Formulae

FRL Collision Domain

The maximum amount of collisions that can occur. Number of devices = `n`

- Cable: n
- Repeater: n

FRL Circuit Switching

- Setup Time, Tear Down time
- Transmission Time: Number of Messages / Bandwidth
- Propagation Delay: Distance / Velocity
- Total Time: Setup Time + Transmission Time + Propagation Delay + Tear Down Time

FRL Packet Switching

- `n` : Number of intermediate devices re-transmitting the data packet
- Total time: n(Transmission Time) + Propagation Delay

FRL Data Link Error Detection

> Window Size: Sender | Receiver

- Stop & Wait: 1 | 1
- Go-Back-N: $N = 2^k - 1$ | 1
- Selective Repeat: $N = 2^{k-1}$ | $N = 2^{k-1}$
  > If we want to transmit 4 bits, $N = 4 = 2^m = 2^2$, and window size for the

sender will be $2^2 - 1 = 3$ bits.

- Efficiency:
  - T_p: Propagation Delay, T_t: Transmission Time
  - Efficiency: Transmission Time / Round Trip Time (= 2* Propagation Delay)
  - $x = T_p/T_t$
  - Stop-and-Wait ARQ: $1 * 1/(1 + 2x)$
  - Go-Back-N: $(2^k - 1) * 1/(1 + 2x)$
  - Selective Repeat: $(2^{k-1}) * 1/(1 + 2x)$

FRL Error Control

- **Pure ALOHA**:
  - **Transmission Time** = Message Size / Bandwidth.
  - **Vulnerable Time**: 2*Transmission Time
  - **Efficiency**: $G * e^{-2G}$. Maximum Efficiency: $G = 1/2$, $1/2 * e^{-2*(1/2)} = 1/2 * e^{-1} = 18.4\%$
    - G: Number of stations currently transmitting data.
- **Slotted ALOHA**:
  - **Vulnerable Time**: Transmission Time
  - **Efficiency**: $G * e^{-G}$. Maximum Efficiency: $G =, 1 * e^{-1} = 36.8\%$
    - G: Number of stations currently transmitting data.
- **Carrier Sense Multiple Access** / **Collision Detection (CSMA/CD)**:
  - **Transmission Time** = Message Size / Bandwidth.
  - Transmission Time $\geq$ 2*Propagation Delay
  - Message Size $\geq$ 2 * Propagation Delay * Bandwidth
  - **Propagation Delay**: Distance / Velocity
  - **Efficiency** = $1 \div (1 + 6.44a)$, a = Propagation Delay / Transmission Time

Basics

Network Types

- Based on Distance, the network types are:
    - PAN: Personal Area Network
    - LAN: Local Area Network
    - CAN: Campus Area Network
    - MAN: Metropolitan Area Network
    - WAN: Wide Area Network
- As distance increases, the number of devices used also increases, and we also have to put repeaters or network extenders to ensure the signal is strong enough to reach the destination.
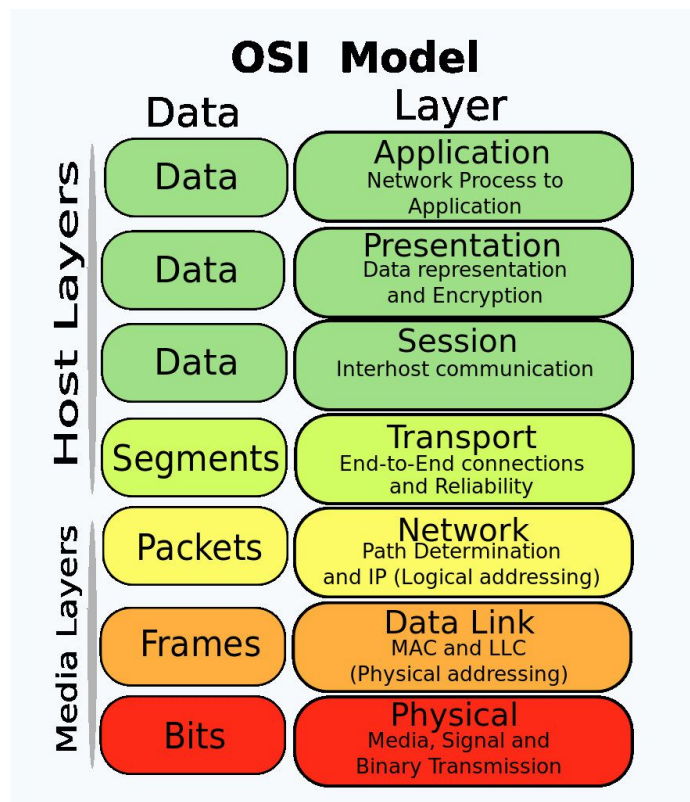- As distance increases, the error rates & maintainence cost also increases.

OSI Model

Image taken from here

Why?

- There are a lot of functionalities and protocols that are necessary to send data from the client to the server, or vice versa.
- OSI is a theoritical model, ie it is not implemented in real life. But, using this model,

all the other models are developed.

- OSI Model defines the layers in a network system, how the protocols are implemented in each layer, and how they work together to achieve it's primary purpose of moving data from Point A to Point B.

Layers

- Application
- Presentation
- Session
- Transport
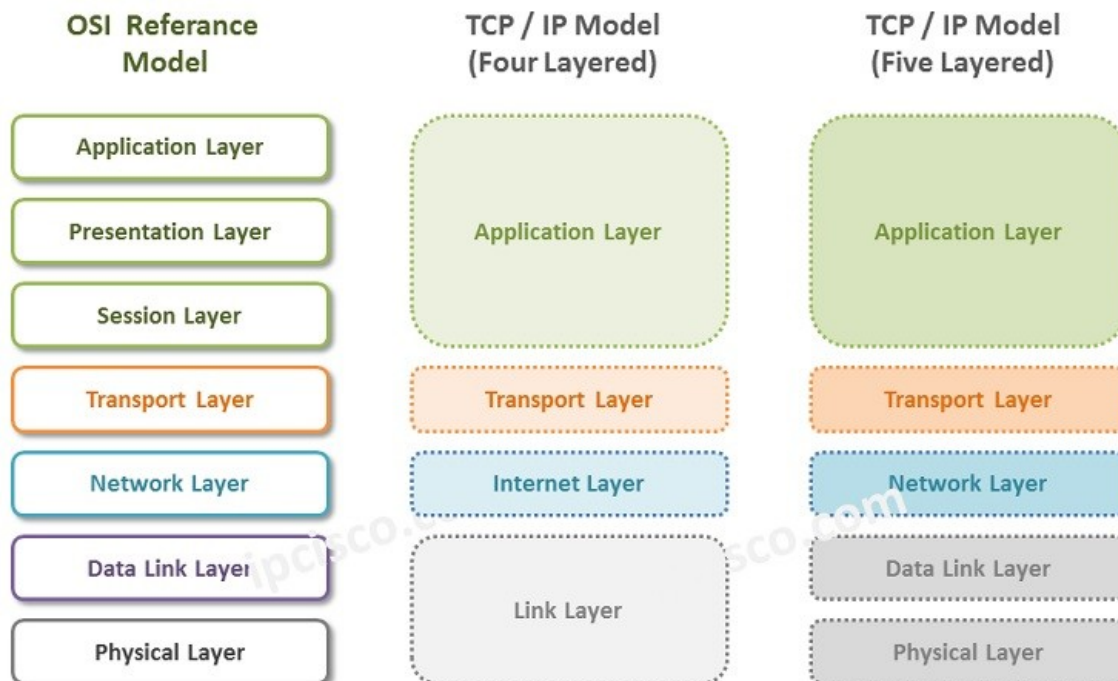- Network
- Data Link
- Physical

Difference vs TCP/IP Model

Image taken from here

- TCP/IP layer is an implementable model.
- It was a protocol defined by ARPANET for use by them.
- It combines the `Application`, `Presentation` & `Session layer` into the

`Application Layer`.

- The `Network Layer` is defined as the `Internet Layer`.
- It combines the `Data Link Layer & Physical Layer` into `Network Access Layer`.

Physical Layer

- Deals with hardware.
- Isn't concerned with security, that part is handled by the software.
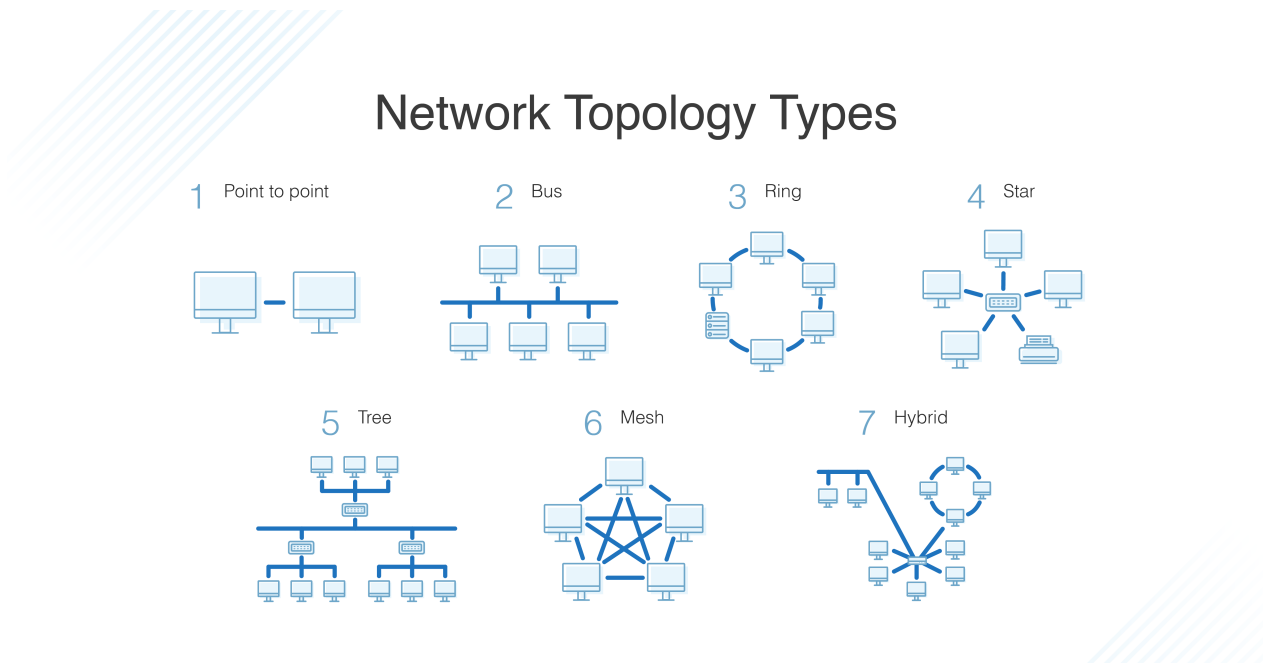
Network Topologies

Image taken from here

Point-to-Point Topology

- Direct communication between two devices.
- Simple and easy to implement.
- Suitable for small networks.
- Efficient and reliable communication.

Mesh Topology

- Every device is connected to every other device.
- Redundant paths enhance reliability.
- High fault tolerance but expensive and complex.
- Common in critical applications where reliability is crucial.

Star Topology

- All devices are connected to a central hub or switch.
- Easy to install and manage.
- Failure in one device doesn't affect others.
- Centralized control and potential single point of failure.

Bus Topology

- Single central cable to which all devices are connected.
- Simple and inexpensive.
- Limited cable length and potential for collisions.
- Suitable for small networks with low traffic.

Ring Topology

- Devices connected in a circular fashion.
- Each device has exactly two neighbors for communication purposes.
- Unidirectional or bidirectional communication.
- Potential for data collisions and a failure in one device can disrupt the entire network.

↓ Parameter | Topology →	Mesh	Star	Bus	Ring
Number of Cables	$(n(n-1))/2$	n	$n+1$	n
Number of ports	$n(n-1)$	n	n	$2n$
Reliability	high	low	low	low
Affected by noise?	no	no	yes	yes
Cost	high	low	low	low
Security	high	medium	low	low

↓ Parameter | Topology →	Mesh	Star	Bus	Ring
P2P between devices	yes	no	no	yes
Multipoint	no	no	yes	no

- Reliability = **low** means there is a single point of failure.
- #reason-hub: The hub broadcasts messages to all devices by default.
- #reason-cable: The cable cannot filter or send data to only a specific device.

Manchester Encoding

- It is used to encode 1's and 0's.

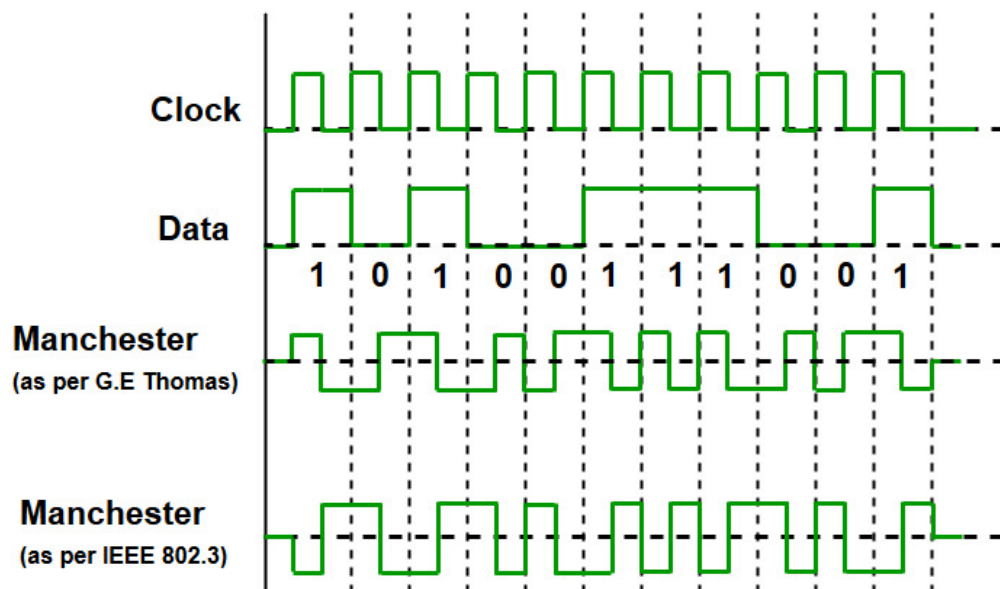- There are 2 variants: The one by G E Thomas, and the IEEE 802.3 version. By default, IEEE 802.3 is used.

Image taken from [here](https://media.geeksforgeeks.org/wp-content/uploads/ETHERNET_1.jpg)

Networking Devices

- Hardware only: Cables, Repeaters, Hubs
- Hardware + Software: Router, Gateway, Firewall, Modem
- Software only: IDS

Cable

- Layer: `Physical`
- Types:

  - Baseband: `Only 1` signal can be transmitted over the cable at any point of time.
  - Broadband: `>1` signals can be transmitted over the cable in parallel.
- Signal filtering/attenuation: Not possible, since it's purely hardware.
- Collision: A maximum of `n` collisions can occur, if there are `n` connected devices.
- **100BaseT**:

  - `100` : speed in Mbps
  - `Base` : Type of cable, Baseband / Broadband
  - `T` / `2` / `5` / `Fx` : 100/200/500 metres or Fiber Channel, range about 2km. The signal quality deteriorates beyond the range.

Repeater

- Layer: `Physical`
- It is a 2 port device.
- While an amplifier amplifies the signal (from x to 2x,3x, etc), a repeater regenerates the signal (x) and retains it's original strength.
- Signal filtering/attenuation: Not possible, since it's purely hardware.
- **Collision Domain**: A maximum of `n` collisions can occur, if there are `n` connected devices.
- **Broadcast Domain**: Any broadcast packet will reach all devices connected, on all sides of the repeater.

Hub

- Layer: `Physical`
- A hub is a multi-port repeater.
- It has > 2 ports.

- There is some basic functionality built-in, using which we can check if all devices are properly connected or not.
- Signal filtering/attenuation: Not possible. If `A` wants to send some data to `B`, then the others also get the message. The message is broadcasted to all connected devices.
- **Collision Domain**: A maximum of `n` collisions can occur, if there are `n` connected devices.
- **Broadcast Domain**: Any broadcast packet will reach all devices connected, on all sides of the hub.

Bridge

- Layer: `Data Link`
- Bridges are used to connect 2 different LANs.
- A packet transmitted by a device contains the source & destination MAC address.
- Signal filtering/attenuation: Possible. If the bridge notices that some packet does not need to be forwarded to the other LAN to reach it's destination, it can filter the packet & directly send it to the destination, on the same LAN it originated in.
- 2 Types: Static & Dynamic
  - **Static**: The MAC to Port mapping table is static. The network administrator sets it up when setting up the device.

    ```
    - Cannot dynamically map MAC Address -> Ports.
    - If the device MAC Address changes for some reason, the mapping wi
    ```

  - **Dynamic**: The MAC to Port mapping table is dynamic. It learns about the mapping by itself, without the interference of the network administrator.

    ```
    + Learns and maps MAC -> Port by itself.
    - Broadcasts the message to all devices, if it doesn't know the des
    - Takes some time to learn about the mappings, at first.
    ```

    - Initially, it is empty.
    - When the switch cannot find the destination in the MAC to Port mapping table, it broadcasts the packet to all devices.
    - The device which has accepted it, sends back the ACK (Acknowledgement) packet.

- Using that packet, the bridge 'learns' about the location of the destination device and adds it to the mapping table.
- It uses Spanning Trees in the form of `Bridge Data Unit Protocol` , which ensures that data packets don't get stuck in a loop.
- **Collision Domain**: The bridge uses the `Store & Forward` strategy. It stores the incoming packet, processes it, and then sends it to the destination. There may be collisions within the networks themselves, but not across them.
- **Broadcast Domain**: Any broadcast packet will reach all devices connected, on all sides of the bridge.

Switch

- Layer: `Data Link`
- It is a multi-port bridge.
- We connect multiple devices with a switch, which in turn is connected to the router, which in turn is connected to the Internet (optional).
- Switch provides full-duplex links to the devices.

  > Duplex: Data can be sent/Received parallely.
  > Simplex: Only 1 signal can be transmitted at a time.

- **Collision Domain**: The switch uses the `Store & Forward` strategy. It stores the incoming packet, processes it, and then sends it to the destination. There may be collisions within the networks themselves, but not across them.
- **Broadcast Domain**: Any broadcast packet will reach all devices connected, on all sides of the switch.

Router

- Layer: `Physical  Data Link  Network`
- It is a multi-port device. Multiple networks can be connected to it at the same time.
- Router can check for both MAC & IP addresses.
- The router receives an IP Address (usually `xxx.xxx.xxx.1`) from each network it is connected to, so that it can identify and differentiate between them.
- Forwarding/Flooding: Once the router receives a data packet containing the source & destination IP & MAC Address, it checks it's **Routing Table**.
  - If it finds the entry, it forwards it to the destination.

- ○ If it does not find the entry ie cannot decide where to send the packet, it floods the network ie broadcasts it to all networks.
- Filtering: If the router receives something like an ARP request, it can send it back to the same network.

> An ARP (Address Resolution Protocol) request is a message sent by a device on a network to discover the MAC (Media Access Control) address associated with a specific IP (Internet Protocol) address.

- **Collision Domain**: The router uses the `Store & Forward` strategy. It stores the incoming packet, processes it, and then sends it to the destination. There may be collisions within the networks themselves, but not across them.
- **Broadcast Domain**: Any broadcast packet will reach all devices connected but only within the network. It will not propagate to the other networks connected to the router.

# Collision Domain & Broadcast Domain

- **Collision Domain**: A collision domain represents the network segment where collisions can occur. If multiple devices send data packets at the same time in a shared medium, the data packets will collide with each other. In the worst case, the collision domain can be n if there are n devices connected.
- **Broadcast Domain**: Whether all devices receive the data packet a particular device on the network sends, regardless of if it was intended for them or not.

Device	Collision Domain	Broadcast Domain
Repeater	Unchanged	Unchanged
Hub	Unchanged	Unchanged
Bridge	Reduced	Unchanged
Switch	Reduced	Unchanged
Router	Reduced	Reduced

Switching Techniques

Circuit Switching

- Layer: `Physical`
- Formulae: [here]
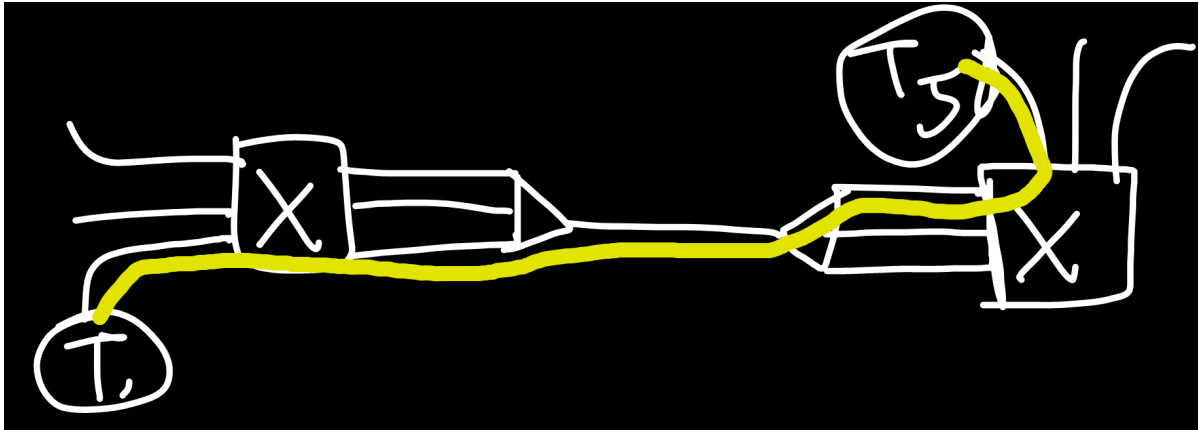
- In a Telephone Exchange, the physical cables are already laid out between all devices. The Exchange themselves are connected to each other using cables.
- When we pick up the phone and dial a number, a connection is established between the sender and the receiver. The time it takes is acalled `Setup Time`.
- By establishing a connection, we're effectively reserving a portion of bandwidth and ports, for the duration of the session.
- Once the connection is established, data flows in-order and is continuous. No headers are needed since a one-to-one connection is established.

  > Header: Section within a data packet which contains source and destination address.

- Efficiency of the overall network is less, since once a connection is established, resources are reserved for it. Even if it's not being used, the resources can't be freed until the connection is terminated.
- Packet delay is minimal since the pathway is already clearly defined at the very beginning.
- Circuit Switching is good for Telephone networks, but not suitable for computer networks because of it's inefficiency.

Message Switching

- Predessor of [Packet Switching]
- Uses the Store & Forward strategy.

- Hop by hop delivery: A message first goes to the first node. It processes it, then transfers it to the next node, and so on. The message keeps hopping between various nodes, till it reaches it's destination.
- In contrast to Packet Switching, here the message is not broken down before being transmitted. It is transmitted as a whole.

Packet Switching

- Layer:
    - Datagram: `Network`
    - Virtual Circuit: `Data Link`
- Formulae: here
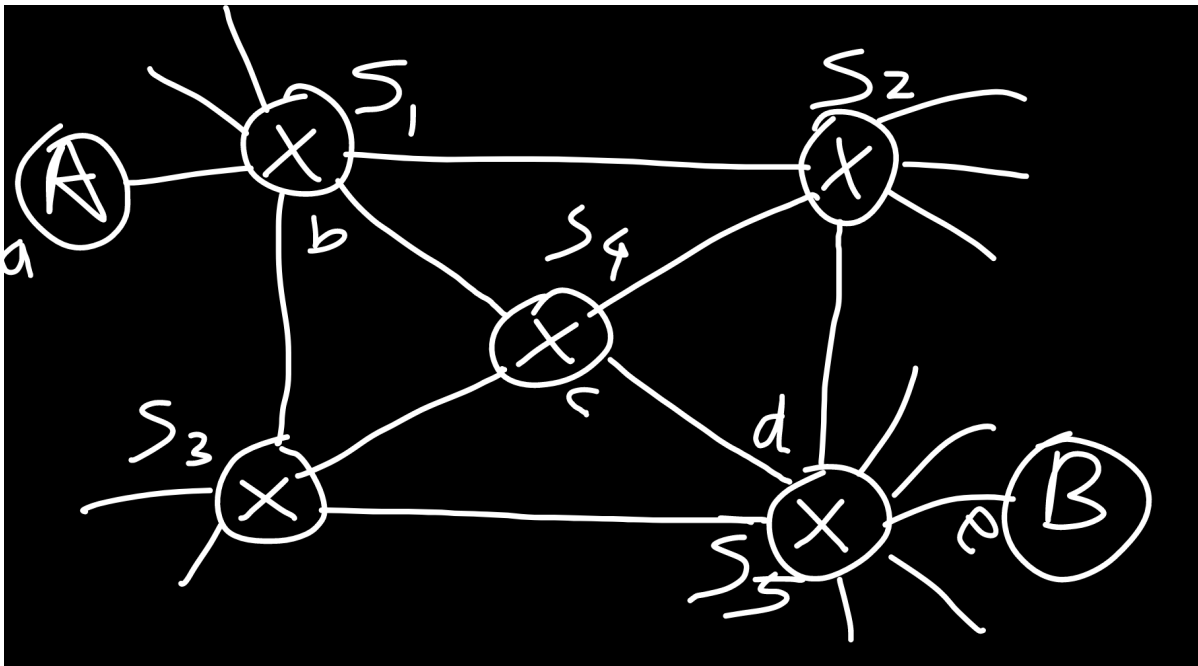
- All connected switches store the received data packet, figure out which path to send it through (using Routing Tables), then send it. This is called **Store & Forward strategy**.
- If there are multiple data packets to be sent, they are sent one after the other. The sender does not wait for the current packet to be received on the other end, before sending the next packet.
- Efficiency: high, since no reserved pathways exist.
- Delay: The more the number of intermediate devices like switches, the more the delay.

Datagram Switching

- Layer: `Network`
- Connectionless: No connection is established before tranmitting data. Resources are provided on-demand.
- No reservation takes place.
- Data is usually received out-of-order, since they're taking different paths to get to the destination.
- There is high overhead, all data packets have headers in them.
- Packet loss can take place.
- Packet Delay depends on the load on the network.
- Used in Internet

Virtual Circuit

Layer: `Data Link`

- Connection-oriented: A global packet is sent first, which establishes a dedicated connection and resources in it's pathway.
- The pathway is reserved.
- Data packets are received in-order.
- Only the global (first) packet has a header.
- Since the connection is dedicated, there is minimal delay.
- Minimal probability of packet loss.
- Used in ATM (Asynchronous Transfer Mode) etc.

Cast

- The meaning of cast is to share information with a number of devices.
- **Unicast**: `one-to-one`. We're sending information from one device to another.
- **Broadcast**: The information is sent to a broad range of devices (for example, all devices within a local area network).
  - Example: Let's assume: Network `A: 90.0.0.1`, `B: 92.0.0.1`.
  - **Limited Broadcast**: A broadcast originating from network `A`, to all devices on `A`. The signal is not received by devices on network `B`. Broadcast address:

```
            255.255.255.255
```
  - ○ **Direct Broadcast**: A broadcast originating from network `A`, to all devices on
    `B`. The signal is not received by devices on network `A`. Broadcast address, in
    this case: `92.255.255.255`.
- **Multicast**: Broadcasting to a specific group of devices.

Data Link Layer

- Within a particular network, this layer is enough to send data between devices.
- Terminology: `Data Frame`. Data packets, received from the Network Layer, are split
  into equal sized Data Frames, and transmitted to Physical Layer.
- **Node-to-Node delivery**: The primary responsibility is to determine which node the
  data packet will hop to, next. At every step, it performs the task of moving the data
  packet to the next node, till it reaches the destination.
- **Flow Control**: Flow control refers to the techniques used to regulate the flow of data
  between two communicating devices to ensure that the sender does not overwhelm
  the receiver. It involves mechanisms for pacing data transmission, buffering, and
  acknowledgment to prevent data loss due to congestion or buffer overflow.

  > Methods: Stop & Wait, Go Back N, Selective Repeat

- **Error Control**: Error control involves techniques used to detect and correct errors
  that may occur during data transmission over a communication channel.

  > Cyclic Redundancy Check, Checksum (in Transport Layer), Parity (in Transport
  > Layer)

- **Access Control**: Access control mechanisms govern the access to the
  communication channel by multiple devices in a network. It manages how devices
  contend for access to the channel, resolve contention issues, and allocate the
  channel efficiently to ensure fair and orderly data transmission.

  > Methods: CSMA/CD (dontention-based), ALOHA, Slotted ALOHA, Token Ring
  > (contention-free)

- **Physical Addressing**: It uses MAC Addresses for identifying devices.

Stuffing

- Stuffing is done on the Data Link Layer to distinguish between 2 data frames.

- General: FLAG **ABCDE** FLAG **BCDEF** FLAG …
  - `FLAG` is used as a delimiter, to separate 2 frames.
- **Byte Stuffing**: In byte stuffing, a special byte sequence is inserted into the data stream to mark the beginning and end of a frame or packet. When the sender encounters the special byte sequence within the data, it inserts an additional byte to distinguish it from the data. Byte stuffing helps to prevent ambiguity and misinterpretation of data frames, particularly when the data contains sequences that might be mistaken for control characters.
  - If `FLAG` itself is present in the data: FLAG **ABCDE** ESC **FLAG BCDEF** FLAG …
    - If `ESC` precedes the `FLAG`, it is meant to be ignored, as it is a part of the data.
  - If `ESC` itself is present in the data: FLAG **ABCDE** ESC **ESC BCDEF** FLAG …
    - If `ESC` precedes the `ESC`, it is meant to be ignored, as it is a part of the data.
- **Bit Stuffing**: Bit stuffing is similar to byte stuffing but operates at the bit level. In bit stuffing, a special bit pattern is inserted into the data stream to mark the beginning and end of a frame. When the sender encounters the special bit pattern within the data, it inserts an additional bit to ensure that the pattern does not appear in the data itself. Bit stuffing is commonly used in synchronous communication protocols, such as HDLC (High-Level Data Link Control) and Ethernet, to maintain synchronization and prevent errors in data transmission.
  - `1` after `011111` (1 followed by 0 followed by 5 1's) is a starting delimiter, used in the header. If it is a part of the data, we stuff a 0 after `011111`.
    - Example: 011111**0**11010110101110011111**0**101010010
- Data is unstuffed before being sent to the next layer.

Error Detection and Correction

- In the `Data Link Layer`, the data is checked bit-by-bit for errors.
- Types of errors:
  - **Single Bit**: Error only occurs in 1 bit. Example: 101**0** ➡ 101**1**
  - **Burst**: Error in more than 1 bit. Example: 1**0**101**0** ➡ 1**1**101**1**
- **Length of error**: The number of bits, between the 1st and last error bit. In the previous example, it is 5 bits.

Error Control

- Formulae: [here](#)

Stop and Wait ARQ

- Formulae: [here](#)
- ARQ: Automatic Repeat Request
- Acknowledgement (ACK): Independent, per frame
- Maximum frames Re-tramsmitted: 1
- It is an error control technique. It is done by keeping a copy of the sent frame, and re-transmitting it when the timer expires.
- **Window size:** Sender: 1 | Receiver: 1
- Sequence numbers used: 0,1.
- Steps:
    i. `Frame 0` is sent by `A` to `B`.
    ii. `B` receives the frame, and sends back the acknowledgement `ACK 1`. The acknowledgement states that `B` is now expecting `Frame 1`.
    iii. `A` receives the acknowledgement, sends `Frame 1`.
    - If `Frame 0` is lost or the acknowledgement is lost, then `A` sends `Frame 0` again to `B`.
        - If the frame was lost, `B` accepts it.
        - If the frame was received but acknowledgement was lost, `B` discards it as a duplicate frame.
    - `A` waits for a specific amount of time (called timeout) before re-sending the frame.
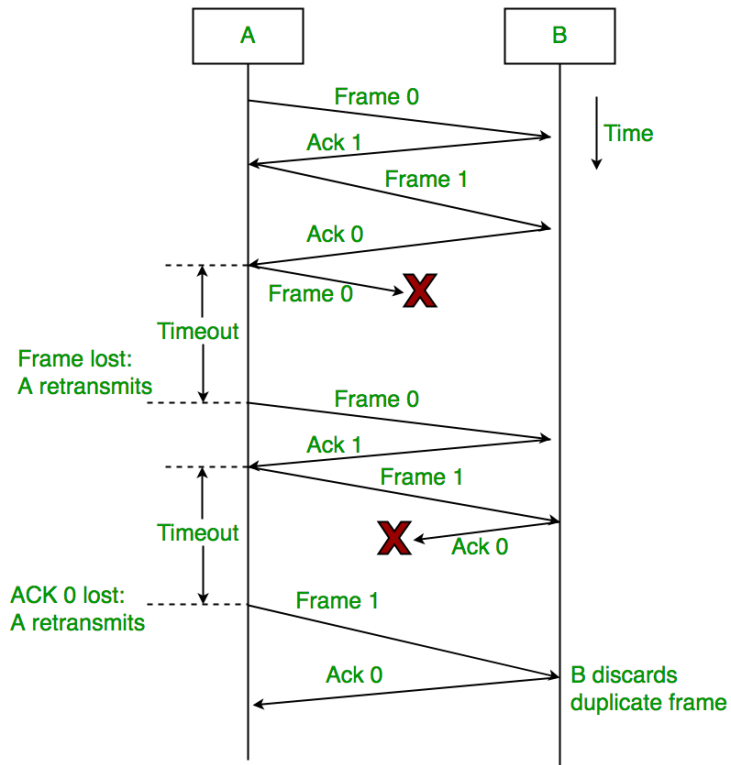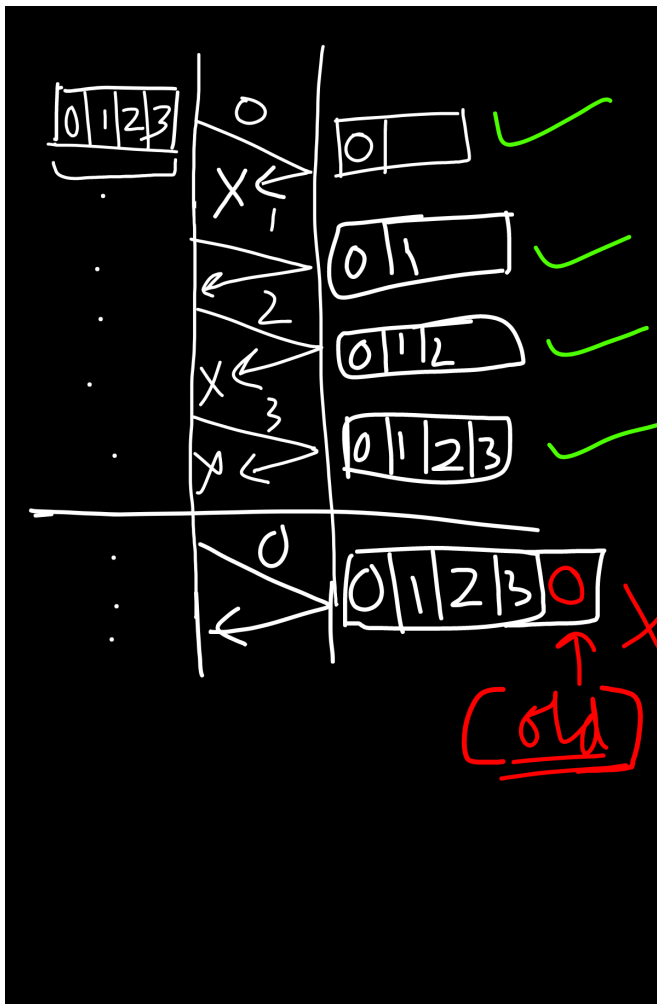
Image taken from

Go-Back-N ARQ

- Formulae: here

- ARQ: Automatic Repeat Request

- Acknowledgement (ACK): Cumulative

- Number of bits used to represent window: k

- Maximum frames Re-tramsmitted: $2^k - 1$

- **Window size:** Sender: $N = 2^k - 1$ | Receiver: 1

- If we want to send N bits, then size must be 1 less than it, otherwise it will accept the previously received packet again.

- If window size of the sender is `N`, it means `N` packets can be sent to the receiver. If any acknowledgement in the middle is lost, and still the receiver expects the correct packet, it is assumed that all frames have been received properly.
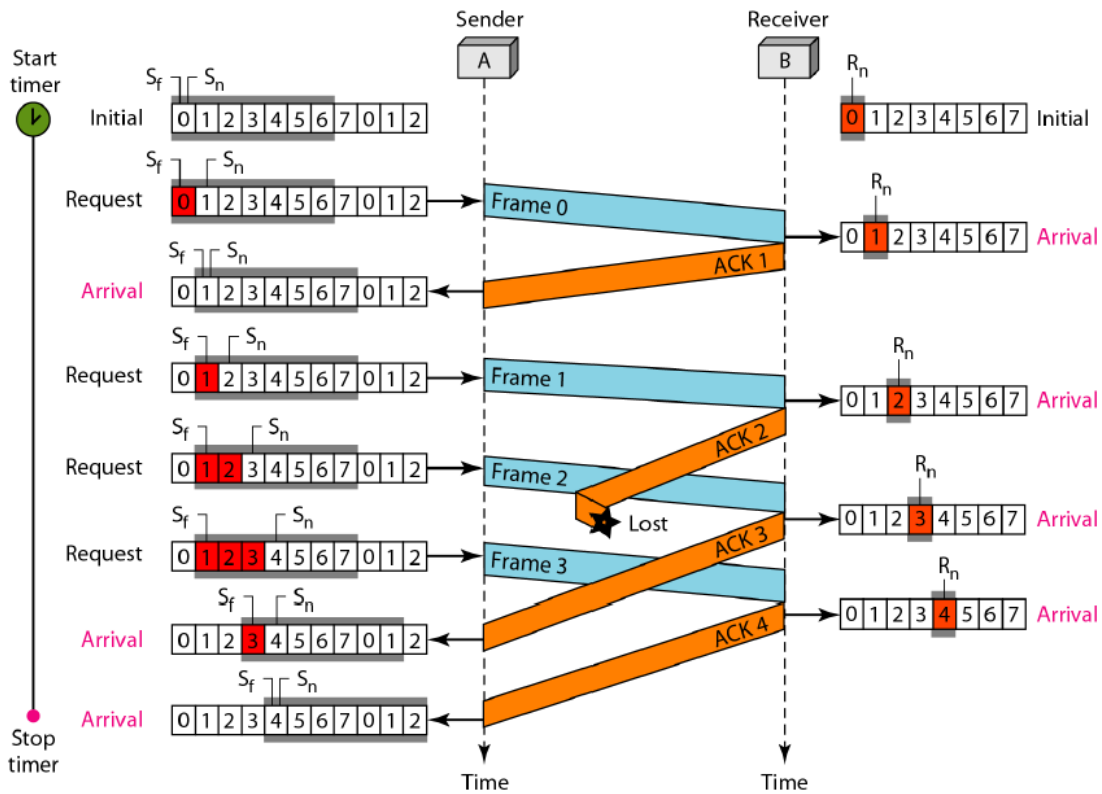
Image taken from here

- If sender window size is $N = 2$, and frames 0,1,2,3 are sent, and if `Frame 2` is lost in the middle, then Frame 3 onwards will not be accepted by the receiver (NAK 2). Afterwards, Frames 2 & 3 will have to be re-sent by the sender. So, **Frames cannot be received out of order.**
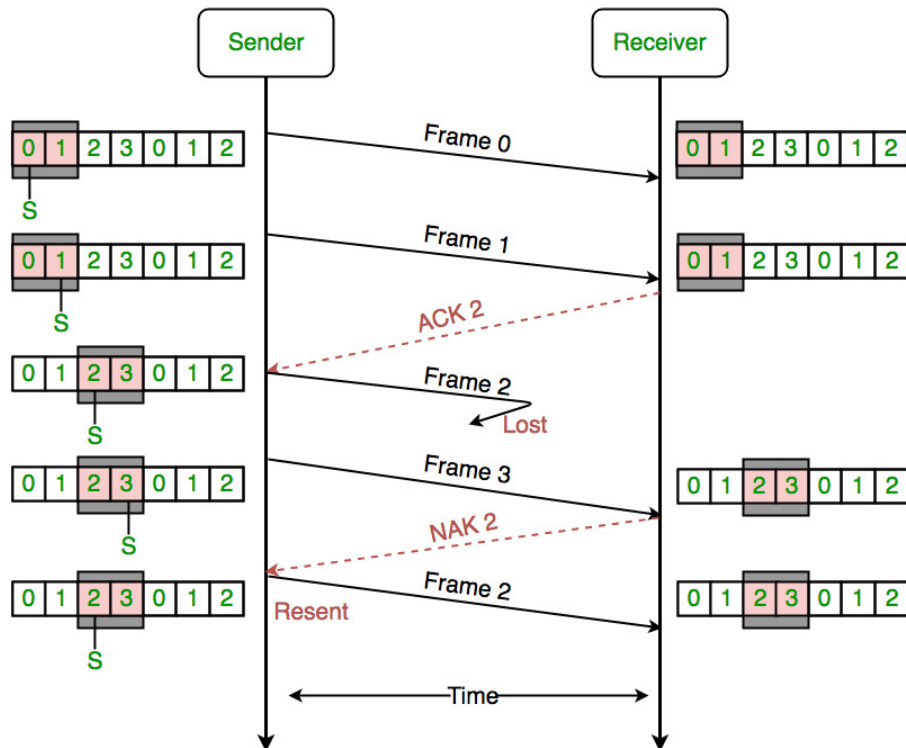
Image taken from here

Selective Repeat ARQ

- Formulae: here
- ARQ: Automatic Repeat Request
- Acknowledgement (ACK): Cumulative & Independent
- Number of bits used to represent window: k
- Maximum frames Re-tramsmitted: 2^{k-1}
- **Window Size:** Sender: $N = 2^{k-1}$ | Receiver: $N = 2^{k-1}$
- Since the receiver can receive any one of the $2^{3-1} = 4$ frames that the sender sends, ie frames 0,1,2 or 3 out of 0,1,2,3,4,5,6,7 if window size for both is $2^{3-1} = 4$, **frames can be received out of order**.
- Window size of receiver should not be greater than 2^{m-1}, otherwise it will accept the previously received packet again.
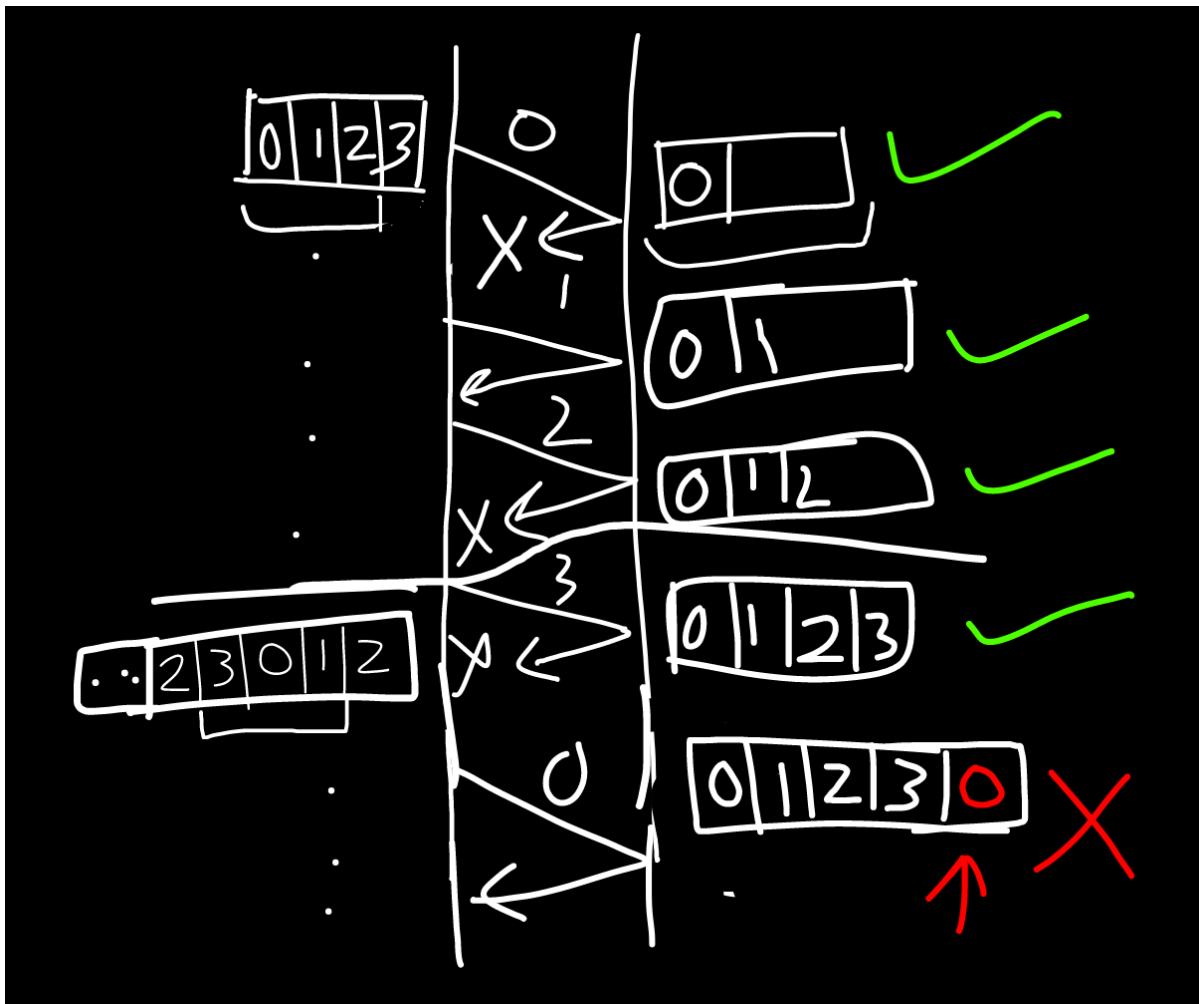
- Sliding Window will not move till all frames have been received. For example, out of frame 0,1,2,3, if 1 was lost in the way, then the receiver will accept 0,2,3 but will not send an ACK for 1. Then, sender will re-send frame 1.
- NAK: If a frame is corrupt, it can send an NAK (negative acknowledgement) for it, so that the sender re-sends it.

Image taken from here

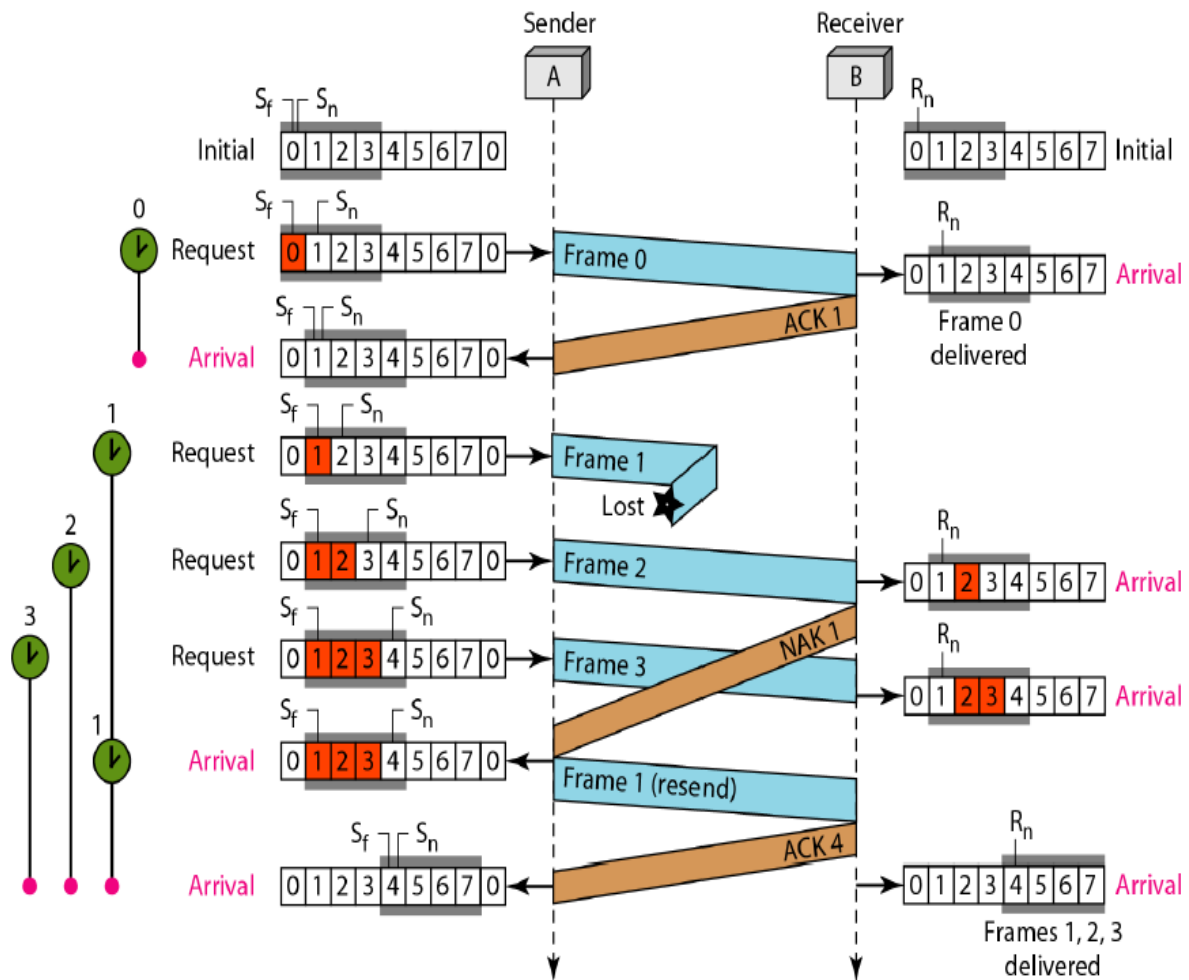# Error Detection

Single Bit Parity

- Least Expensive
- Number of bits added: 1
- **It can only detect errors when the number of error bits is odd..**
- 2 types:
  - Even Parity: Number of 1's in the data word should be even.
    - Example: 1010 -> 1010**0**, 1110 -> 1110**1**

  - Odd Parity: Number of 1's in the data word should be odd.
    - Example: 1010 -> 1010**1**, 1110 -> 1110**0**

- If there is some error in the data, for example, in case of even parity, if 1**0**100 becomes 1**1**100, then the number of 1's is no longer even like it was supposed to be. Also, if **11**10**1** becomes **00**11**1**, then also it can be detected.
- **Hamming Distance**: Perform a XOR operation, then count the number of 1's. Example: $1010 \oplus 1100 = 0110 : 2, 0000 \oplus 1111 = 1111 : 4$.
  - If the Minimum Hamming Distance is `d` , then a maximum of `d-1` errors can be detected.
  - Example 0:

0	1	2	3	Even Parity
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

  - Here, the Minimum Hamming Distance (example between 00000 & 00011) is 2, so only 1-bit errors can be detected.

Cyclic Redundancy Check (CRC)

- Based on Binary division
- Number of Redundant bits: r
- It can detect single bit, odd number of errors, and burst error of length equal to the maximum polynomial degree.
- How it works is explained using an example.
- Example 0:
    i. Data Word: 1010101010
    ii. Divisor, **tells us how many 0 bits to add at the end**:
        - If given in polynomial: $x^4 + x^3 + 1 = 1.x^4 + 1.x^3 + 0.x^2 + 0.x^1 + 1.x^0$ ie 11001, 1 if x has a degree, 0 in all the other places. Number of digits to append = degree of highest polynomial = 4.
        - If given in binary, 11001, then Number of digits to append = 1 less than the length of the Divisor = 4
    iii. Divide: 10101010100**0000** $\div 11011$
        - $10101 \oplus 11001 = 01100$
        - $01100\ 0 \oplus 11001 = 00001$ Start with the leading 1, add the same number of bits as needed to make it equal to 11001
        - $00001\ 1010 \oplus 11001 = 00011$
        - $00011\ 000 \oplus 11001 = 000010$, Remainder$= 000010$.
    iv. Append 4 bits from the Remainder, starting from the last. So, add 0010 to 1010101010 to make it 10101010100010.
    v. When we send these 14 bits to the receiver, the receiver divides this same number with the same Divisor.
    vi. Receiver receives: 1010101010 0010
    vii. Divide: 10101010100010 $\div 11011$
        - $10101 \oplus 11001 = 01100$
        - $01100\ 0 \oplus 11001 = 00001$
        - $00001\ 1010 \oplus 11001 = 00011$
        - $00011\ 001 \oplus 11001 = 000000$
    viii. **The Remainder will always be 0 for the receiver. But, if there's an error, it will be a non-zero value.**
    ix. Efficiency: $10/14 * 100 = 71.42\%$

Hamming Code

1. Data-word:

   - Parity bits are spaced out according to 2^n, ie they're located at positions $2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8$, ie $1, 2, 4, 8$ etc.

   - 7 bits:

7	6	5	4	3	2	1
D_3	D_2	D_1	P_2	D_0	P_1	P_0

   - 11 bits:

11	10	9	8	7	6	5	4	3	2	1
D_6	D_5	D_4	P_3	D_3	D_2	D_1	P_2	D_0	P_1	P_0

   - **Example**: Let data word (7 bits) be 1010:

7	6	5	4	3	2	1
1	0	1	P_2	0	P_1	P_0

2. Calculating parity bits:

   - For P_0:
     - $2^0 = 1$. Start from 1. Take 1 bit, leave the next 1 bit, take the next 1 bit, etc.
     - Take positions $1, 3, 5, 7$, etc.
     - $P_0 = 0 \oplus 1 \oplus 1 = 0$
   - For P_1:
     - $2^1 = 2$. Start from 2. Take 2 bits, leave the next 2 bits, take the next 2 bits, etc.
     - Take positions $2, 3, 6, 7, 10, 11$, etc.
     - $P_1 = 0 \oplus 0 \oplus 1 = 1$
   - For P_2:
     - $2^2 = 4$. Start from 4. Take 4 bits, leave the next 4 bits, take the next 4 bits, etc.
     - Take positions $4, 5, 6, 7, 12, 13, 14, 15$, etc.
     - $P_2 = 1 \oplus 0 \oplus 1 = 0$

3. When the receiver receives it, it calculates the parity bits too. If they match with that of the sender, there is no error.
4. **Example of error**:
    - Sent data: 10011100101
    - Received data: 10111100101
    i. For sender, $P_0 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 1$
    ii. For receiver, $P_0 = 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 0$
    - Clearly, the parity bit is different, so the error is detected.

Error Control

MAC: Multiple Access Control

- It is only used in networks where a single link is shared by multiple devices, to transmit data, example in Bus Topology.
- **Random Access Protocol**:
    - There is no priority system between devices.
    - They can send data at any time, whenever the link is ready.
    - **Examples**: Pure ALOHA, Slotted ALOHA, CSMA/CD, CSMA/CA
- **Controlled Access Protocol**:
    - There is a controller that controls who can send data at any point in time.
    - **Examples**: Polling, Token Passing
- **Channelization Protocol**:
    - We divide the frequency bands into multiple channels, and allocate devices to specific channels.
    - **Examples**: FDMA, TDMA

Pure ALOHA

- Formulae: here
- Type: Random Access Protocol
- After the receiver receives the data packet, it sends back an acknowledgment (ACK).
- It is a LAN based protocol.
- Collision can occur if 2 or more stations transmit data at the same time, or if a

station starts transmitting data before another one has finished.

- **Transmission Time** = Message Size / Bandwidth.
- **Vulnerable Time**: 2*Transmission Time
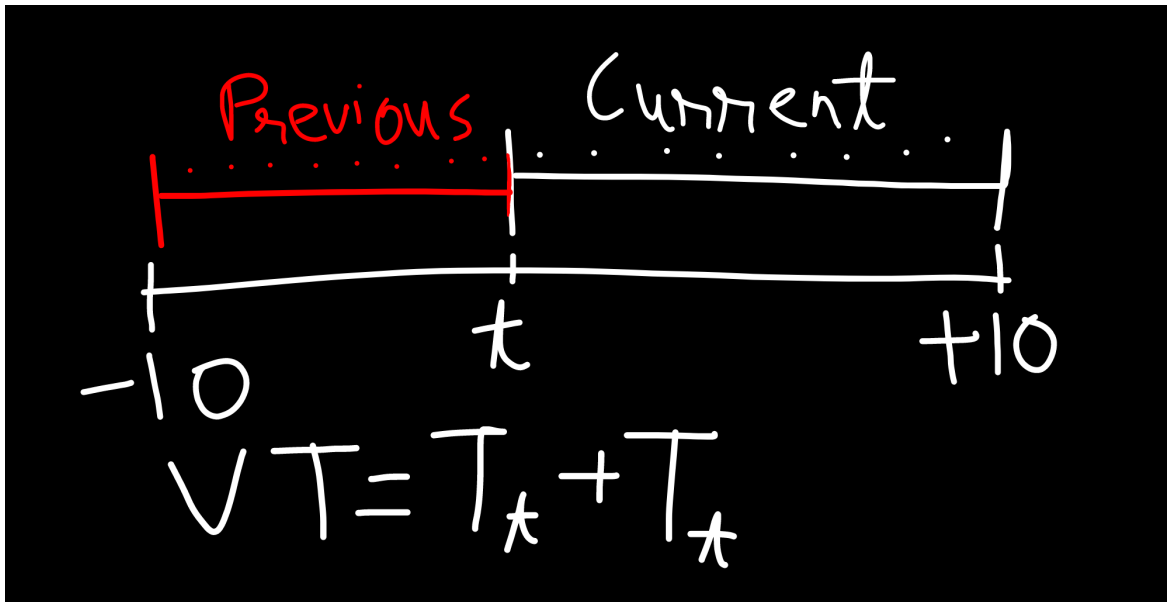
- **Efficiency**: $G * e^{-2G}$. Maximum Efficiency: $G = 1/2, 1/2 * e^{-2*(1/2)} = 1/2 * e^{-1} = 18.4\%$
  - G: Number of stations currently transmitting data.

Slotted ALOHA

- Formulae: here
- The time is divided into slots, each slot size being the Transmission Time.
- A station can only start transmitting data at the beginning of the slot.
- Collision can only occur if 2 or more stations start transmitting data at the beginning of the same slot.
- **Vulnerable Time**: Transmission Time
- **Efficiency**: $G * e^{-G}$. Maximum Efficiency: $G =, 1 * e^{-1} = 36.8\%$
  - G: Number of stations currently transmitting data.

CSMA (Carrier Sense Multiple Access)

- Formulae: here

- A particular station senses if there is any data being actively transmitted. It cannot sense the entire medium, it can only sense if there's any data transmission occuring in front of it.

- 3 Types:

  - **1-Persistent**:
    - Station will constantly utilize resources and sense if the medium is being used.
    - Collision: If multiple stations are scanning the medium at the same time, then the moment it becomes free (idle), all of them will start transmitting data, which will lead to collisions.
    - Use-case: Wired LAN / Ethernet.
  - **0-Persistent**:
    - If the station checks the medium and it is busy, it will wait for a random amount of time before checking again.
    - Medium can get freed up before the wait time of the station expires, which would mean the station waits longer than needed.
    - Collision: Less than 1-Persistent, since all stations are waiting for distinct and random amount of time.
 - **P-Persistent**:
    - Station will constantly utilize resources and sense if the channel is being used.
    - Once medium is free, it will start transmitting data according to the value of Probability §, which ranges between 0 and 1.
    - Hybrid approach, combines 1-Persistent & 0-Persistent.
    - Use-case: WiFi (Wireless Fidelity).
- **CSMA/CD**:

  - Formulae: here
  - Usage: Wired Networks
  - CSMA, with Collision Detection.
  - **No Acknowledgement is used**, because transmitting both data and acknowledgments will lead to more collisions.
  - When A is transmitting data and it receives a collision signal. Then, A knows that the data it sent, collided.
  - When A is not transmitting data (it has stopped transmission), and it receives a collision signal, A will not be able to figure out which station's data collided.
  - Transmission Time should be atleast twice the amount of Propagation Delay.
    i. `A` starts transmitting data at **12pm**. **Propagation Delay is `1 hour`** . So, it

will complete transmitting data by **1pm**.

   ii. At **12:59:59**, `B` senses the channel. It notices that it's free, and start transmitting data. This will cause a collision.

  iii. But, the collision signal from it will take another hour to reach `A`, ie at **2pm**.

  iv. If `A` stops transmitting data at **1pm**, then at **2pm** it will not be able to detect whether it was it's own data that collided, at **2pm**.

   v. This is why collision detection is only possible if Transmission Time is atleast twice the amount of Propagation Delay.
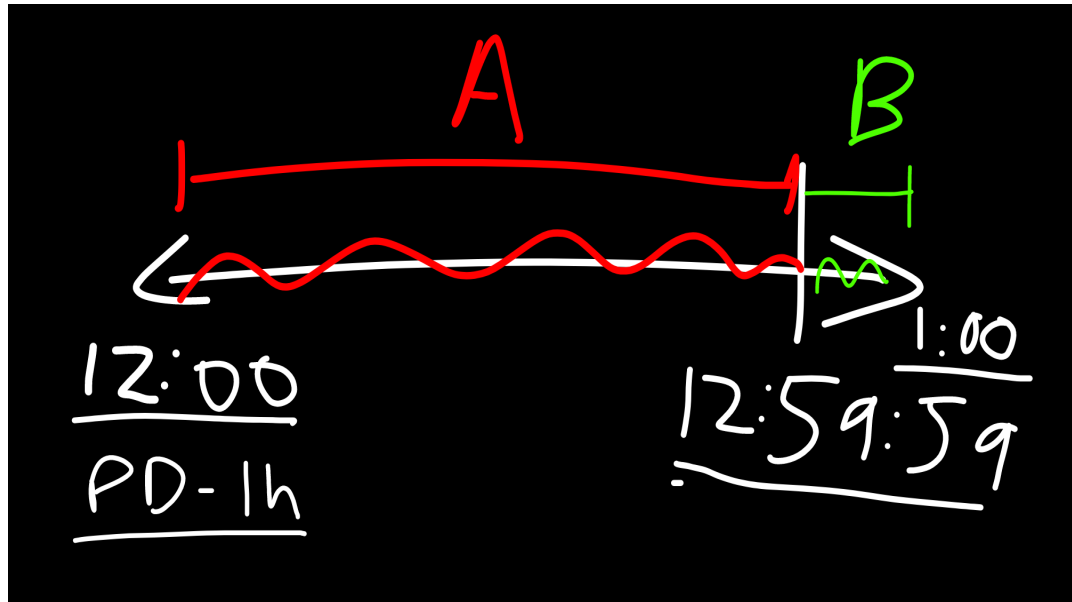
- Propagation Delay: The amount of time it takes for the receiver to receive the data, after the sender has sent it.
- **Transmission Time** = Message Size / Bandwidth.
- Transmission Time $\geq$ 2*Propagation Delay
- Message Size $\geq$ 2 * Propagation Delay * Bandwidth
- Example 0: Consider a CSMA/CD network that transmits data at a rate of 100Mbps (10^8 bits per second) over a 1km cable with no repeaters. If the minimum frame size required for this network is 1250bytes, what is the signal speed (km/second) in the cable?
  - Transmission Time $\geq$ 2*Propagation Delay
  - Transmission Time = Message Size / Bandwidth
  - Message Size $\geq$ 2 * Propagation Delay * Bandwidth
  - Propagation Delay: Distance / Velocity
  - **We have to find Velocity**.

   i. Transmission Time = 2 * Propagation Delay

ii. Message Size = 2 * Propagation Delay * Bandwidth

iii. Message Size = 2 * (Distance/Velocity) * Bandwidth

iv. **Velocity = (2 * Propagation Delay * Bandwidth) / (Message Size)**

v. $Velocity = (2 * 1km * 10^8)/(1250 * 8) = 20000 \ km/second$

- **CSMA/CA**:

  ○ CSMA, with Collision Avoidance.

  ○ Steps:

    i. START, set $k = 0$ (k=number of attempts).

    ii. Sense the carrier.

      ▪ If medium is being used, wait.

      ▪ If medium is idle, wait for an amount of time equivalent to IFS (Interframe Space)/DCF (Distributed Coordination Function) / DIFS (Distributed Interframe Space).

    iii. In the Contention Window, **it will choose it's slot number** equal to a random number between $0 \ to \ 2^k - 1$. For 1st attempt, the slot number will be $0 \ to \ 2^0 - 1$.

    iv. It will sent an RTS (Ready-To-Send). This signal will be transmitted to the hub the device is connected to. Then, it'll wait for a CTS (Clear-to-send) timer.

    v. If it receives the CTS before the timeout, it waits for another IFS amount of time, **then finally sends the frame**.

      ▪ If it receives an ACK, well and good.

      ▪ If it does not receive an ACK, then $k = k + 1$. It is checked if $k < limit$, then the station waits for T_b seconds.

      ▪ $T_b = R * T_s$ (T_b=Back-off time, R=Random Value, T_s=Slot Time).

Start

K = 0

Idle channel — No

K = number of attempts
Tp = max propagation time
Tb = Backoff time

Yes

Wait IFS time

Still idle — No

Yes

Choose a random
number R between
0 and 2^k - 1

Wait R slots

Send frame

Wait time-out

No

K>Kmax

K = K+1

ACK
Received

Yes

Abort

Success
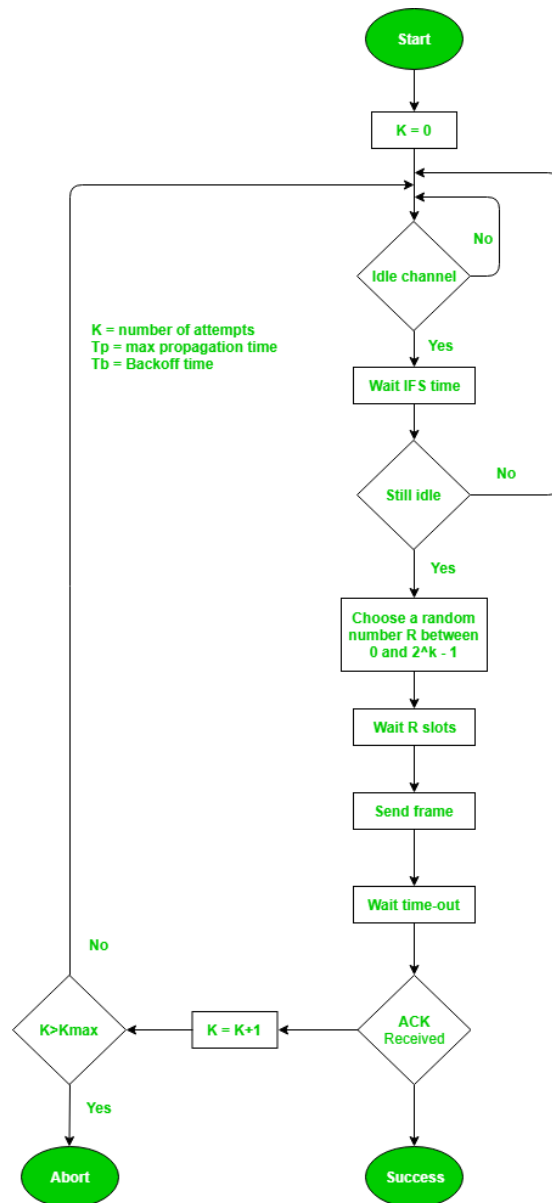
Image taken from here

Ethernet Frame Format

- IEEE 802.3 | Used in Bus Topology

- Defined in Data Link Layer

- Types of Ethernet:

    - 10Base2: 10mbps, Baseband, 200 metre range (Thin)
    - 10Base5: 10mbps, Baseband, 500 metre range (Thick)
    - 10BaseT: 10mbps, Baseband, 100 metre range
    - 100BaseFx: 100mbps, Baseband, Fiber Channel

- 10GBaseT: 10Gbps, Baseband, 100 metre range
    - …etc
- Format:

Preamble (7)	SFD (1)	DA (6)	SA (6)	Length (2)	Data (46-1500)	CRC (4)

- **Preamble**:
    - 7 bytes | 56 bits
    - Constant: 10101010101010101010101010101010101010101010101010101010
    - Set by the Physical Layer, it is used to alert the receiver that the sender is about to send data.
- **SFD**:
    - 1 byte | 8 bits
    - Constant: 10101011
    - Used to alert the receiver that the sender is about to send data.
    - Set by the Physical Layer. Initially, only SFD was used because it was assumed that the pattern will never occur within the data, but later Preamble was added.
- **Destination MAC Address (DA)**:
    - 6 bytes | 48 bits
    - Variable
    - The function of the data link layer is to make sure the data frame reaches the next node. So, it puts the DA of the next node (hop) here.
- **Source MAC Address (SA)**:
    - 6 bytes | 48 bits
    - Variable
    - The MAC Address of the sender is put here.
- **Length**:
    - 2 bytes | 18 bits
    - Variable
    - Length of the data frame, that is being sent.
- **Data**:
    - 46 - 1500 bytes | 360 - 12000 bits

- The pure data frame that is being sent
  - **Cyclic Redundancy Check (CRC)**:
    - 4 bytes | 32 bits
    - Used for Error Detection
  - Total Size: ($6 + 6 + 2 + 46 + 4 = 64$ bytes) to ($6 + 6 + 2 + 1500 + 4 = 1518$ bytes), usually Preamble & SFD are excluded since they're added in the Physical Layer.

Token Ring

- IEEE 802.5 | Used in Ring Topology

- Access Control method used: is **Token Passing**. The token circulates through the ring. If a node wants to send data, it grabs the token, and then sends the data. While the token is held by a node, no other node can transmit data.

- Types of Token Release:

  - **Early Release**: Token is released as soon as the sender has finished sending the data.
  - **Delayed Release**: Sender sends the data, it passes through the ring to all devices, comes back to the sender. After sender receives it, it releases the token.
- The direction is Unidirectional, either clockwise or anti-clockwise.

- Data Rate: `4mbps` & `16mbps`

- Acknowledgement: Piggyback acknowledgment is used. Acknowledgement is sent alongwith the data.

- Differential Manchester Encoding is used.

- Variable size framing is used. Data frames are of different sizes.

- A monitor station is used.

  - If some data becomes corrupt, then it will keep travelling through the ring and never get accepted by any node. So, monitor station removes it.
  - It is used for various other purposes.

- Data Frame Format:

SD (1)	AC (1)	FC (1)	DA (6)	SA (6)	Data	CRC (4)	ED (1)	FS (1)

  - **Start-Delimiter (SD)**:
    - 1 byte | 8 bits
    - Signals the start of the data frame.
  - **Access Control (AC)**:
    - 1 byte | 8 bits
    - Information related to what bits are present in the frame.
  - **Frame Control (FC)**:
    - 1 byte | 8 bits
    - Type of Frame (Data/Control Frame)
  - **Destination MAC Address (DA)**:
    - 6 bytes | 48 bits
    - Variable
    - The function of the data link layer is to make sure the data frame reaches the next node. So, it puts the DA of the next node (hop) here.
  - **Source MAC Address (SA)**:
    - 6 bytes | 48 bits
    - Variable
    - The MAC Address of the sender is put here.
  - **Data**:
    - 0 - MAX, depends on Token Holding Time
    - The pure data the sender wants to send.
    - Token Holding Time means the maximum amount of time, the node can hold onto the token.
  - **Cyclic Redundancy Check (CRC)**:
    - 4 bytes | 32 bits
    - Used for Error Detection
  - **End-Delimiter (ED)**:
    - 1 byte | 8 bits
    - Signals the end of the data frame.

- - **Frame Status (FS)**:
      - 1 byte | 8 bits
  - Token Format: SD (1 byte) | AC (1 byte) | ED (1 byte)

    - **Start-Delimiter (SD)**:
      - 1 byte | 8 bits
      - Signals the start of the data frame.
    - **Access Control (AC)**:
      - 1 byte | 8 bits
      - Information related to what bits are present in the frame.
    - **End-Delimiter (ED)**:
      - 1 byte | 8 bits
      - Signals the end of the data frame.

Network Layer

- While Data Link layer moves data from one node (hop) to the next, Network Layer is responsible for moving data from source to destination, within a network.
- It uses logical Addressing based on IP addresses.
- Network Layer uses a variety of routing techniques to route the data, like RIP, OSPF, etc.
- Routers operate in the Network Layer.
- **Fragmentation**: If the message size than what an intermediate node can accept (intermediate node's buffer size < message size), then the message is divided into multiple fragments and transmitted further.
- **Congestion**: We try to limit the amount of data frames that can get transmitted over the network, in order to not overwhelm it.

Classful IP Addressing

Class A

- Class: IPv4 | 4 bytes | 32 bits | 2^32 IP Addresses

- Representation: Dotted Decimel

- How to identify? If the first octet (1st 8 bits) lies between 0-127, it is a Class A address.

- Format: |Network ID (8 bits)| Host ID (8+8+8 bits)|

    i. Network ID:
        ○ 1st bit is `0` , ie constant.
        ○ length: 7 bits, number of ID's possible: $2^7 = 128$
        ○ Out of 128 addresses, the first and last one as reserved as well.
        ○ So, 126 Network ID's are available in total. So, 126 networks are available.
    ii. Host ID:
        ○ length: 24 bits, number of ID's possible: $2^{24} = 16777216$
        ○ Out of 16777216 addresses, the first and last one as reserved as well.
        ○ So, 16777214 Network ID's are available in total.
    ○ Default Mask: $255.0.0.0$. It is used to find the Network ID.
- Example 0: IP Address=$64.0.0.0$

    ○ 64 is between $0 - 127$, so it is a Class A address.
    ○ $64.0.0.0$ is the `Network ID` . It represents the network.
    ○ $64.255.255.255$ is the `Direct Broadcast Address` . If someone wants to broadcast a data packet to all devices on a network, they will send it to this address.
    ○ Default Mask: $255.0.0.0$. It is used to find the Network ID from an IP address within the network.
    Steps if IP Address: $64.0.0.8$:
        i. IP Address: $64.0.0.8 = 01000000.00000000.00000000.00001000$
        ii. Default Mask: $11111111.00000000.00000000.00000000$
        iii. **Perform an AND Operation**. Result: $01000000.00000000.00000000.00000000 = 64.0.0.0$, which is the Network ID.

Gateway

IDS (Intrusion Detection System)

Firewall

Modem

Cables & Connectors

Networking hardware

Transmission modes

Multiplexing

Encoding