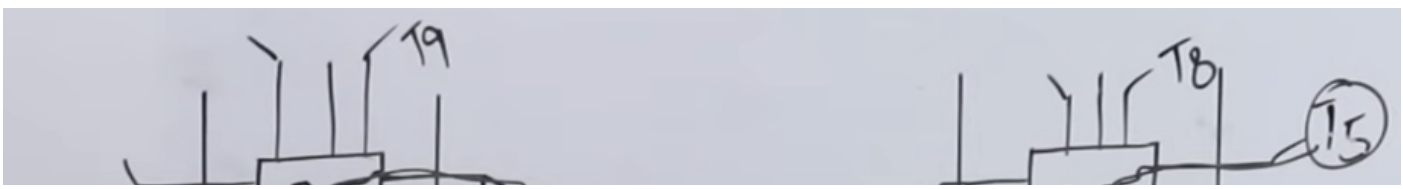
- multiport bridge
- full duplex links (suppose switch send a message to A, at the same time A can send message to switch without collision)
- collision is zero and traffic is minimal

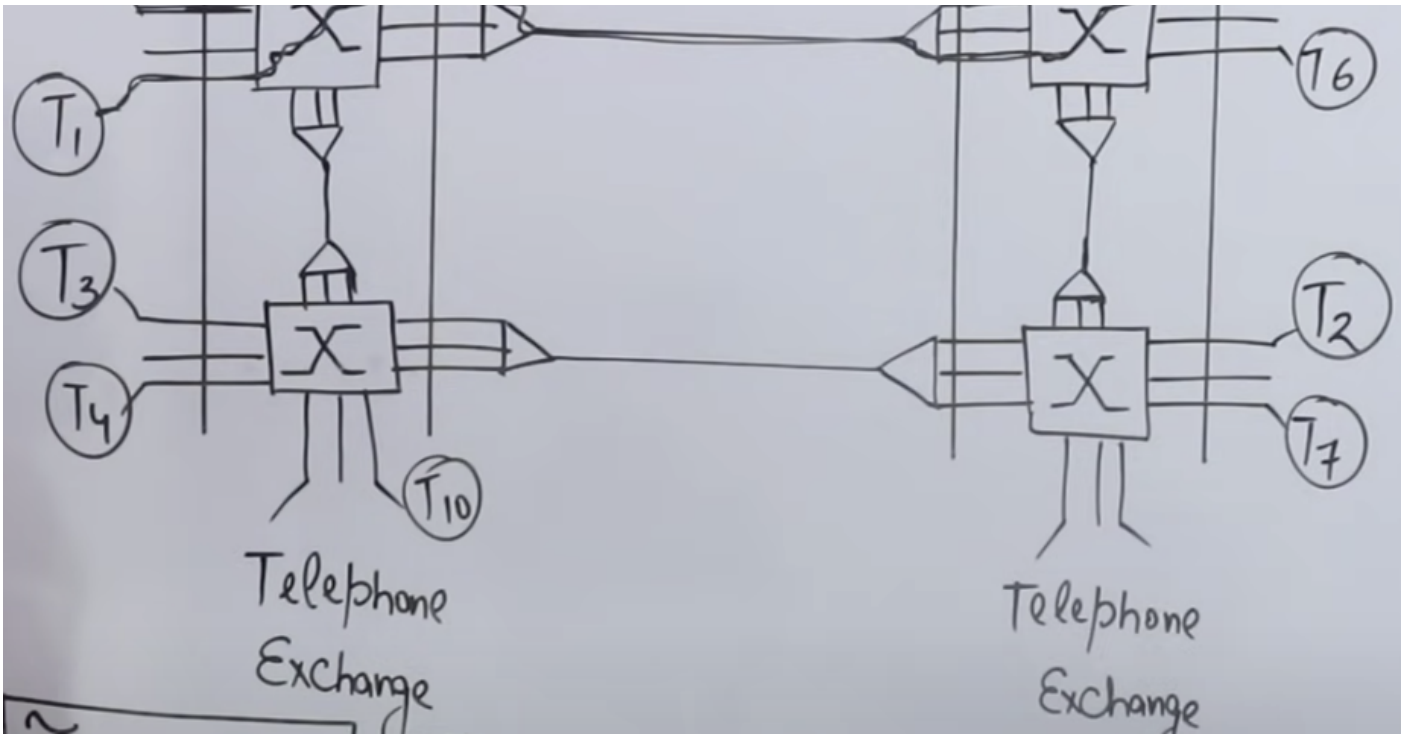
**Router:** (Physical layer, Data Link Layer, Network Layer)

- Stores IP addresses instead of MAC address
- Router connects networks all over the world
- Allows forwarding and filtering
- Has a routing table which contains all IP addresses
- In case the routing table is not able to figure where to send the message it does flooding i.e. broadcasts the message everywhere
- Collision is low because it uses store and forward technique

| **Device** | **Collision Domain** | **Broadcast Domain** |
|---|---|---|
| Repeater | No change | No change |
| Hub | No change | No change |
| Bridge | Reduce | No change |
| Switch | Reduce | No change |
| Router | Reduce | Reduce |

**Circuit Switching:**






- Physical Layer
- Suppose T1 wants to talk to T5, so a setup will be created and then the connection will stay there till it's terminated
- A dedicated path is created
- Once connection is made, there will be contiguous flow of data in the same order it is sent
- No headers are used because no address and all is needed. Once connection is made you need nothing, no IP address, no MAC address nothing
- While setup you reserve resources. You reserve the bandwidth of the dedicated path
- Efficiency is low because once resource is reserved it can't be shared with other users
- Delay is less because there is no checking or processing of signals or data, it just keeps going
- Total time = Setup time(reserve resources and create connection) + Transmission time(message/bandwidth) + Propagation delay(distance/velocity) + Tear Down Time(unreserve the resources)
- In short Total time = Setup time + TT + PD + Tear Down Time

**Packet Switching:**

- Data link layer and network layer divide the contiguous data into packets
- Two types of packet switching
  - Datagram (Network Layer)
  - Virtual Circuit (Data Link Layer)
- Packets are sent to switch and then the switches stores it in the buffer, decides the route and then sends forward (store and forward)
- Higher efficiency
- Higher delay because of store and forward
- Pipelining used to increase efficiency and decrease delay
- Total time = n(TT) + PD

| Datagram Switching | Virtual Circuit |
|---|---|
| Connectionless | Connection oriented |
| No reservation | Reservation |
| Out of order | Same order |
| High overhead (headers) | Less overhead |
| Packet loss high | Packet loss low |
| Used in Internet | Used in ATM, X.25 |
| Cost lower | Cost higher |
| Delay higher | Delay lower |

**Message switching:**

- Predecessor of packet switching
- Store and forward
- Hop by hop delivery
- Higher efficiency than circuit switching
- Higher delay than circuit switching

**Types of casting:**

1. Unicast: one to one communication
2. Broadcast: two types
    1. limited broadcast (broadcast to all nodes within the same network)
    2. direct broadcast (broadcast to all nodes in another network)
3. Multicast: one-to-many or many-to-many

---

Data Link Layer: Takes data from network layer and gives it to physical layer
Responsibility:

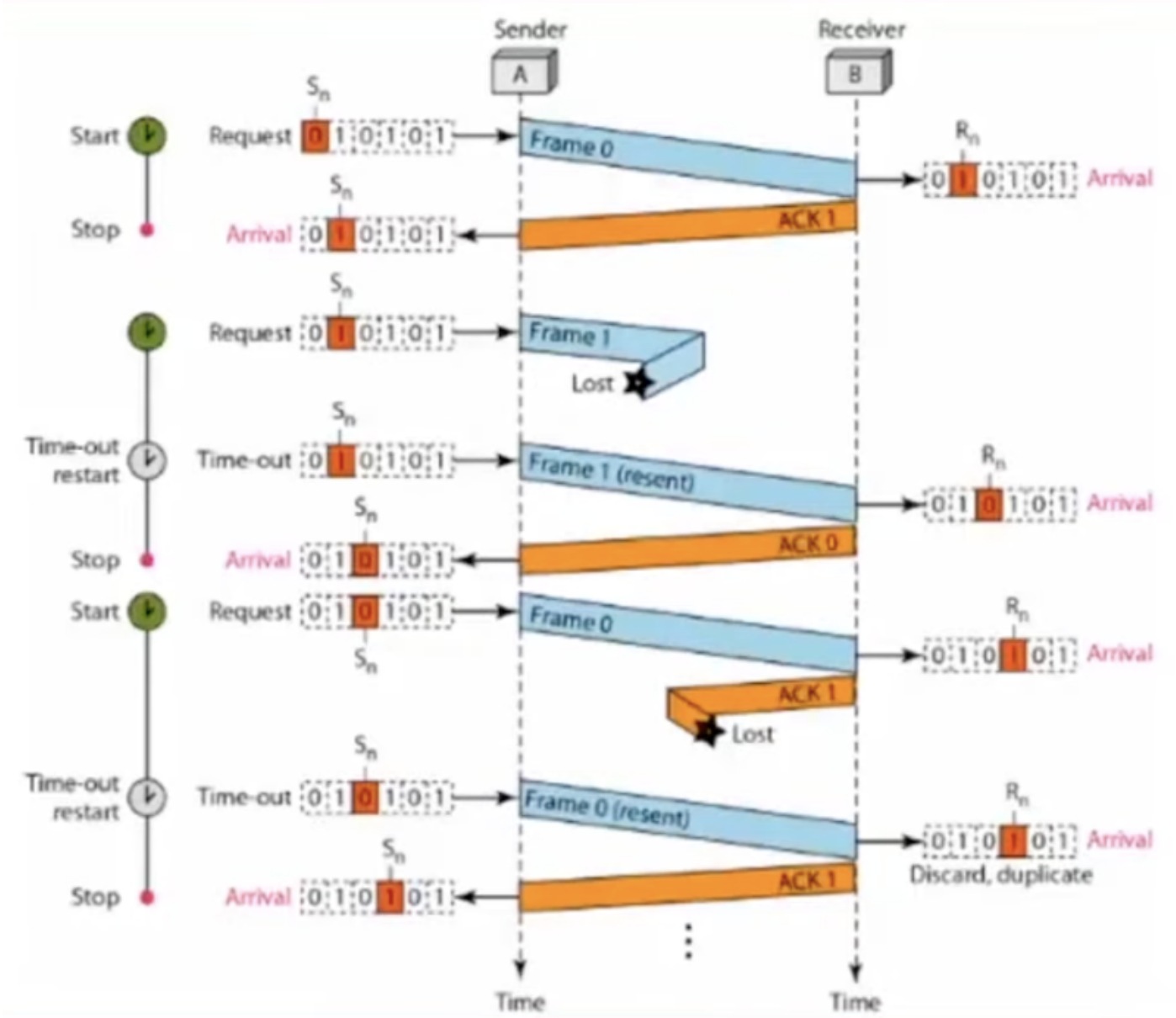
1. Hop to hop/Node to node delivery
2. Flow control:
    1. Stop and Wait
    2. Go-Back-N
    3. Selective Repeat
3. Error Control
    1. CRC
    2. Parity
4. Access Control
    1. CSMA/CD

2. Aloha
3. Token Ring/Bus

5. Physical Address (MAC)

**Stop-and-wait:**

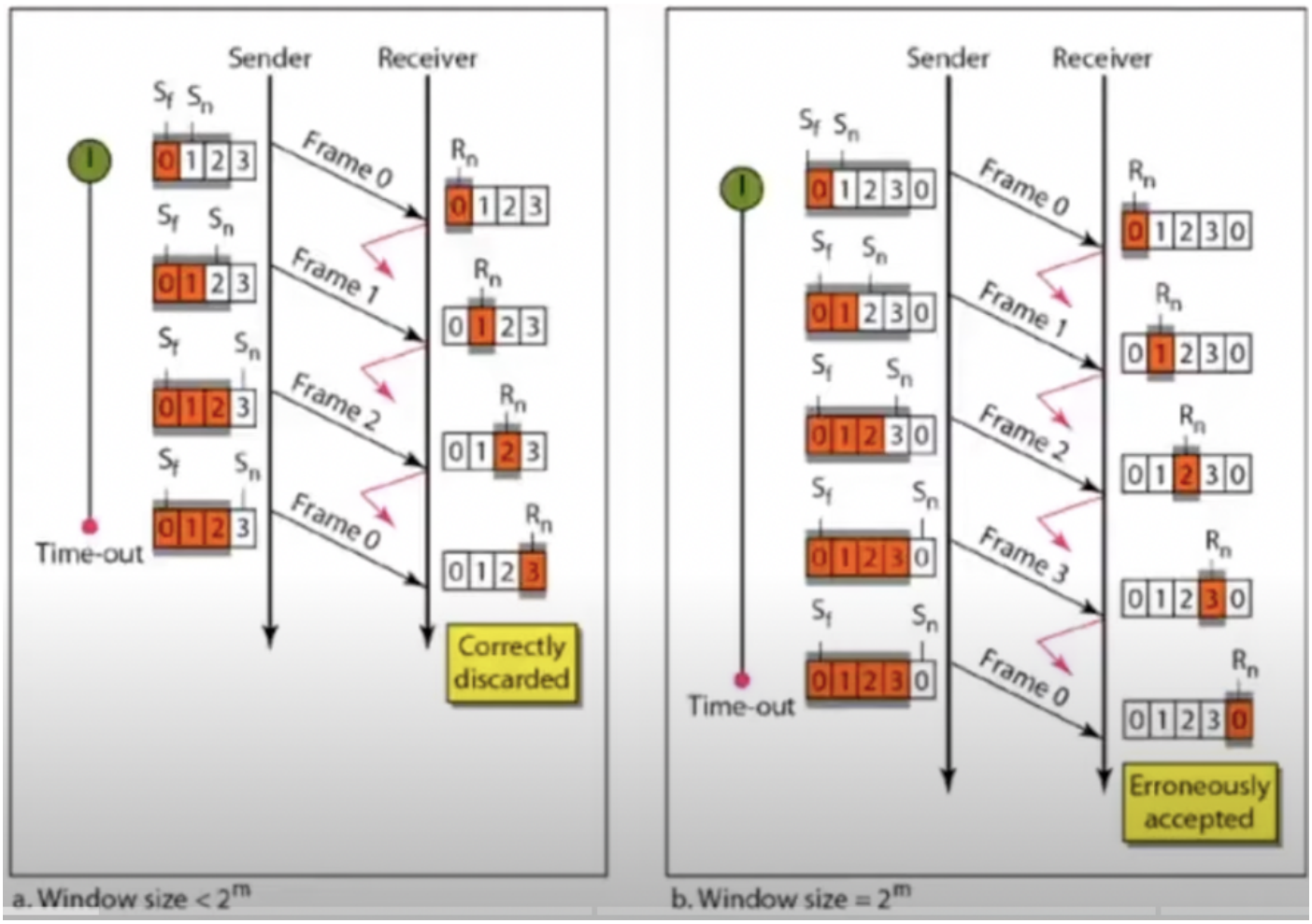
- Error correction is done by keeping a copy of the sent frame and retransmitting of the frame when the time expires.
- We use sequence numbers to number the frames. They are based on modulo-2 arithmetic
- The acknowledgement number always announces in modulo-2 arithmetic the sequence number of the next frame expected.



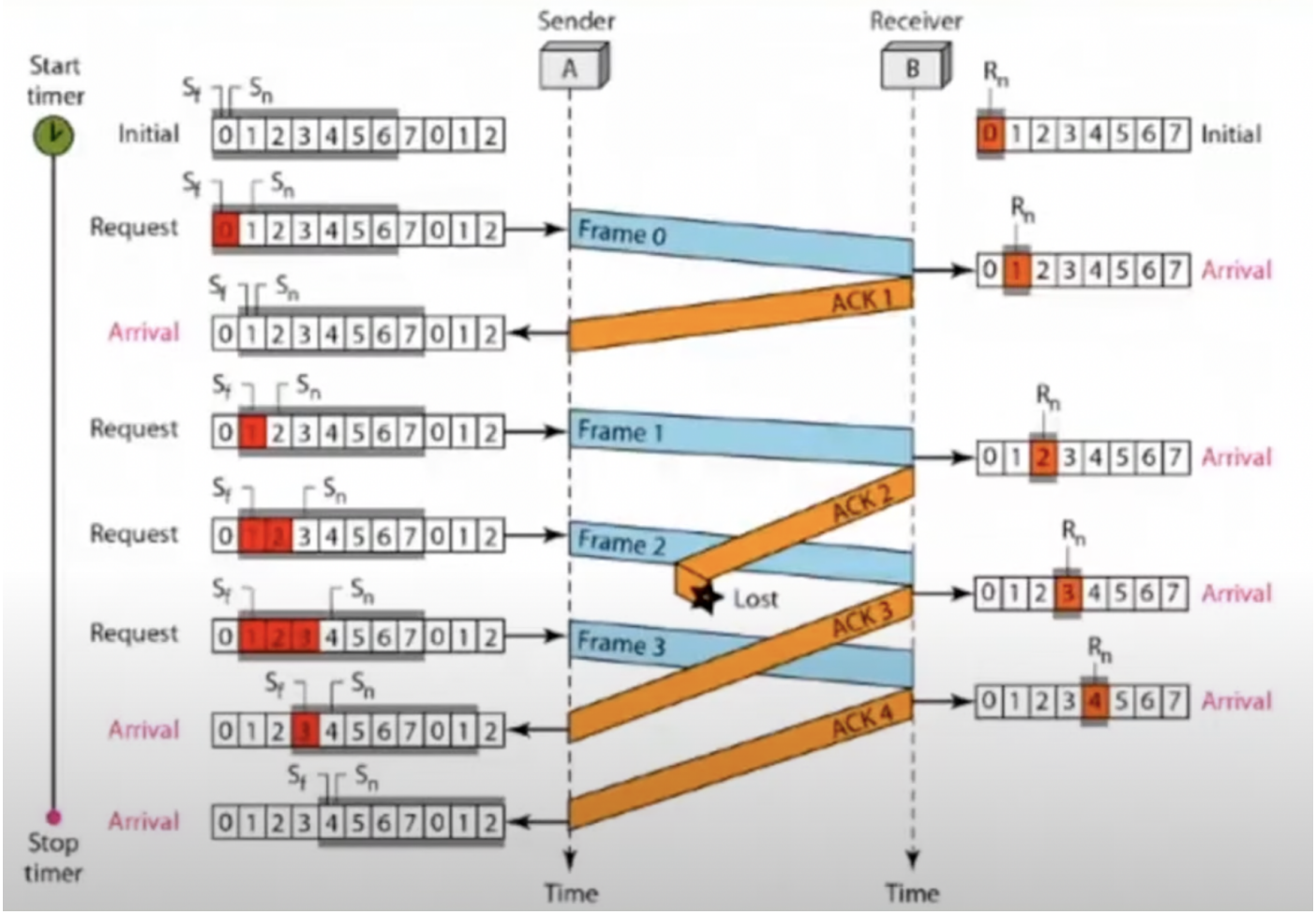

**Go-Back-N:**

- The sequence numbers are $2^m$ where m is the size of the sequence number field in bits
- The send window is an abstract concept defining an imaginary box of size $2^m - 1$ with three variables: $S_f$, $S_n$, $S_{size}$
- The send window can slide one or more slots when a valid acknowledgement arrives
- The receive window is an abstract concept defining an imaginary box of size 1 with one single variable $R_n$. The window slides when a correct frame has arrived. Sliding occurs one slot at a time

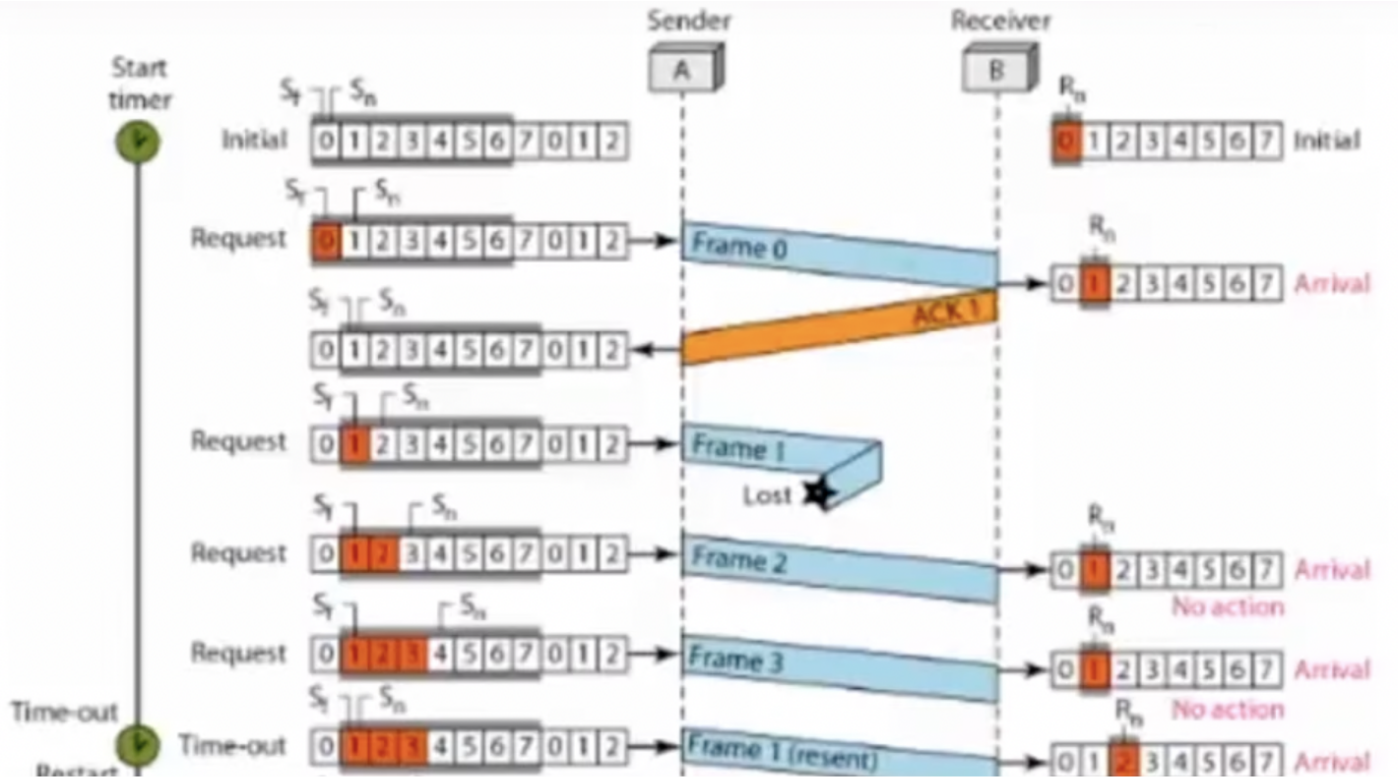
- Why window size should be $2^m$ - 1?




- If acknowledgement gets lost in between but next acknowledgement is received, then sender will assume that previous frame was also received.



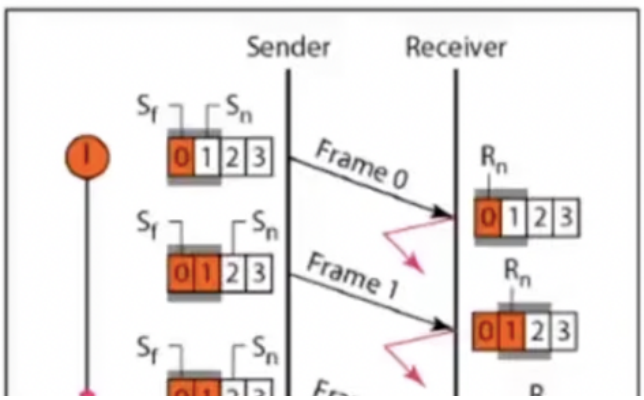
- Can not receive out of order packet because window size of receiver is 1

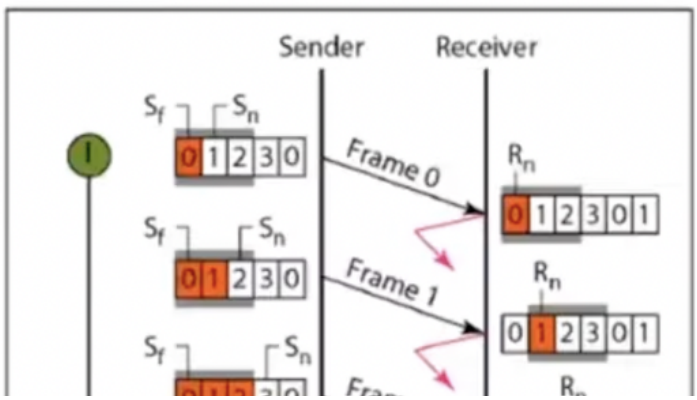


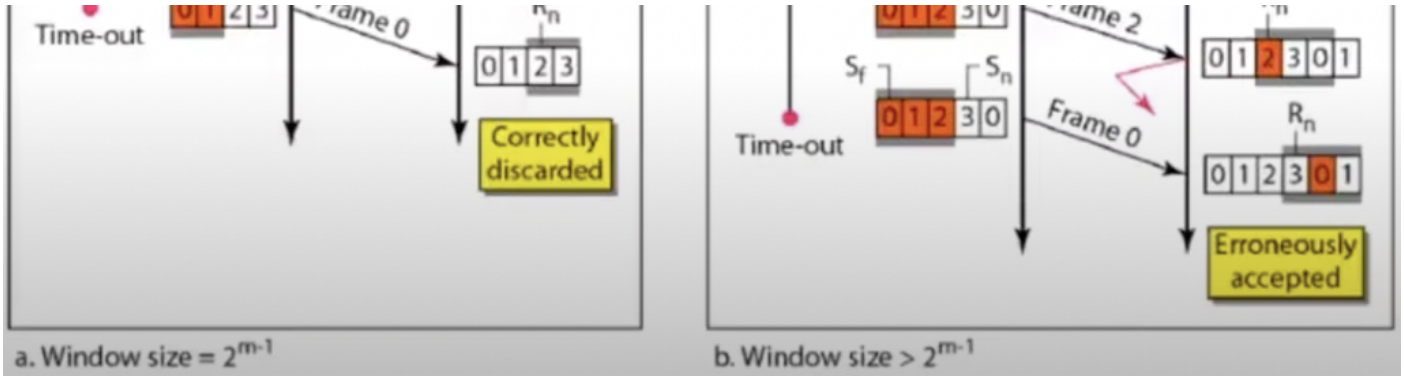

**Selective Repeat:**

- Window size of both sender and receiver is $2^{m-1}$
- Sequence numbers are $2^m$

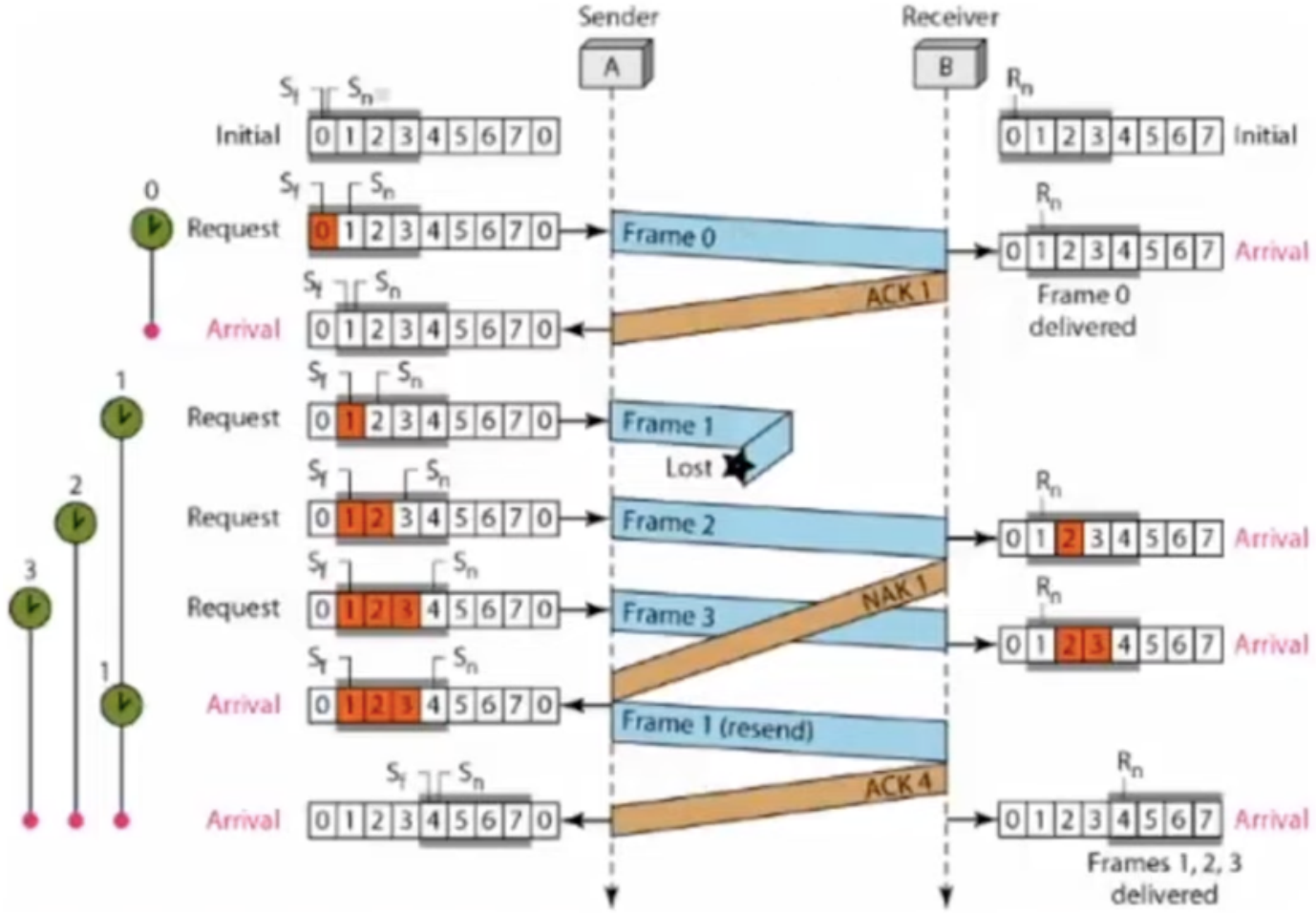
Frame size has to be $2^{m-1}$






NAK is negative acknowledgment




Flow control protocol summary:

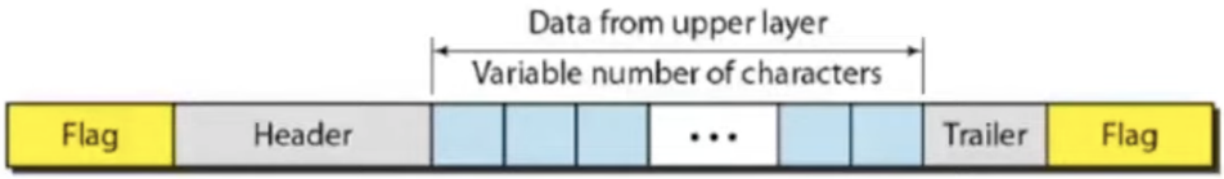
| **Stop and wait** | **Go Back N** | **Selective Repeat** |
|---|---|---|
| Only 1 frame transmit | Multiple frames | Multiple frames |
| Sender window=1 | Sender window=$2^k$-1 | Sender window=$2^{k-1}$ |
| Receiver window=1 | Receiver window=1 | Receiver window=$2^{k-1}$ |
| Efficiency(η)=1/(1+2x) where x=$T_p/T_t$ | Efficiency(η)=($2^k$-1)/(1+2x) | Efficiency(η)=$2^{k-1}$/(1+2x) |

| | | |
|---|---|---|
| Retransmission=1 | Cumulative ACK<br>Retransmission=$2^k$-1 | Cumulative and independent ACK<br>Retransmission=1 |

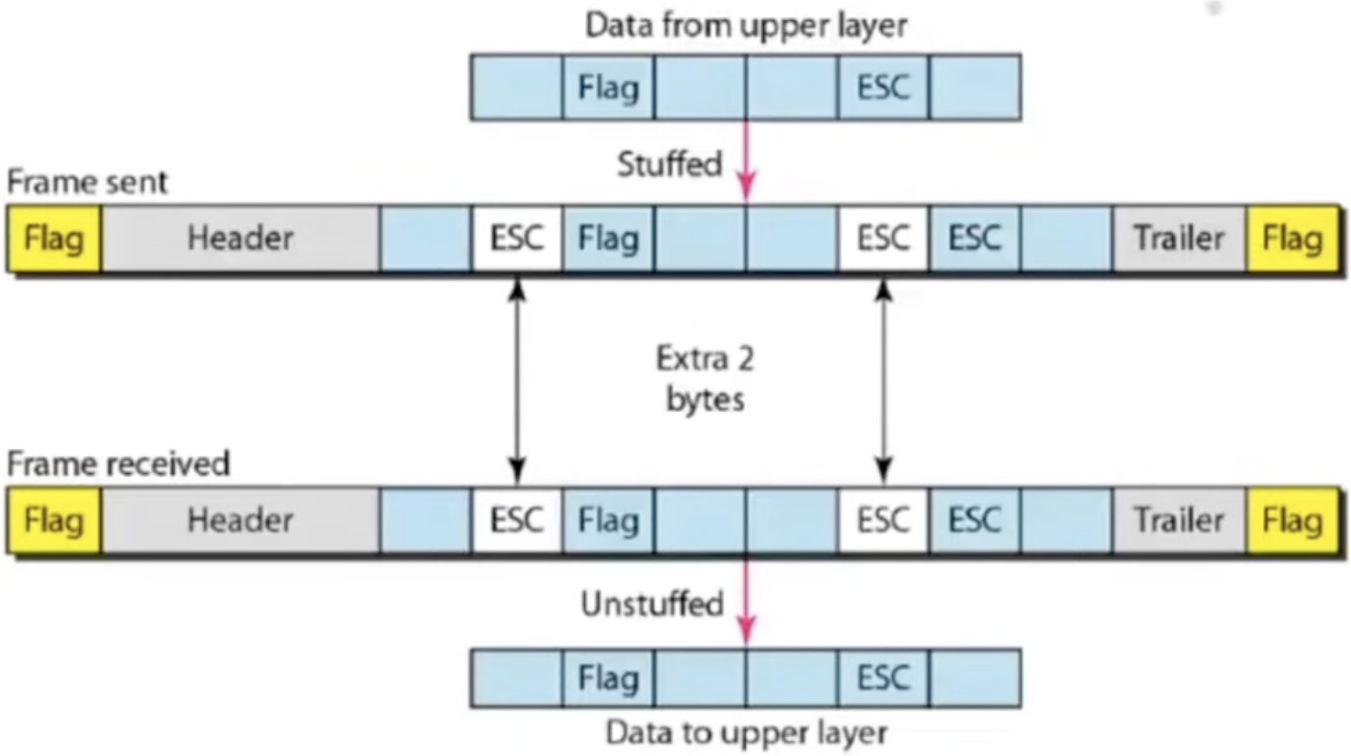
Efficiency=TT/(TT+RTT)
TT=Transmission time
RTT=Round trip time
$T_p$=Propagation delay
$T_t$=Transmission time
SW=sender window size
RW=receiver window size

**Framing:**

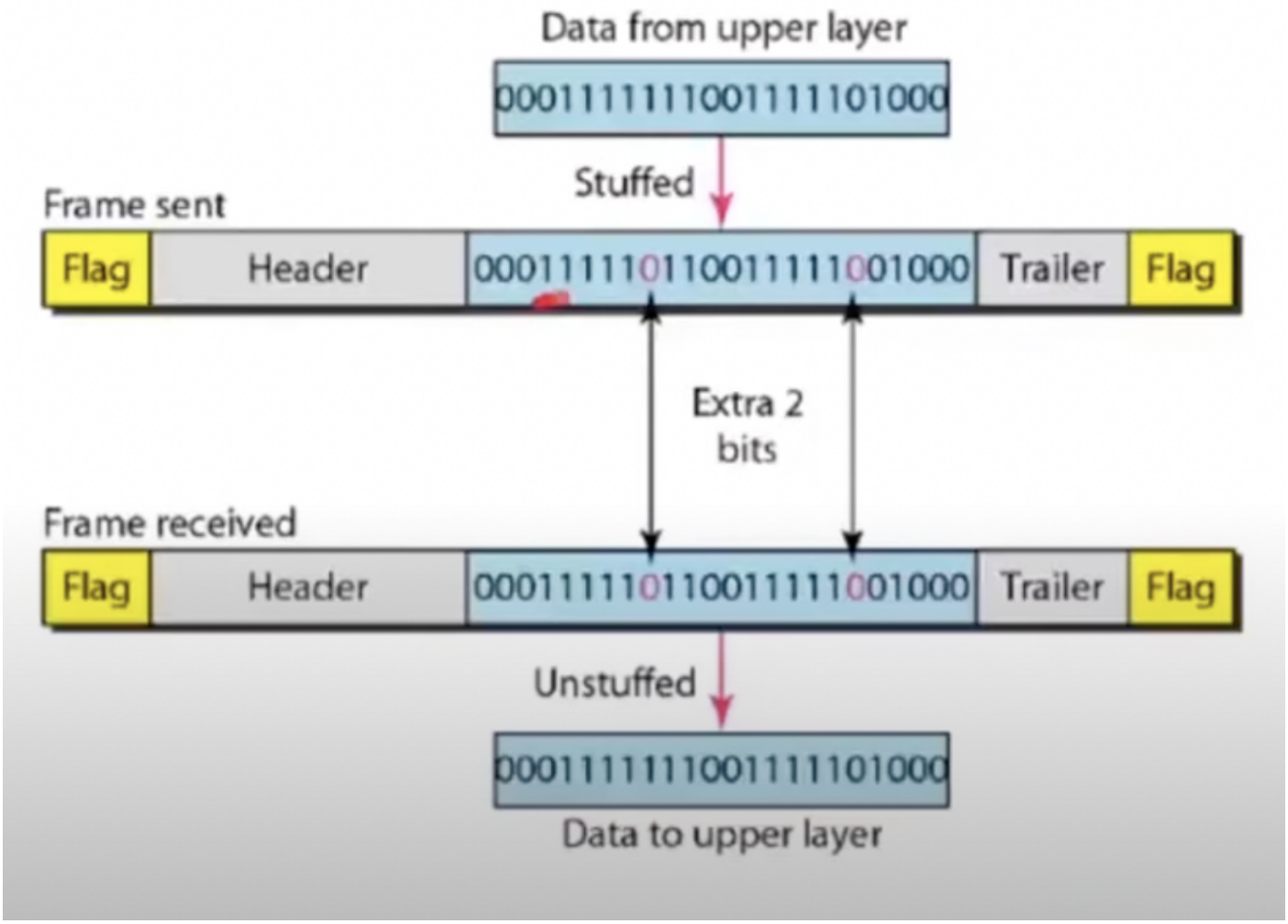
- Data link layer packs bits into frames so that each frame is distinguishable
- Byte-stuffing:
  - frame is character-oriented protocol, that means we stuff characters between flags like Flag ABCD Flag




  - In case Flag is also there in the data we use ESC, Example: data is AFlagB, then stuffing will be Flag A ESC Flag B Flag




- Bit-stuffing: If you get 5 continuous 1s, add a zero after it




---

**Error Detection and Correction:**

- **Detection:**
  - Simple Parity
  - 2D Parity Check
  - Checksum
  - CRC (Cyclic Redundancy Check)
- **Correction:**
  - Hamming codes

- Two types of errors:
  - Single Bit Error (1 bit changed)
  - Burst Error (Many bits changed)

**Simple Parity:**

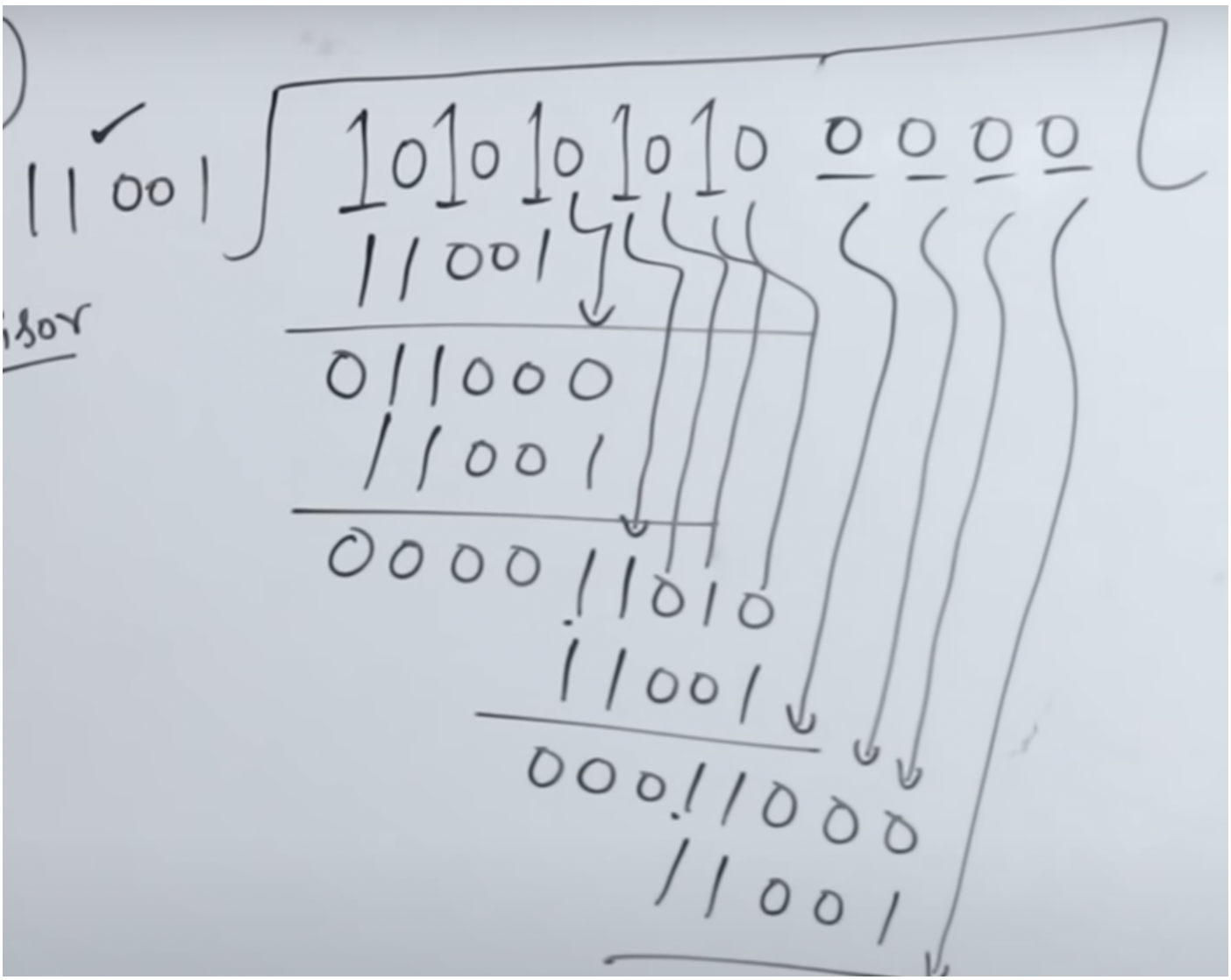
- m+1 bits
- even parity (no. of 1s should be even)
- can detect single bit errors
- can detect all odd number of errors too
- Example: Data is 0100 then parity bit is 1. So data sent will be 01001. So now if receiver gets 01101, then you can see the parity bit is wrong as there are 3 1s now. So receiver gets to know there is error

then you can see the parity bit is wrong as there are 3 1s now. So receiver gets to know there is error

- Find Hamming distance of two numbers:
  - first do xor of two numbers
  - count number of 1s in the result
- Minimum hamming distance in 4 digit numbers is 2
- Example of hamming distance: 1001 XOR 0101=1100 (So 2)
- If minimum hamming distance is d, then we can check errors of d-1 bit errors

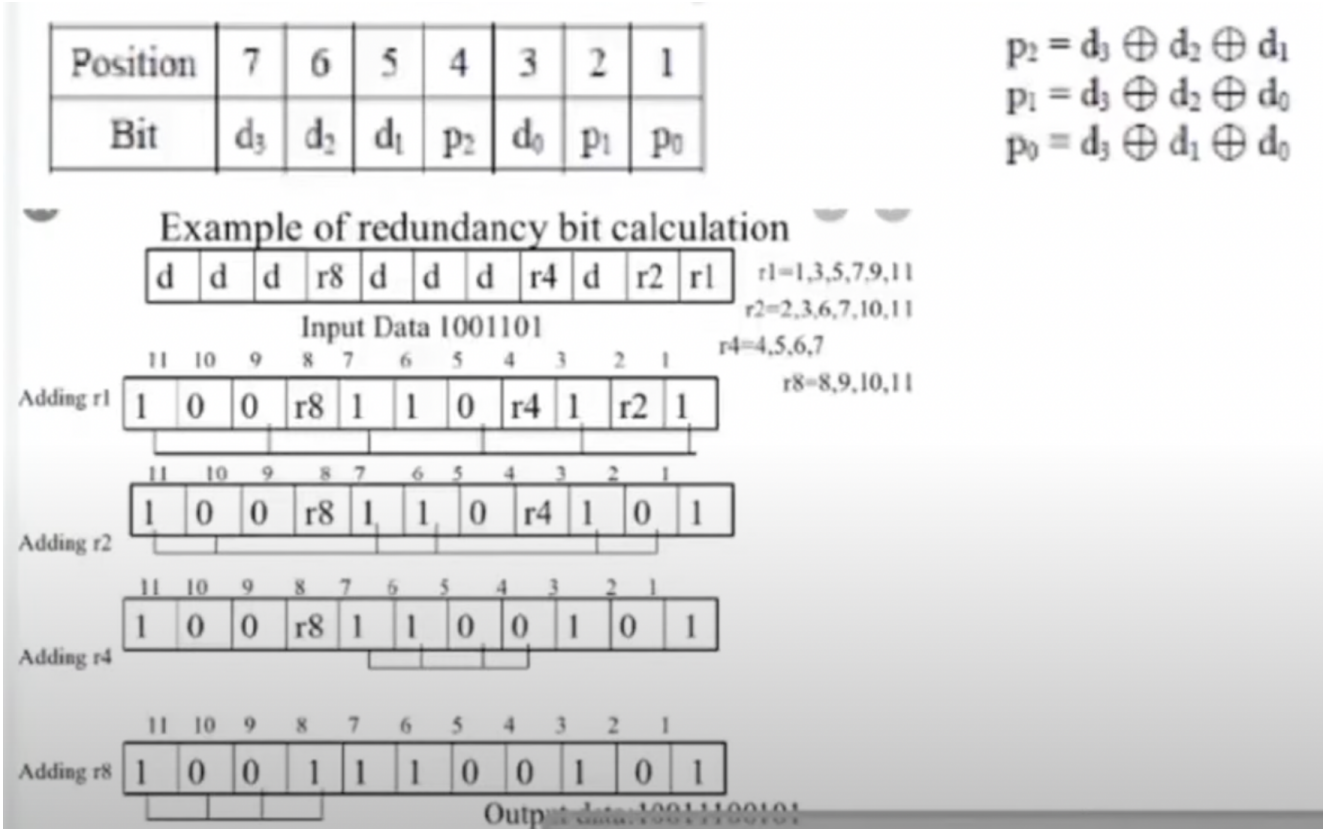
**CRC (Cyclic Redundancy Check):**

- Based on binary division
- Polynomial should not be divisible by x or x+1
- Total bits = m+r where m = number of bits in message, r = number of redundant bits
- Can detect all odd errors, single bit errors and burst errors of length equal to polynomial degree
- How to solve:
  - Let message be 1010101010 which will be the **dividend**
  - The polynomial is $x^4 + x^3 + 1$
  - Find coefficients of polynomial to find divisor i.e $1x^4 + 1x^3 + 0x^2 + 0x^1 + 1x^0 = 11001$ which will be the **divisor**
  - Add 4 zeroes to the dividend since that is the highest degree in polynomial
  - If direct polynomial is given, append number of digits of divisor -1 to the dividend
  - Now divide by XOR:




- From the remainder take last 4 bits and replace the zeroes we appended in the dividend.
- So final message that will go is 1010101010**0010**
- Now if receiver wants to check if message is correct, just divide the message sent by divisor and the remainder must be zero if data is right
- Efficiency=(10/14)*100

**Hamming code for error detection and correction:**




- So redundancy bit positions will be at $2^n$ i.e. at position=1,2,4,8,16,...
- Now for r1 take 1 bit, then skip 1 bit, then take 1 bit, then skip 1 bit so r1 will be xor of 1,3,5,7,9...
- For r2 take 2 bits, skip 2 bits, then take 2 bits and skip 2 bits so r2 will be xor of 2,3,6,7,10,11..
- Similarly for r4 take 4 bits so r4 will be xor of 4,5,6,7,..
- Now basically to check if parity bit is right, count the number of 1s and check if it matches the even parity. If it does write 0 and if it doesn't write 1
- For example: Let the message received be 10**1**11100101. Now for r1 i.e. 1,3,5,7,9,11 i.e. 110101 there are 5 1s that means parity is wrong so we'll mark r1 as 1. Similarly we do for all parity bits and we get r8 r4 r2 r1 as 1 0 0 1 which in decimal is 9. Therefore the error is in the 9th bit!!

**Multiple Access Protocols/Medium Access Control(MAC) Protocols:**

When multiple people are trying to access data through a shared link(like bus topology), MAC protocols are needed to avoid collisions.

- Random Access Protocols
  - Aloha

    - CSMA
    - CSMA/CD or CSMA/CA
- Control Access
    - Polling
    - Token Passing
- Channelisation Protocol
    - FDMA
    - TDMA

**Pure Aloha:**

- random access protocol (anyone can send a data at any time)
- Acknowledgement system is there (So on receiving a message receiver sends a ack and if sender doesn't receive it, it means there is a collision)
- LAN based
- Only transmission time, No propagation time
- Vulnerable time(i.e. collision won't happen if we make sure we have this much time before sending) = 2*Transmission Time
- Efficiency = $G \times e^{-2G}$ where G is the number of stations who want to transmit the data in the transmission time slot
- Maximum Efficiency (find by differentiating efficiency)

  $d\eta/dG = G \times e^{-2G} \times (-2) + e^{-2G} \times (1) = 0$

  $\Rightarrow e^{-2G} ( -2G +1 ) = 0$

  $\Rightarrow -2G +1 = 0$

  $\Rightarrow G = 1/2$

  So replacing value of G to get max efficiency

  $\eta_{max} = 1/2 \times e^{-1} = 0.184 = 18.4\%$ (which is very low)

  So when half of the total stations transmit at a time, efficiency is highest

**Slotted Aloha:**

- The timeline is divided into fixed number of slots assuming all messages have same transmission time
- Size of slot is the transmission time
- It is necessary that a sender can begin sending a message only at the beginning of any of the time slots not in between

| **Pure Aloha** | **Slotted Aloha** |
|---|---|
| Any time anyone can transmit | The time is divided into fixed slots. |
| Vulnerable time= 2 X TT | Vulnerable time= TT |
| Efficiency = $G \times e^{-2G}$ | Efficiency = $G \times e^{-G}$ |

| | |
|---|---|
| Max efficiency=18.4% | Max efficiency=36.8% |

**Carrier-Sensor Multiple Access (CSMA):**

- Suppose there is bus topology, so if a node wants to transmit a message, it will check the point where its connected to the main bus and see if there is already any signal transmitting.
- If there is, then it will wait with the data and if there isn't, it will start transmitting
- It will only check that point on the channel it is connected to and not the entire channel
- 3 types:
  - 1-persistent:
    - Continuously keeps checking if the channel is free or not and as soon as it is free, it send the message
    - Worst case: If three nodes are waiting for a message to get completed, and they find out that the channel is free at the same moment, they will all send their messages together leading to collision
  - 0-persistent:
    - Checks if the channel is free after every set amount of time
    - Worst case: If suppose the node waiting has set a timer for 30 mins and the current ongoing signal finishes in 5 mins, then for 25 mins the channel is idle for no reason
  - p-persistent:
    - Continuously checks if channel is free or not
    - Once the channel is idle, it has p probability that it will start sending message immediately
    - Mixture of 0 and 1 -persistent

**Carrier-Sensor Multiple Access/Collision Detection (CSMA/CD):**

- used in ethernet
- No acknowledgement system
- If a node while transmitting, gets back another signal which is a collided signal, it gets to know that its signal actually collided
- If the node receives the collided signal after it stops transmitting, then it won't know if its signal actually collided
- So for this to work, TT>PD
- In worst case suppose, the node get collided at the absolute last moment before reaching the destination, then by the time the collided signal reaches the node, it has most probably stopped transmitting so won't be able to recognise the collided signal
- In worst case TT >= 2 X PD so that even if the node's message is collided at the last moment, the collided signal can reach the node before the node stops transmitting

  TT >= 2 X PD

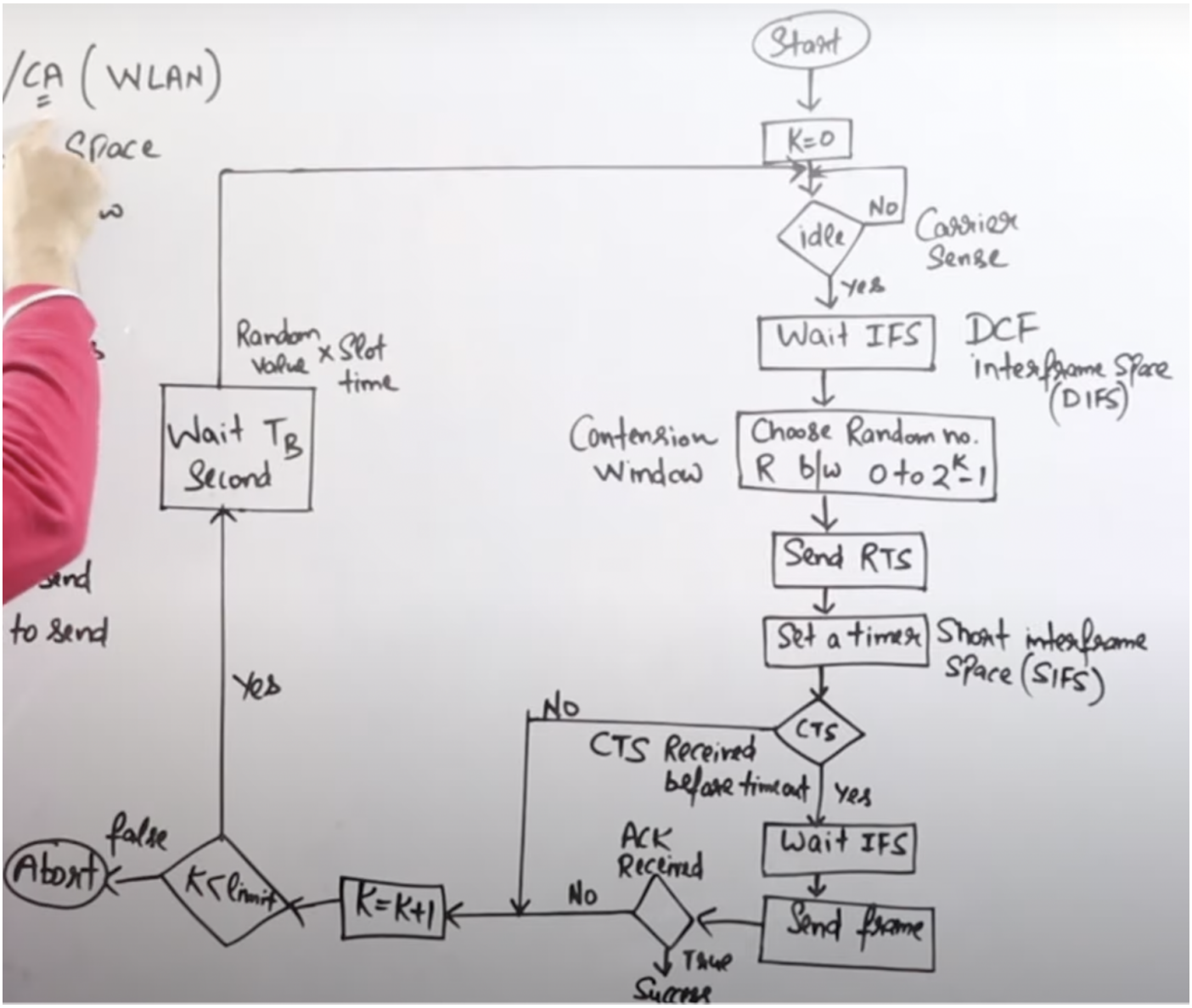
  => L/BW >= 2 X PD

  => L >= 2 X PD X BW

  where BW is bandwidth and L is length of message
- $\eta=1/(1+6.44a)$ where a = PD/TT

**Carrier-Sensor Multiple Access/Collision Avoidance (CSMA/CA):**

- used in wireless/wifi




k=no. of attempts

$T_B$= Back off time

IFS= inter frame space

RTS=ready to send

CTS=clear to send

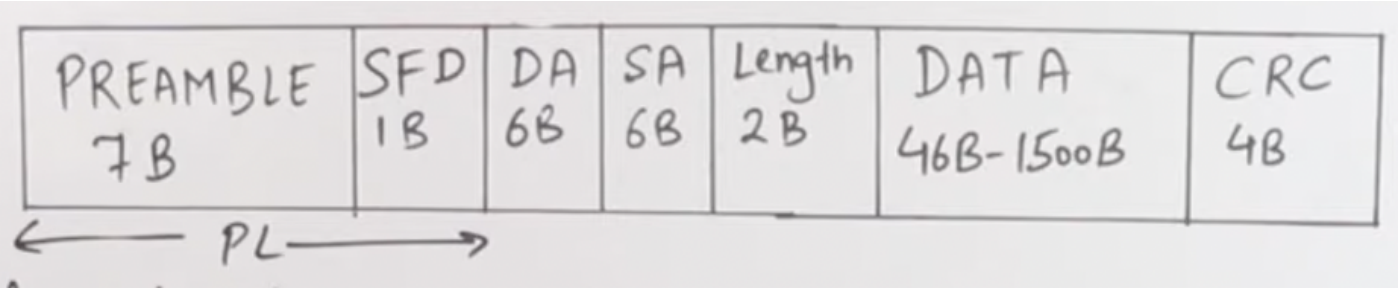
DIFS= DCF inter frame space = Distributed Coordinated function inter frame space

---

Ethernet Frame Format

- Given by IEEE 802.3
- Types of ethernet:
  - 10 Base 2 - Thin (10Mbits/sec - 200m)
  - 10 Base 5 - Thick
  - 10 Base T
  - 100 Base Fx - Fast
  - 10G base T - Gigabit
- Topology - Bus(most preferable), Star

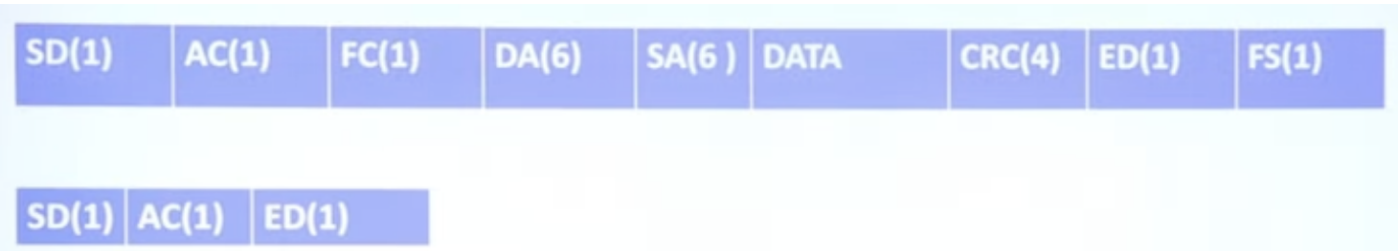
- 1Mbits/sec - 400Gbits/sec




- Preamble and SFD are added in Physical layer. They basically are for stations to get alert and synchronise
- Preamble=10101010.....10 (56 bits or 7 bytes)
- SFD=10101011 (8 bits or 1 byte) (Start of frame delimiter)
- Rest all are added in datalink layer.
- DA= Destination Address (6 bytes or 48 bits) (in case of wifi it is IP address of the destination, but in case of lan it is the MAC address of the next node/router, then the next node decides where to go next)
- SA=Source Address (6 bytes or 48 bits)
- Length= length of entire ethernet frame. This 16-bit/2 byte field can hold the length value between 0 to 65534, but length cannot be larger than 1500 because of some own limitations of Ethernet.
- Data= This is the place where actual data is inserted, also known as Payload. Both IP header and data will be inserted here if Internet Protocol is used over Ethernet. The data has to be between 46 bytes to 1500 bytes.
- CRC-=Cyclic Redundancy Check. 32- bits hash.If the checksum computed by destination is not the same as sent checksum value, data received is corrupted.
- Total size of ethernet frame excluding physical layer is 64 bytes to 1518 bytes.

**Token Ring (IEEE 802.5):**

- Ring topology is used
- Access control method used is token passing
- Token ring is unidirectional
- Data rate/bandwidth used is 4Mbps and 16Mbps
- Piggybacking acknowledgement is used (Send ack with data)
- Differential Manchester encoding is used
- Variable size framing
- Monitor station is used
- Frame format:




  - 2 frame formats - Data and Token
  - ******* **Will explain later** ********

**Network Layer Responsibilities:**

- host to host (Source to destination delivery/machine to machine delivery)
- uses logical address (IP): has two parts
  - network id
  - host id (machine)
- routing:
  - RIP
  - OSPF
- fragmentation
- congestion control

**Classful Addressing:**

- Before 1980s, IP address was 32 bits out of which 8 bits were reserved for the network id and 24 bits for host id. So in each network there could be $2^{24}$ hosts.
- But after 1980, the demand for IP increased a lot, so they came up with classful addressing
- In classful addressing, we divide the 32 bits into 4 octets(8 bits)

**Class A in IP addressing:**

- The first bit is always 0
- The first octet is the network ID and the rest are host ID
- No of networks = $2^7$-2= 128 -2=126
- Why -2? Because network id=00000000 and 01111111 are not given to any network
- No of hosts in each network =$2^{24}$-2
- Why -2? The first host address represents the network and the last host address is the direct broadcast address. Limited broadcast address for all classes is always 255.255.255.255
- If the first octet value is between **0-127**, then it is class A address
- Example: Let first ip address of google be 64.0.0.0 and last ip address is 64.255.255.255
- How to find network id from IP address:
  - Default mask of class A=255.0.0.0
  - Let IP address be 64.0.0.8
  - Do AND operation with mask and IP address
  - 64.0.0.8 = 01000000.00000000.00000000.00001000
  - 255.0.0.0=11111111.00000000.00000000.00000000
  - 64.0.0.8 AND 255.0.0.0 = 01000000.00000000.00000000.00000000 = 64.0.0.0
  - So network id is 64.0.0.0

**Class B in IP addressing:**

- The first two bits are always 10
- Range = **128 to 191** (10000000 to 10111111)
- No. of addresses = $2^{30}$
- The first two octets are the network ID and the next two are host ID
- No. of networks = $2^{14}$= 16384
- No. of hosts = $2^{16}$ -2 = 65536 -2 =65534

No. of hosts = $2^{16}$ -2 = 65536 -2 =65534

- Default mask = 255.255.0.0

**Class C in IP addressing:**

- The first three bits are always 110
- Range = **192 to 223** (11011111 to 11000000)
- No. of addresses = $2^{29}$
- The first three octets are the network ID and the last one is host ID
- No. of networks = $2^{21}$
- No. of hosts = $2^8$ -2 = 256 -2 = 254
- Default mask = 255.255.255.0

**Class D in IP addressing:**

- The first four bits are always 1110
- Range = **224 to 239** (11100000 to 11101111)
- No. of addresses = $2^{28}$
- Reserved for multicasting, group email and broadcast
- No networks and hosts

**Class E in IP addressing:**

- The first three bits are always 1111
- Range = **240 to 255** (11110000 to 11111111)
- No. of addresses = $2^{28}$
- Reserved for military service
- No networks and hosts

**Disadvantages of Classful Addressing:**

- Wastage of IP address:
  - Class D and E and reserved but not many are used
  - Class A has $2^{24}$ hosts. Even really big organizations don't need these many hosts, so wastage
  - It is not flexible. Imagine you need 1024 host addresses. Now Class C has 256 and Class B has 65K, so if we choose C, it is not enough and if we choose B it has a lot of wastage
- Maintenance is time consuming
- More prone to errors
- Security issues

**Classless Addressing/Classless Inter-Domain Routing:**

- No classes
- Only blocks
- Notation: x.y.z.w/n where n is mask/no of bits used to represent block/network.
- Mask means number of continuous 1s
- Example: 200.10.20.40/28 means there are 28 bits of block id. So out of the 32 bits IP address, 28 bits are for block id and host id has 4 bits. So total number of hosts=$2^4$ =16.
- Mask is 28 continuous 1s. So mask=11111111.11111111.11111111.11110000= 255.255.255.240

Mask is 28 continuous 1s. So mask=11111111.11111111.11111111.11110000=255.255.255.240

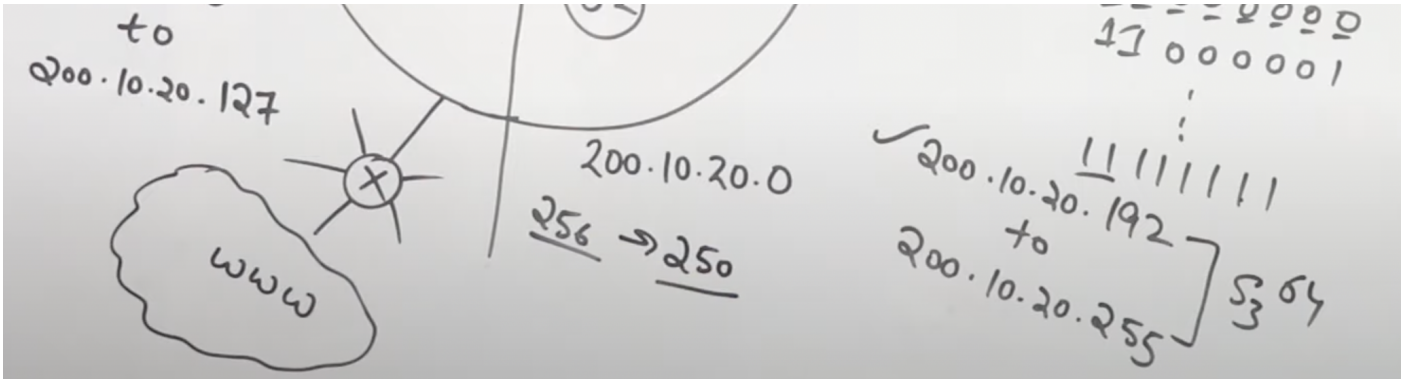
- To find network id, since mask is 28 bits. So host id is 4 bits. Now just convert the 4 bits to 0. And rest is same.
  - 200.10.20.40 = 200.10.20.00101000
  - Network id=200.10.20.00100000= 200.10.20.32/28
- Another way to find network id, mask AND IP address:
- Mask=255.255.255.240
- IP address=200.10.20.40
- AND = 200.10.20.32
- So network id=200.10.20.32/28
- Rules:
  - Address should be contiguous
  - No of addresses in a block must be in power of 2
  - First address of every block must be evenly divisible with size of block

**Subnetting:**

- Dividing the big network into small networks
- Advantage is ease in maintenance and improved security
- Suppose you're subnetting a class C network into 2:
  - Let the network address be 200.10.20.0 = 200.10.20.00000000
  - Now to divide it into two, we reserve the 1st bit.
  - So addresses in S1 will be from 200.10.20.**0**0000000(200.10.20.0) to 200.10.20.**0**1111111 (200.10.20.127)
  - And addresses in S2 will be from 200.10.20.10000000 (200.10.20.128) to 200.10.20.11111111 (200.10.20.255)
  - Subnet ID of S1 = 200.10.20.0
  - Broadcast ID of S1 = 200.10.20.127
  - Subnet ID of S2 =200.10.20.128
  - Broadcast ID of S2 = 200.10.20.255
  - So total usable addresses are 256 -2 -2 = 252, while without subnetting it was 256-2=254
  - Subnet mask=255.255.255.10000000=255.255.255.128

**Variable Length Subnet Masking (VLSM):**

- Suppose you want to divide network into three parts - 50%,25%,25%
- Total addresses=126+62+62=256-2-2-2==256-3(2)=250




**Subnetting in Classless Inter domain Routing (Subnetting in CIDR)**


In this case we fixed one extra digit during subnetting, so that's why we changed /26 to /27

**Important question!**

YouTube Video

Lec-51: Numerical Question on CIDR | Classless ...