UNIT - 1

Introduction : Network Architecture

A network is a combination of hardware and software that sends data from one location to another. The hardware consists of the physical equipment that carries signals from one point of the network to another. The software consists of instruction sets that make possible the services that we expect from a network

Computer networks are very complex, hence it is partitioned into vertical set of levels and each level is called layers.

★ Layered Tasks:

In networking, layering means to break up the sending of messages into separate components and activities. Each component handles a different part of the communication.

Each lower layer adds its services to the higher layer to provide a full set of services to manage communications and run the application.

Layered architecture provides the independence between layers by providing the services from lower to higher layer without defining how the services are implemented. Therefore, any modification in a layer will not affect the other layers.

The basic elements of layered architecture are:

- Services : set of actions that a layer provides to the higher layer
- Protocol : set of rules that a layer uses to exchange the information with peer entity.
- Interface : It is a way through which the message is transfered from one layer to another.

A set of layers and protocols is called as network architecture Hence the requirement of a layered architecture is adopted by Divide and conquer approach which makes a

process in such a way that the unmanageable tasks are divided into small mandgable tasks thus reducing the complexity of the design.

* **OSI Architecture:**

An ISO (International standards Organization) standard that covers all aspects of network communications is the open systems Interconnection (OSI) model.

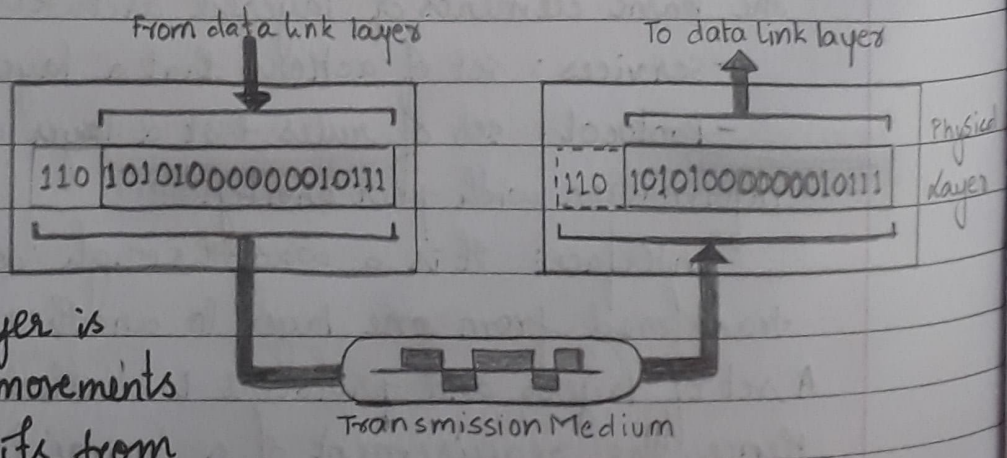
It is to facilitate communication between two different systems without requiring to change to the logic of the underlying hardware and software.

User support layers	7	Application
	6	Presentation
	5	session
	4	Transport
Network support layers	3	Network
	2	Data link
	1	Physical

Reasons for layering:

- Layered architecture
- Peer to Peer Process : The process one each machine that communicates at a given layer are called as peer to peer process.
- Interface between layers : It defines the information and services a layer must provide for the layer above.
- Organization of layers into subgroups.

- **Physical Layer:**



The physical layer is responsible for movements of individual bits from one node to another

- Physical Characteristics of interface and medium

The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.

- Representation of bits

The physical layer data consists of a stream of bits with no interpretation. Bits must be encoded into signals (electrical or optical) to be transmitted. The physical layer defines the type of encoding.

- Data Rate

The physical layer defines the transmission rate (the number of bits sent each second)

- Synchronization of bits:

The sender and receiver clocks must be synchronized.

- Line Configuration:

The physical layer is concerned with the connection of devices to media (Ex: point to point configuration or multipoint configuration)

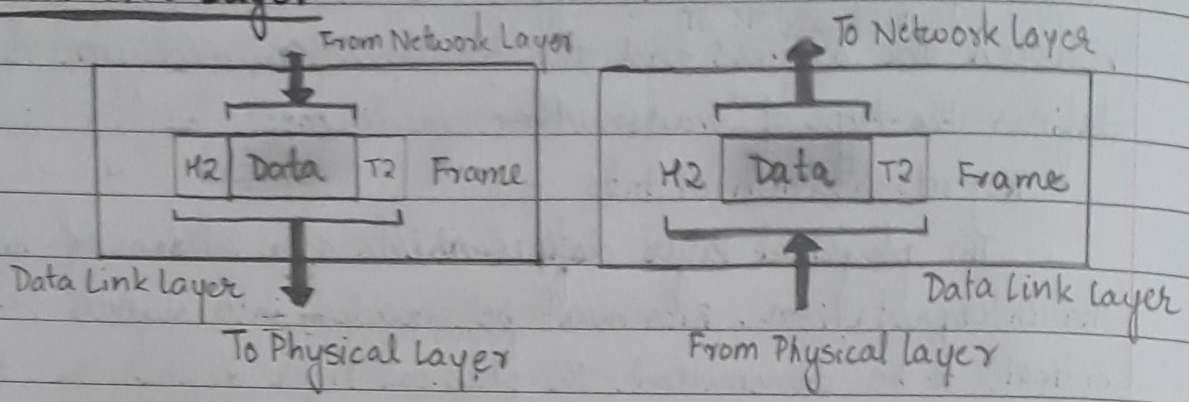
- Physical Topology:

It defines how the devices are connected to make a network such as mesh topology (every device is connected to every other device), a star topology (devices are connected through a central device), a ring topology (each device is connected to the next device), a bus topology (every device is on a common link), a hybrid topology (combination of two or more topologies)

- Transmission mode:

The physical layer also defines the direction of transmission between two devices: simplex (one device can send and other can only receive) half duplex (two devices can send and receive but not at the same time) and full duplex mode (two devices can send and receive at the same time)

- Data link Layer:



The data link layer is responsible for moving frames from one node to the next.

- Framing:

The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

- Physical Addressing:

If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame.

- Flow control:

If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

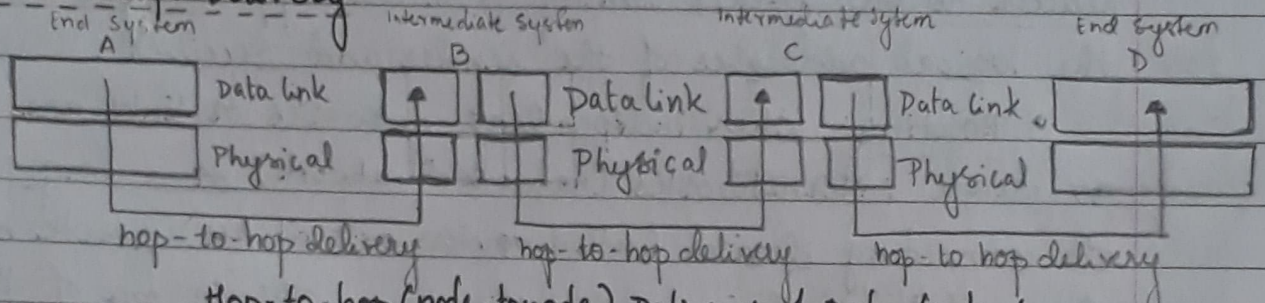
- Error Control.

The data link layer adds reliability to physical layer by adding mechanisms to detect and retransmit damaged or lost frames and also to recognize duplicate frames. Error control is normally achieved by a trailer added at the end of the frame.

- Access control:

When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time

Hop-to-Hop Delivery:

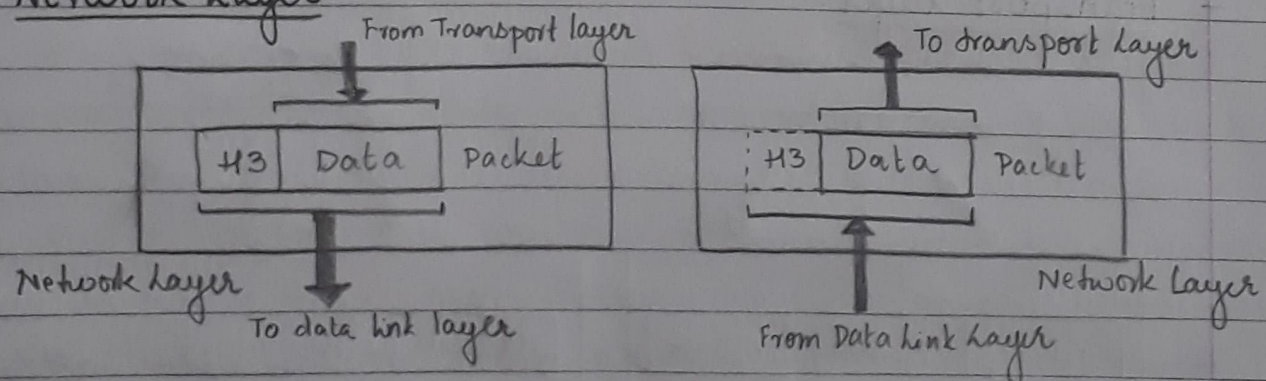


Hop-to-hop (node-to-node) Delivery of Data Link Layer.

To send data from A to D three partial deliveries are made as the communication at the data link layer occurs between two adjacent nodes.

- Data link layer at A sends a frame to the data link layer at B (a router). The frame from A to B has B as the destination address and A as the source address.
- Data link layer at B sends a new frame to the data link layer at C. The frame from B to C has C as the destination address and B as the source address.
- Data link layer at C sends a new frame to the data link layer at D. The frame from C to D has D as the destination address and C as the source address.

Network Layer:



The network layer is responsible for the delivery of individual packets from the source host to the destination host.

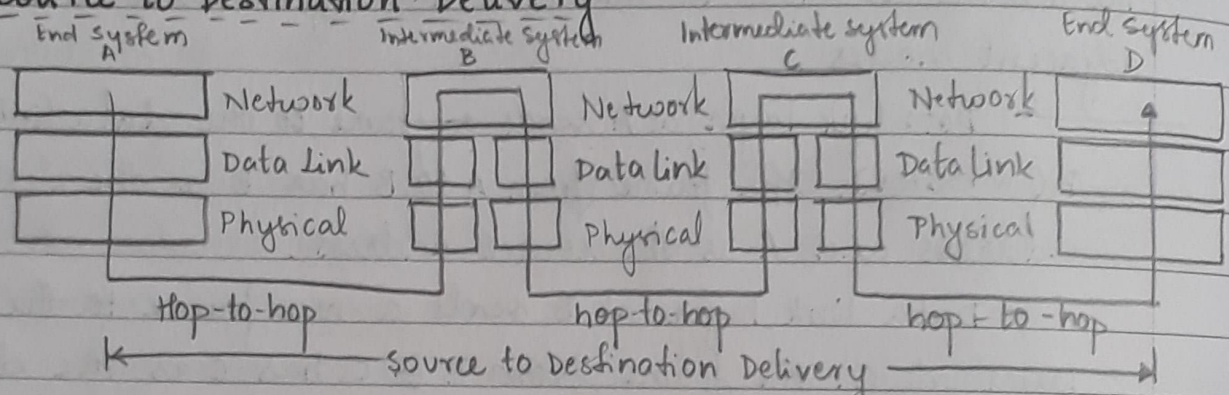
- **Logical Addressing:**

If a packet passes the network boundary, another addressing system (other than physical address) is used to distinguish the source and destination systems. The network layer adds a

header to the packet coming from the upper layer which includes the logical addresses of the sender and receiver.

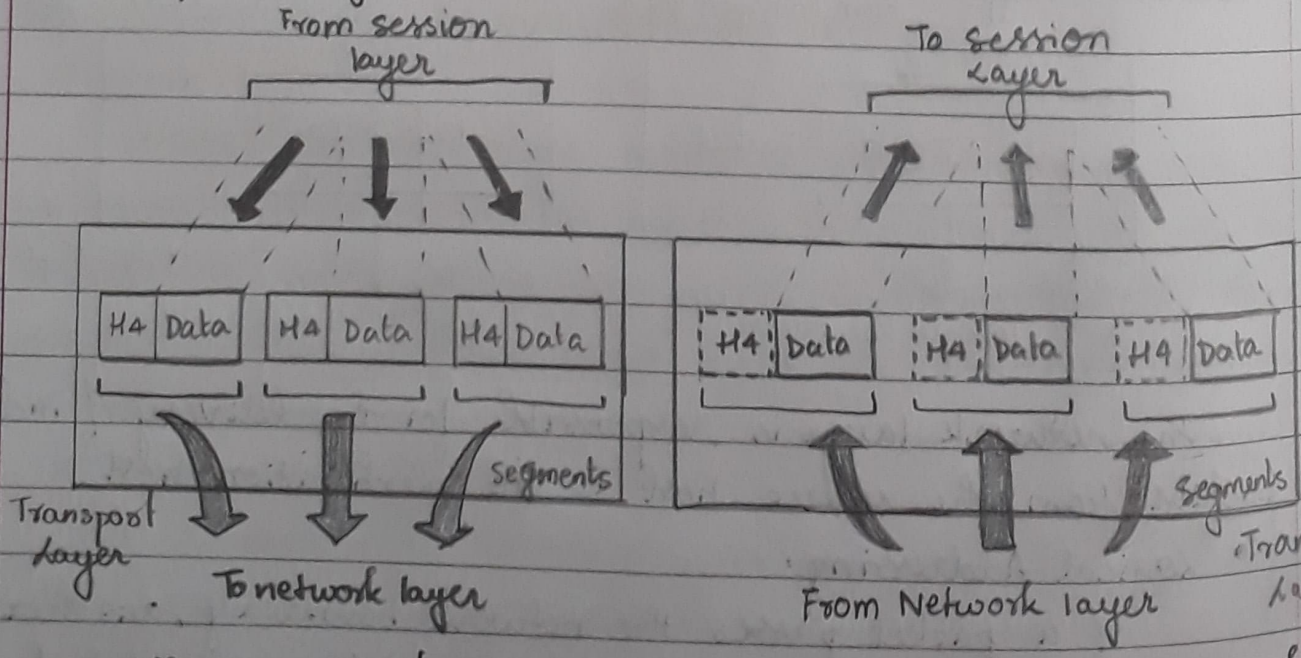
- Routing: When independent networks or links are connected to create internetworks (network of network) or a large network, the connecting devices (called routers or switches) route or switch the packets to their final destination. This is one of the functions of the network layer.

Source to Destination Delivery:



- The network layer at A sends the packet to the network layer at B. When the packet arrives at router B, the router makes a decision based on the final destination (F) of the packet.

- Transport Layer:

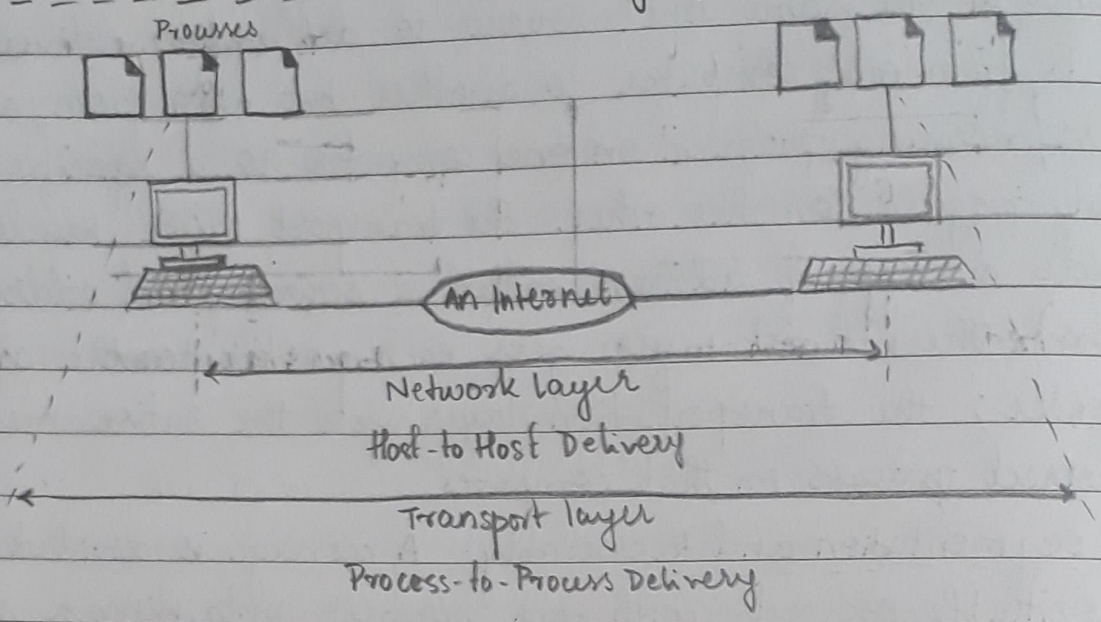


The Transport layer is responsible for the delivery of a message from one process to another. (delivery of datagrams)

Keerthana Ashok

- Service Point Addressing: Computers often run several programs at the same time. Source-to-destination delivery not only is from one computer to another but also from a specific process/running program on one computer to a specific process/running program on the other. The transport layer header must include a type of address called a service-point address (port address). The network layer gets each packet to the correct computer; the transportation layer gets the entire message to the correct process on that computer.
- Segmentation and Reassembly: A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.
- Connection Control: A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection-oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets and the connection is terminated once the data is transferred.
- Flow Control: The transport layer is responsible for flow control. which is performed end to end rather than across a single link.
- Error Control: Error control in this layer is performed process-to-process rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, duplication). Error correction is usually achieved through retransmission.

Reliable Process-to-Process delivery of a message:

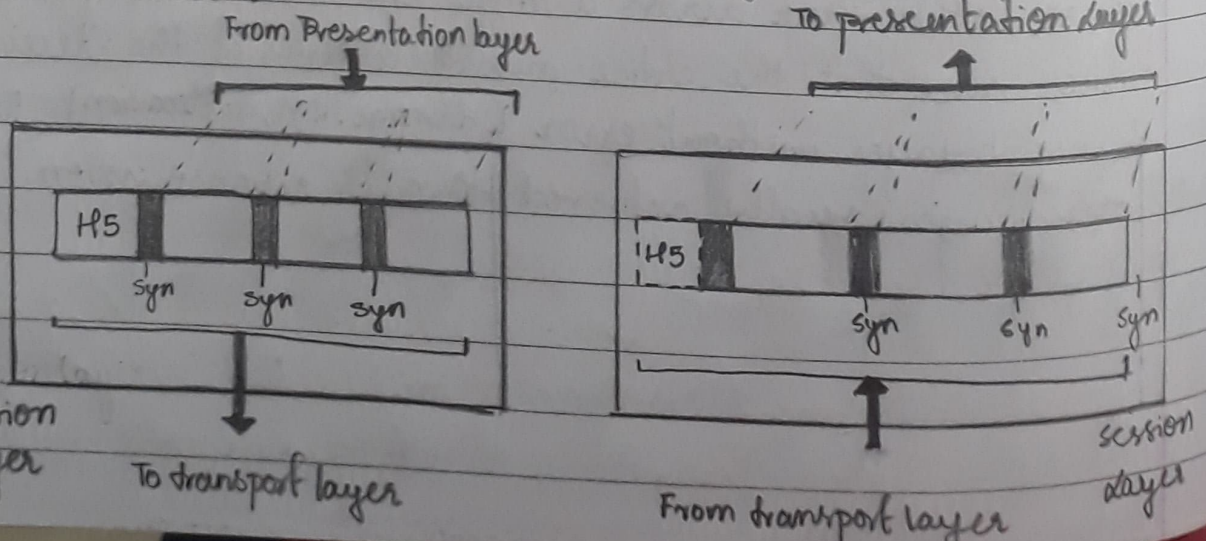


- **Session Layer:**

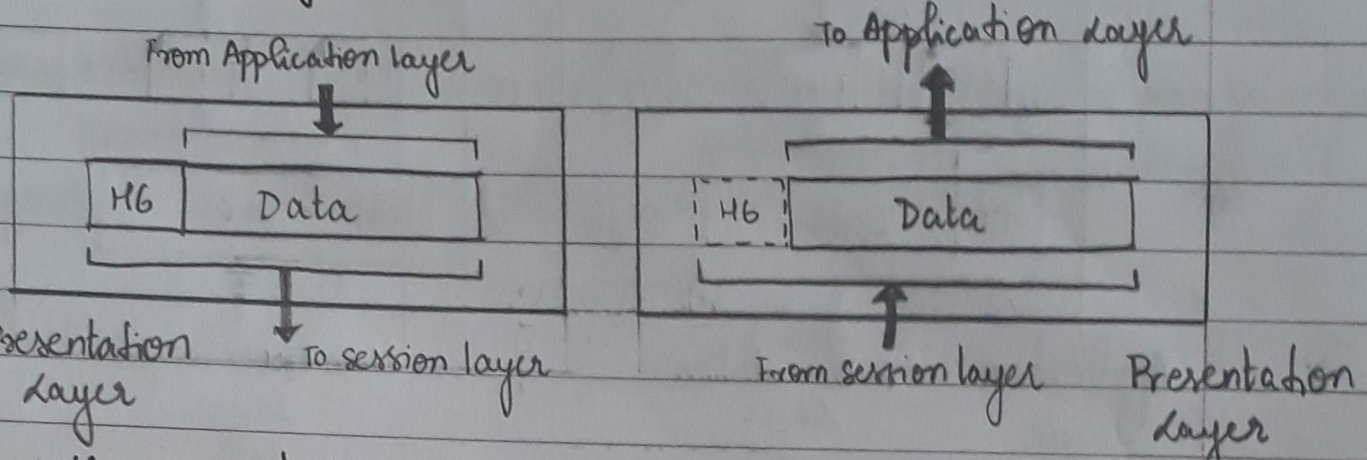
The services provided by physical, datalink and network layer is not sufficient for some processes. The session layer is the network dialog controller. It establishes, maintains and synchronizes the interaction among communicating systems.

- Dialog control: It allows two systems to enter into a dialog. It allows communication between two processes to take place either in half duplex or in full-duplex mode.
- Synchronization: It allows a process to add checkpoints or synchronization points to a stream of data.

Ex: If a system is sending a file of 2000 pages, a checkpoint after every 100 pages can be inserted to be received and acknowledged independently. If it crashes at 523 page, then only 501 to 523 must to resent.



- **Presentation Layer:**

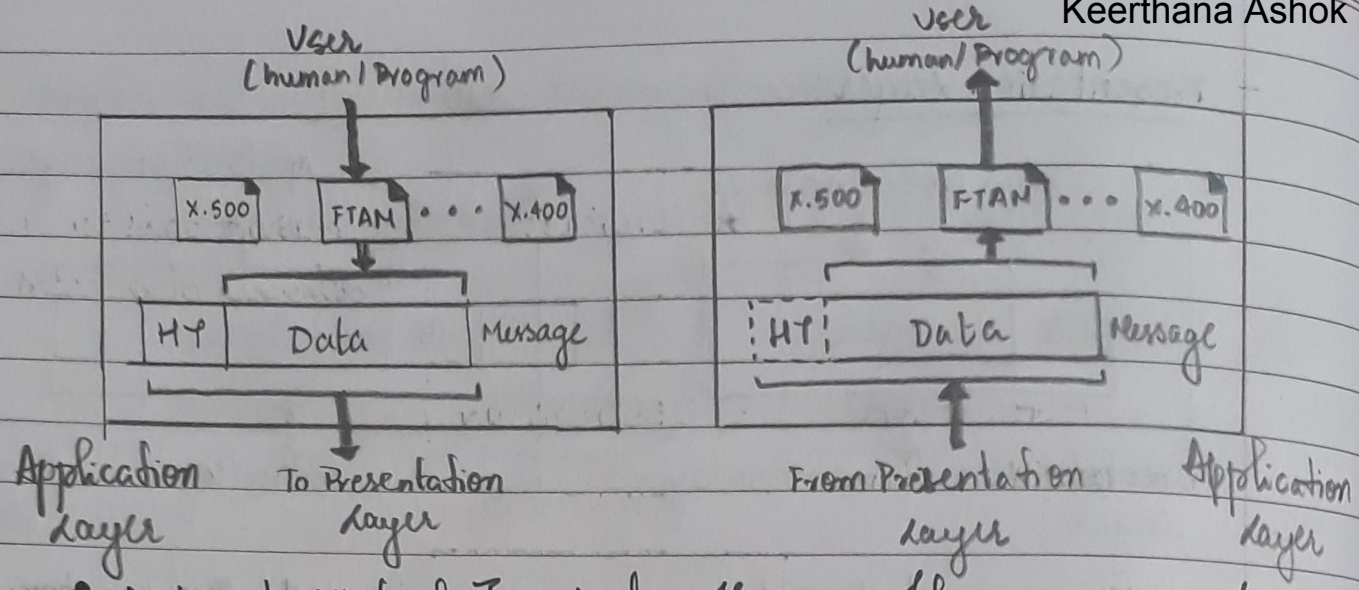


The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems. It is responsible for translation, compression and encryption.

- Translation: The information must be changed to bit streams before being transmitted. This is because different computers use different encoding methods. The presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information into common format from its sender-dependent format. and at the receiving machine it changes the common format into its receiver-dependent format.
- Encryption: To carry sensitive information the sender transforms the original information to another form and sends the resulting message over the network. Decryption reverses the original process to its original form.
- Compression: Data compression reduces the number of bits in the information which is important in the transmission of multimedia such as text, audio and video.

- **Application Layer:**

The application layer enables the user (human/software) to access the network. It provides user interface and support for services such as electronic mail, remote file access and transfer, shared database management and other types of distributed information services.



- Network Virtual Terminal: It is a software version of a physical terminal which allows a user to log on to a remote host. The application creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal which in turn talks to the host and vice versa. The remote host believes it is communicating with one of its own terminals and allows the user to log on.
- File Transfer, Access and Management: It allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer and to manage or control files in a remote computer locally.
- Mail Services: It provides the basis for e-mail forwarding and storage.
- Directory Services: This application provides distributed database sources and access for global information about various objects and services.

- Failure of OSI model:
  - TCP/IP was already in existance.
  - The protocol was not defined accurately
  - It was costing more to change from 5 to 7 layers both hardware and software changes.

Summary of layers

APPLICATION	To allow procars to process delivery of a message
PRESENTATION	To translate, encrypt and compress data
SESSION	To establish, manage and terminate sessions
TRANSPORT	To allow end to end delivery of datagram
NETWORK	To move patkets from source to destination and to provide internetworking
DATA LINK	To organize bits into frames and to provide hop-to-hop delivery
PHYSICAL	To transmit bits over a medium and to provide mechanical and electrical specifications

★ **TCP/IP Protocol suite:**

When TCP/IP is compared to OSI, it is made of five layers:

- Application
- Transport
- Network
- Data link
- Physical

Original TCP/IP protocol suite was four layer: host-to-network, internet, transport and application.

TCP/IP is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality.

The term hierarchical means that each upper level protocol is supported by one or more lower-level protocols. The layers of TCP/IP protocol suite contain relatively independent protocols that can be mixed and matched depending on the needs of the system.

- Physical and Data Link Layers:

Here TCP/IP doesnot define any specific protocol. It supports all the standard and proprietary protocols. A network in a TCP/IP internetwork can be a local-area network or a wide area network.

- Network Layer:

Here TCP/IP supports the Internetworking Protocol (IP) and in turn uses four supporting protocols: ARP, RARP, ICMP and IGNP.

- Internetworking Protocol (IP):
  - It is the transmission mechanism used by TCP/IP protocols.
  - It is unreliable and connectionless protocol.
  - It is a best effort delivery service, i.e., no error checking or tracking
  - It transports data in packets called datagrams.
  - It will store source and destination address but does not keep track of the routes.
- Address Resolution Protocol (ARP):
  - It is used to associate a logic address with a physical address.
  - It is used to find the physical address of the node when its Internet Address is known.
- Reverse Address Resolution Protocol (RARP):
  - It allows a host to discover its internet address when it knows only its physical address.
  - It is used when computer is connected to a network for the first time or when a diskless computer is booted.

- Internet Control Message Protocol (ICMP):
  - It is a mechanism used by hosts and gateways to send notification of datagram problems back to the sender.
  - It sends query and error reporting messages.
- Internet Group Message Protocol (IGMP):
  - It is used to facilitate the simultaneous transmission of a message to a group of recipients.

- Transport Layer:

Traditionally it is represented in TCP/IP by two protocols: TCP and UDP. They are transport level protocols responsible for delivery of a message from a process to another process. SCTP has been devised for newer applications needs.

- User Datagram Protocol (UDP):
  - It is a process to process protocol that adds only port addresses, checksum error control and length information to the data from the upper layer.
- Transmission Control Protocol (TCP):
  - It is a reliable stream transport protocol where stream means connection-oriented. A connection is established both ends of a transmission before either can transmit data.
  - At the sending end of each transmission, TCP divides a stream of data into smaller units called segments. Each segment has a sequence number for reordering after receipt, together with an acknowledgement number for the segments received.
  - At the receiving end, TCP collects each datagram as it comes in and reorders the transmission based on sequence numbers.
- Stream Control Transmission Protocol (SCTP):
  - It provides support for newer applications such as voice over the Internet. It combines the best features of UDP and TCP.

TCP	UDP
- connection oriented	- connectionless
- segments	- datagrams
- longer distance	- shorter distance
- flow control, error control	- no such control
- congestion	- no congestion
- always enabled	- enabled only when required

- Application Layer:

Application Layer in TCP/IP is equivalent to the combined session, presentation and application layers in the OSI model. It is responsible for process to process communication.

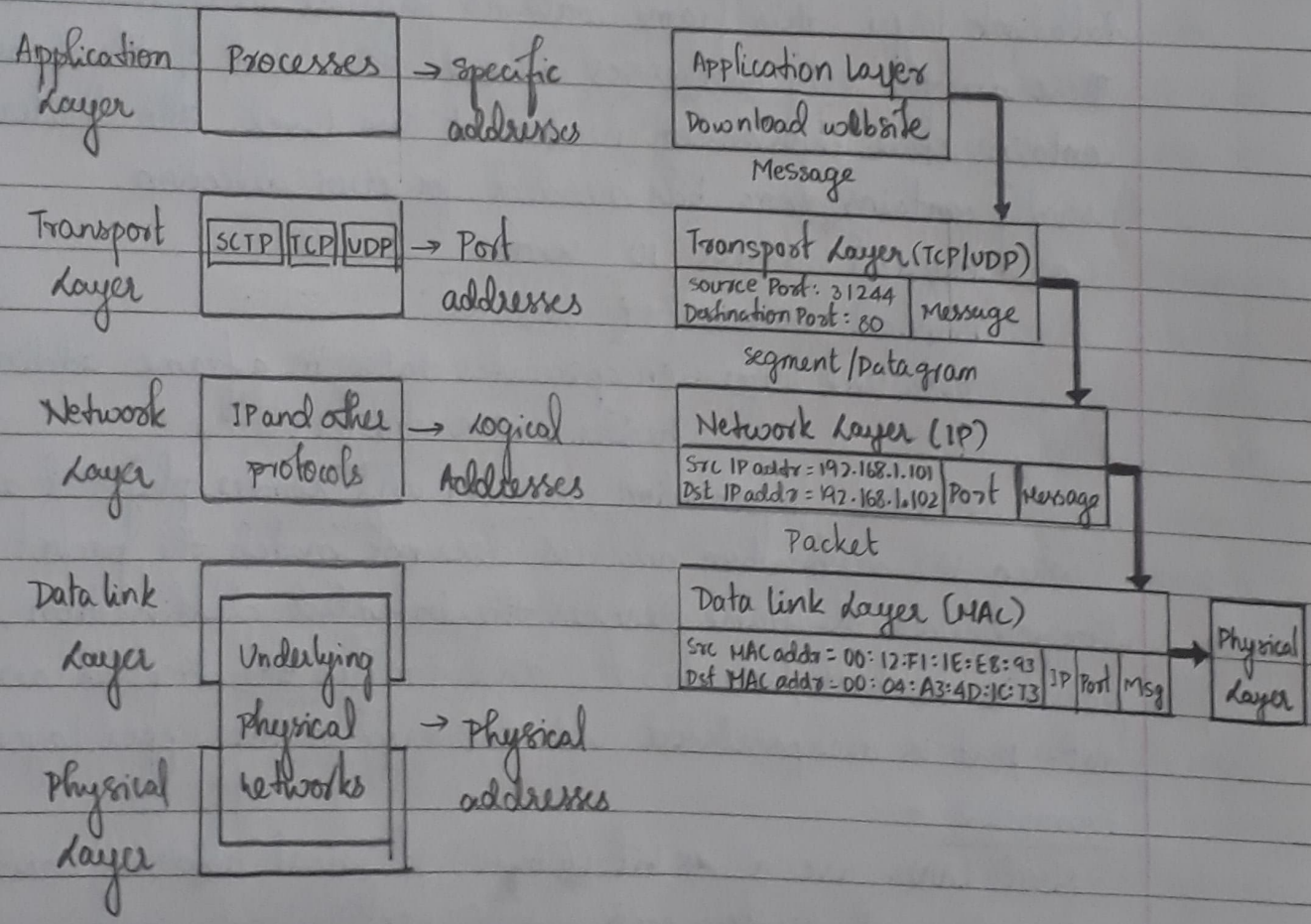
- Hyper Text Transport Protocol (HTTP):
  - For communication between web browsers and web servers
- File Transfer Protocol (FTP):
  - For the transfer of computer files between a client and server on a computer network.
- Simple Mail Transfer Protocol (SMTP):
  - It is a communication protocol for electronic mail transmission.
- Domain Name System (DNS):
  - Used by other protocols to find network layer address of computer.
- TELNET: Teletype Network:
  - Used on the internet or local area network to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection.

* Internet Architecture:
* Addressing:

Four level of addresses are used in an internet employing the TCP/IP protocols:

- physical (link) addresses
- logical (IP) addresses
- port addresses
- specific addresses.

- Relationship of layers and addresses in TCP/IP

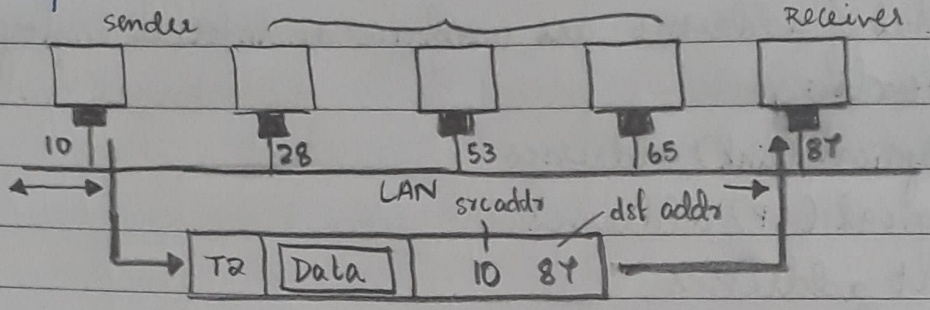


- Physical Addresses:

It is also known as link address, which is the address of a node as defined by its LAN or WAN. It is included in the frame used by the data link layer. It is the lowest-level address. The size and format of these addresses vary depending on the network

Ex: Ethernet: 6 byte (48-bit) imprinted on network interface card (NIC)

Example 1: Destination address does not match hence the packet is dropped



A node with physical address 10 sends a frame to node with physical address 87. The two nodes are connected by a link. At the datalink layer this frame contains physical addresses in the header. These are the only addresses needed. The rest of the header contains other information needed at this level. The trailer usually contains extra bits needed for error detection.

Physical address 10 : sender

Physical address 87: receiver.

Data link layer: Encapsulates data in a frame adding a header and a trailer.

Header: information, receiver and sender physical address

When the destination address does not match the packet is dropped. Once the frame reaches the intended destination, the frame is checked, the header and trailer are dropped and the data part is decapsulated and delivered to the upper layer.

Example 2

Most LANs use a 48 bit (6 byte) physical address written as 12 hexadecimal digits; each byte (2 hexadecimal digits) is separated by a colon:

07:01:02:01:2C:4B

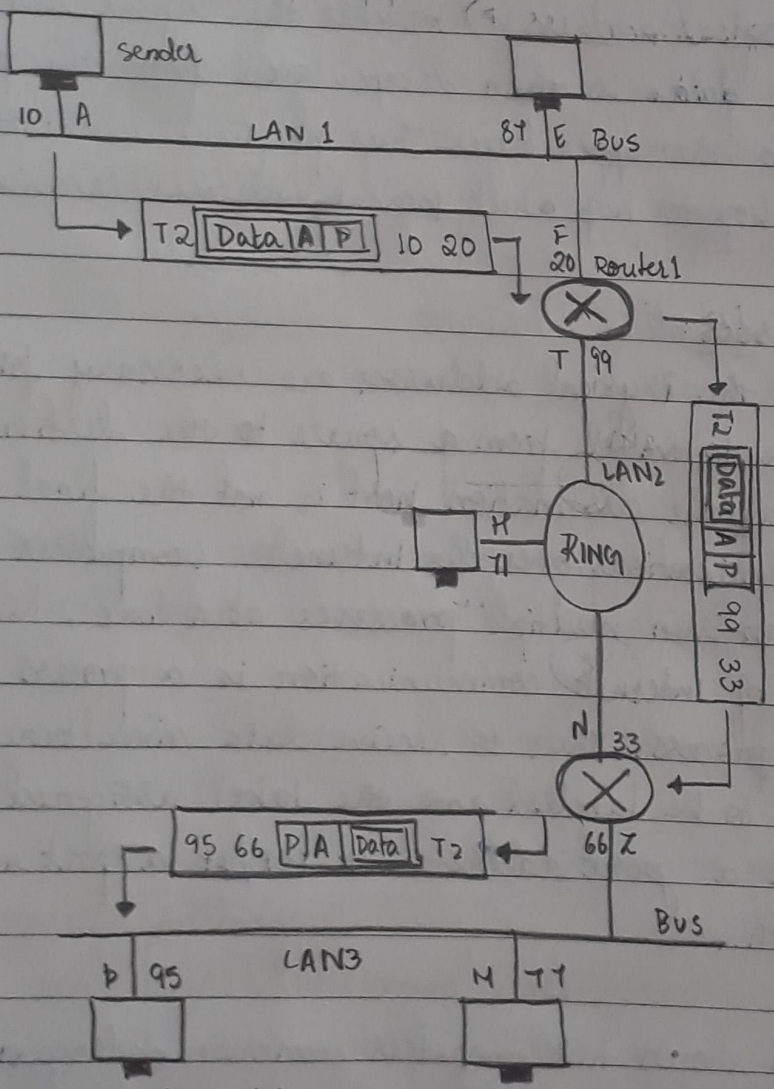
A 6byte (12 hexadecimal) physical address.

- **Logical Address:**

It is necessary for universal communications that are independent of underlying physical networks. Physical address are not adequate in an internetwork environment as different networks have different formats.

A logical address in the internet is currently a 32-bit address that can uniquely define a host connected to the Internet. No two publicly addressed and visible hosts on the Internet can have the same IP address.

Example 3



In this example it shows a part of an internet with two routers connecting three LANs.

A computer with logical address A and physical address 10 needs to send a packet to the computer with logical address P and physical address 95 on another LAN. Because the two devices are located on different networks, we cannot use physical addresses only as they have only local jurisdiction.

The logical address provides the universal address that can pass through the LAN boundaries.

The sender encapsulates its data in a packet at the network layer and adds two logical addresses (A and P). The logical source address comes before the destination address (opposite to physical addres

Stage 1: Source Physical Address : 10 (LAN1) source
Destination Physical Address: 20 (Router 1)

Stage 2: Source Physical Address : 99
Destination Physical Address : 33 (Router2)

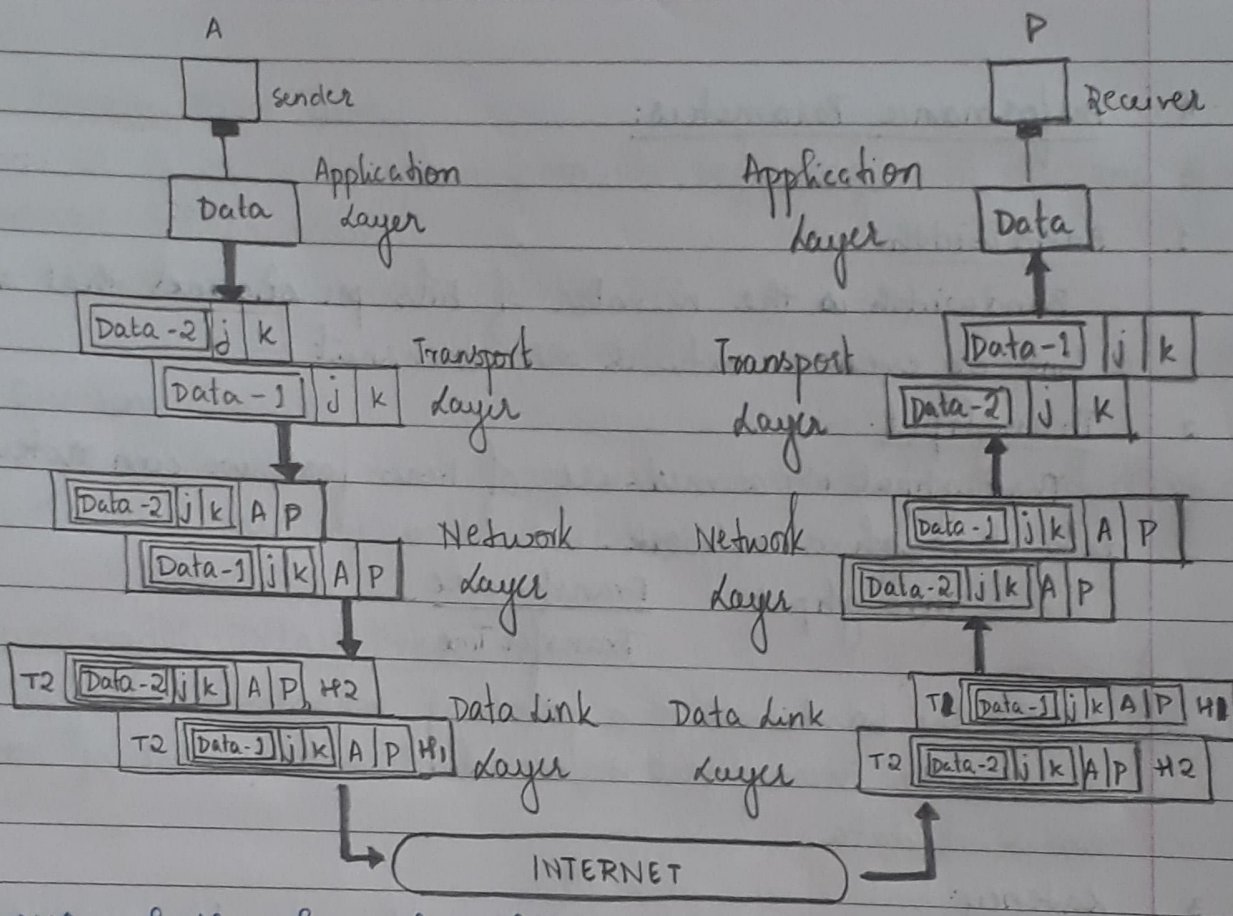
Stage 3: Source Physical Address: 66
Destination Physical Address: 95 (LAN 3) Destination the logical source and destination address must remain the same, otherwise the packet will be lost. When the frame reaches the destination, the packet is decapsulated and the destination logical address (P) matches the logical address of the computer. The data is then decapsulated from the packet and is delivered to the upper layer. Thus physical addresses will change from hop to hop but logical addresses remain the same.

- Port Addresses:

The IP and the Physical addresses are necessary for a quality of data to travel from a source to the destination host. But arrival at the destination host is not the final objective of data communications over the Internet. Computers are devices that can run multiple processes at a time, hence the final objective of Internet communication is a process communicating with another process. Thus to receive data simultaneously processes need to be labeled and the label assigned to a process is called a port address. In TCP/IP a port address is 16 bits in length.

Example 4:

The example shows two computers communicating via the internet. Here j is the address of sending process and k is the address of the receiving process. Since the data size is larger than the network layer can handle, the data is split into two packets with each packet retaining the port addresses j and k. Then in the network layer it is encapsulated with logical source and destination addresses (A and P). This packet is encapsulated in a frame with physical source and destination addresses for the next hop. The physical addresses is not shown because they change from hop to hop inside the cloud designated as Internet



Although the physical addresses change from hop to hop, logical and port addresses remain the same from the source to destination.

Example 5:

The port address is a 16-bit address represented by one decimal number as shown below:

753 (A 16 bit port address represented as one single number)

- Specific Addresses:

Some applications have user friendly addresses that are designed for that specific address.

Ex: e-mail address: keerthi.ashok23@gmail.com

It defines the recipient of an e-mail.

Universal Resource Locator (URL): www.google.com

It is used to find a document on the World Wide Web.

These addresses, however, get changed to the corresponding port and logical addresses by the sending computer.

* Performance Parameters:

1. **Bandwidth:**
Bandwidth is the number of bits per second that a channel, a link or even a network can transmit.

2. **Throughput:**
Throughput is a measure of how fast we can actually send data through a network.

$$\text{Throughput} = \frac{\text{Transfer size}}{\text{Transfer Time}}$$

NOTE: The bandwidth is a potential measurement of a link whereas the throughput is an actual measurement of how fast we can send data.

3. **Latency:**
Latency / delay defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is sent out from the source. The latency has four components: propagation time, transmission time, queuing time and processing delay.

$$\text{Latency} = \underset{\text{Time}}{\text{Propagation}} + \underset{\text{Time}}{\text{Transmission}} + \underset{\text{Time}}{\text{Queuing}} + \underset{\text{Delay}}{\text{Processing}}$$

- **Propagation Time:**
It defines the time required for a bit to travel from the source to the destination.

$$\text{Propagation Time} = \frac{\text{Distance}}{\text{Propagation Speed}} = \frac{D}{V}$$

The propagation speed of electromagnetic signals depends on the medium and frequency of the signal.

Ex: In vacuum, light is propagated with a speed of 3×10^8 m/s. It is lower in air and much lower in cable.

- **Transmission Time:**

It is the time taken from the beginning until the end of a message transmission.

$$\text{Transmission Time} = \frac{\text{Message size}}{\text{Bandwidth}}$$

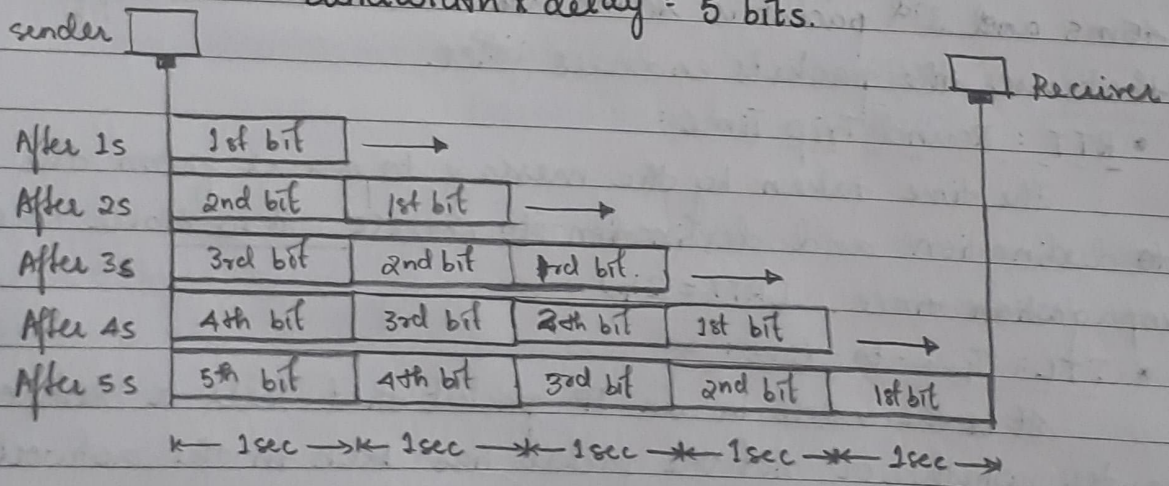
- **Queuing Time:**

It is the time needed for each intermediate or end device to hold the message before it can be processed.

- **Bandwidth-Delay Product:**

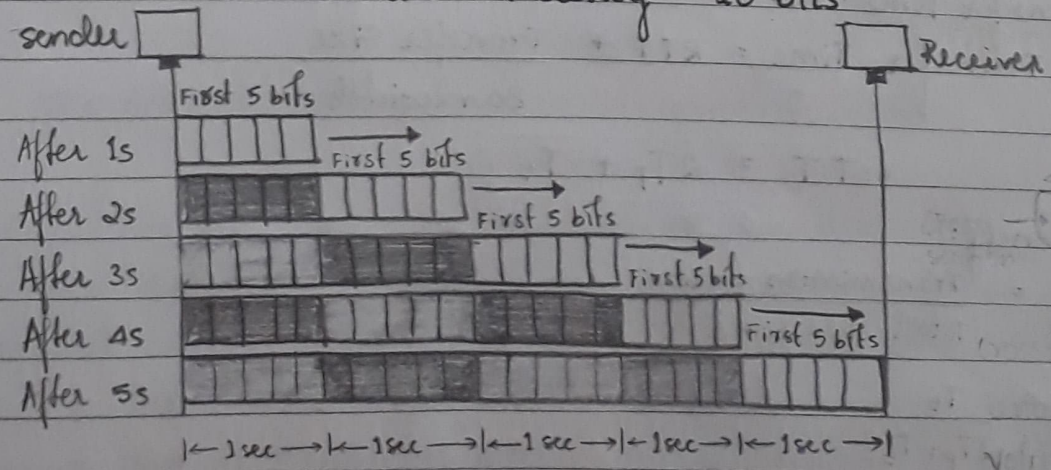
Case 1: bandwidth : 1bps delay : 5s

Bandwidth x delay = 5 bits.



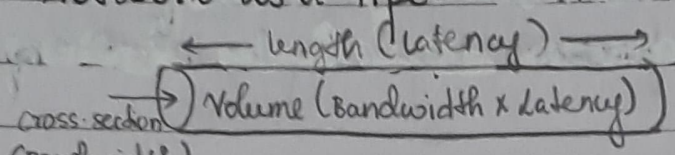
Case 2: bandwidth : 4 bps delay : 5s

Bandwidth x delay = 20 bits



The band-width - delay product defines the number of bits that can fill the link.

Network as a Pipe:



Ex: Latency: 50ms
Bandwidth: 50 Mbps
Pipe can hold
Bandwidth-Delay Product
$= 50 \times 10^6 \times 50 \times 10^{-3} = 50 \times 10^3$ bits
If the pipe is not filled, bandwidth is wasted by the sender.

- **Jitter:**

Another performance issue that is related to delay is jitter. It is a problem when different packets of data encounter different delays and the application using the data at the receiver site is time sensitive (audio and video data). Ex: If the delay for 1st packet is 20ms, 2nd packet is 45ms and 3rd packet is 40ms, then the real-time application that uses the packets endures jitter.

- **RTT: Round Trip Time:**

The time taken by the message to travel from the source to destination and destination to source. It is twice the propagation time. ($RTT = 2T_p$)

- **TTL: Time-to-Live:**

It is the value in an IP packet that tells a network router whether or not the packet has been in the network too long and should be discarded.

- **Transfer Time: TT:**

$$\text{Transfer Time} = RTT + \frac{\text{Transfer Size}}{\text{Bandwidth}}$$

$$TT = 2T_p + T_t$$

- **Efficiency:**

$$\eta = \frac{\text{Transmission time}}{\text{Total cycle Time}}$$

$$\eta = \frac{T_t}{2T_p + T_t}$$

* **Problems:**

Q1. A network with bandwidth of 10 Mbps can pass only an average of 12,000 frames per minute with each frame carrying an average of 10,000 bits. What is the throughput of this network?

— Throughput = $\frac{\text{Transfer size}}{\text{Transfer time}} = \frac{10,000 \times 12,000}{60} = \frac{200}{} = 2\ \text{Mbps}$

The throughput is almost one-fifth of the bandwidth.

Q2. What is the propagation time if the distance between the two points is 12,000 km? Assume the propagation speed to be 2.4×10^8 m/s in cable.

— Propagation Time = $\frac{\text{Distance}}{\text{Propagation speed}} = \frac{12000 \times 10^3}{2.4 \times 10^8} = 50\ \text{m sec}$

Q3. What are the propagation time and the transmission time for a 2.5 kbyte message if the bandwidth of the network is 1 Gbps? Assume the distance between the sender and the receiver is 12,000 km and that light travels at 2.4×10^8 m/s.

— Propagation time = $\frac{\text{Distance}}{\text{Speed}} = \frac{12000}{2.4 \times 10^8} = 50\ \text{m sec}$

Transmission time = $\frac{\text{Message size}}{\text{Bandwidth}} = \frac{2.5 \times 10^3 \times 8}{1 \times 10^9} = 0.02\ \text{m sec}$

The transmission time can be ignored.

Q4: Calculate the total time required to transfer 1.5 MB in the following cases: (Assume RTT = 80×10^{-3} sec). A packet size of 1 kB data and an initial 2RTT of handshaking before data is sent.

i. The Bandwidth of 10 MBps and data packets can be sent continuously

ii. The Bandwidth is 10 MBps but after we finish sending each data packet we must wait 1 RTT before the next.

iii. The link allows infinitely fast transmission but limits bandwidth

such that only 20 packets can be sent per RTT.

i) Bandwidth = 10MBps and data packets sent continuously.

Total time = $T_p + T_t + 2RTT$

$T_t = \frac{\text{Message size}}{\text{Bandwidth}} = \frac{1.5 \times 2^{20} \times 8}{10 \times 10^6} = 1.26$ sec

$\therefore$ Total time = $\frac{80 \times 10^{-3}}{2} + 1.26 + 2(80 \times 10^{-3})$

Total time = $0.04 + 1.26 + 0.16 = 1.46$ sec

ii) Bandwidth = 10MBps but after we finish sending each data packet, we must wait 1 RTT before the next.

Packets required : $\frac{1.5 \times 2^{20}}{1 \times 2^{10}} = 1536$ packets

Total time = 1536 (RTT) + $T_t + T_p + 2RTT$

Waiting Time

$= 1536 (80 \times 10^{-3}) + 1.46$

$= 124.34$ sec

doubt iii) The link allows infinitely fast transmit but limits the bandwidth such that only 20 packets can be sent per RTT.

T_p and T_t is not considered (ideal case).

Number of packets = 1536 packets.

$\frac{1536}{20} = 76.8$

Total Time = 76.8 (RTT) + 2RTT

$= 6.14 + (80 \times 10^{-3}) 2$

$= 6.3$ secc

Q5: Consider point to point link 50kms in length and at what bandwidth could propagation delay (at speed of 2×10^8 m/sec) equal transmit delay for:

i) 100 byte packets ii) 512 byte packets

i. $T_p = T_t$

$$\frac{\text{Distance}}{\text{speed}} = \frac{\text{Message size}}{\text{Bandwidth}}$$

$$\frac{50\times10^3}{2\times10^8} = \frac{100\times8}{BW}$$

$$BW = 32\times10^5 = 3.2 \text{ kbps}$$

ii. $T_p = T_b$

$$\frac{\text{Distance}}{\text{speed}} = \frac{\text{Message size}}{\text{Bandwidth}}$$

$$\frac{50\times10^3}{2\times10^8} = \frac{512\times8}{BW}$$

$$BW = 16.38 \text{ kbps}$$

Q6: Suppose a 128 kBps point to point link is set up between earth and rover on Mars. The distance from the earth to Mars is approximately 55 Gm and data travels over the link at the speed of light (3×10^8 m/s).

i. calculate the minimum RTT for the link.

ii. Bandwidth-Delay Product

iii. A camera on rover takes pictures of its surroundings and sends this to earth. How quickly after the picture is taken can it reach mission control on earth? Assume each image is 5MB in size.

Given: $BW = 128$ kBps $\quad d = 55\times10^9$ m $\quad c = 3\times10^8$ m/s

i) $RTT = 2T_p = 2\left(\frac{\text{Distance}}{\text{Speed}}\right) = 2\left(\frac{55\times10^9}{3\times10^8}\right) = 366.7$ sec

ii) Bandwidth-Delay Product

$BW \times \text{Delay} = 128\times10^5 \times \left(\frac{366.7}{2}\right) = 23.47$ Mbits

iii) Total time $= T_p + T_t$

$$= \frac{55\times10^9}{3\times10^8} + 328$$

$$= 183.33 + 328 = 512 \text{ sec}$$

$$T_t = \frac{5\times2^{20}\times8}{128\times10^3} = 328 \text{ sec}$$

UNIT - 2

Data Link Layer and Introduction to Socket Programming

The two main functions of the data link layer:

- data link control - node to node communication
- media access control - how to share the link.

Data link control functions include framing, flow and error control, and software implemented protocols that provide smooth and reliable transmission of frames between nodes.

★ **Framing:**

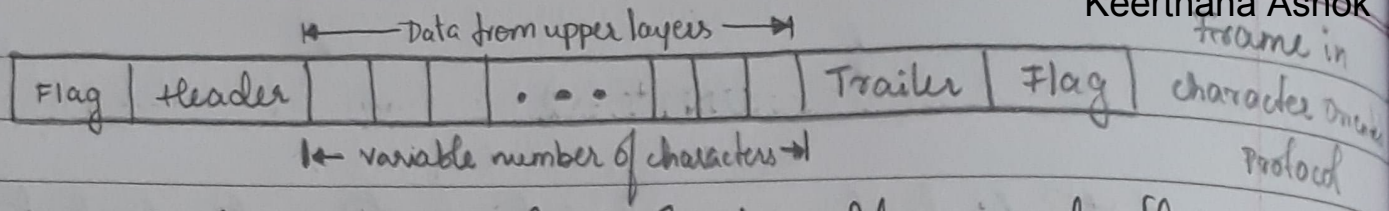
Framing in the datalink layer separates a message from one source to a destination by adding a sender address and a destination address. The destination address defines where the packet has to go and the sender address helps the recipient acknowledge the receipt.

Types of framing:

- Fixed size Framing: Here there is no need for defining the boundaries of the frames, the size itself can be used as a delimiter. Ex: Asynchronous Transfer mode (ATM) wide area network which uses frames of fixed size called cells.
- Variable Size Framing: Here the end of the frame and the beginning of the next has to be defined. Two approaches are used for this purpose.

  - Character Oriented Protocols:

    Data to be carried are 8-bit characters from a coding system such as ASCII. To separate one frame from the next, an 8-bit (1byte) flag is added at the beginning and the end of the frame. It was popular when only text was exchanged by the data link layers.

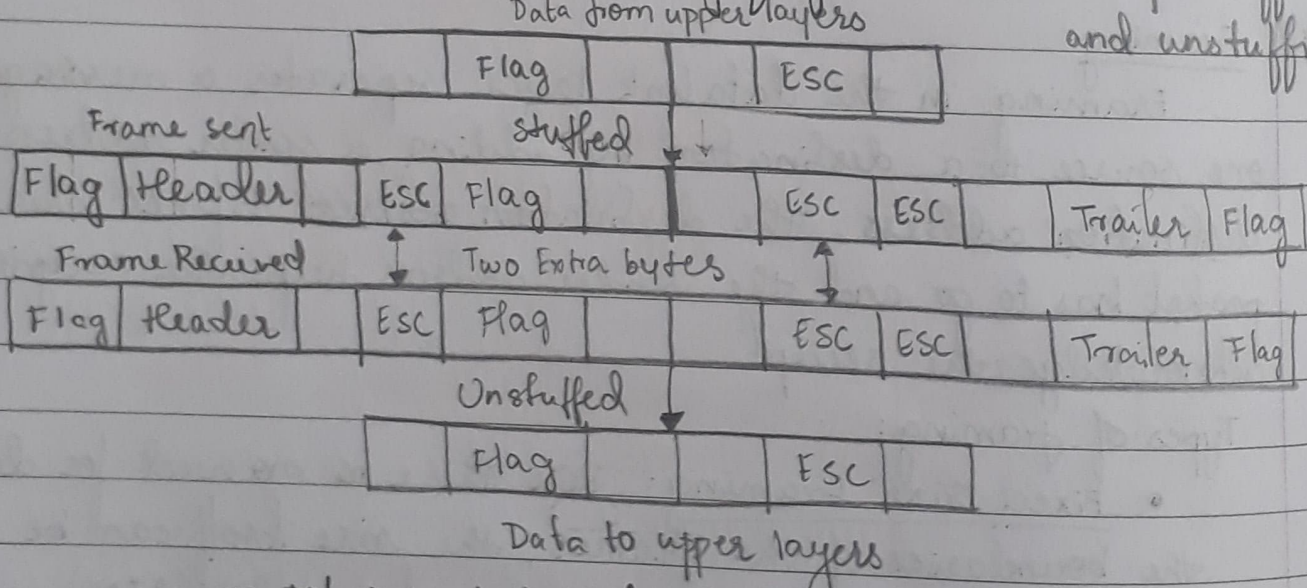


Header: source and destination address and other control information

Trailer: error detection or correction redundant bits.

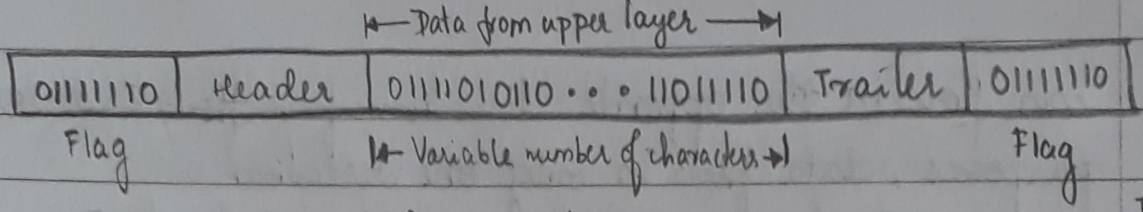
In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. This extra byte is usually called the escape character (ESC), which has a predefined bit pattern. When ever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data and not a delimiting flag.

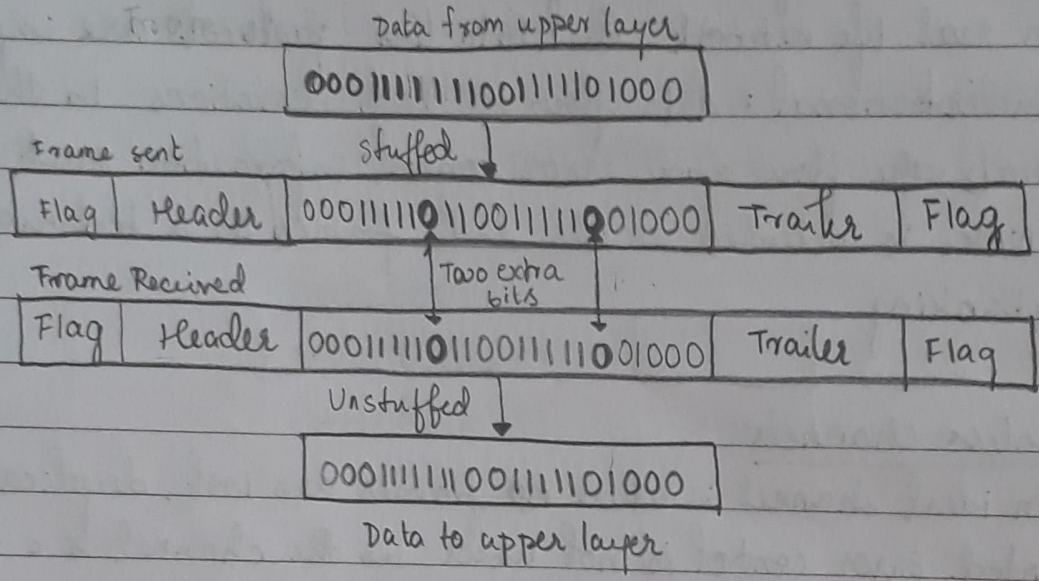
Byte stuffing and unstuffing



- Bit Oriented Protocols:

In addition to headers we still need a delimiter to separate one frame from the other. A special 8-bit pattern flag 01111110 is used as the delimiter to define the beginning and end of the frame. If the flag pattern appears in the data, we need to inform the receiver that it is not the end of the frame. This is done by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. This is called bit stuffing. In bit stuffing if a 0 followed by five 1's are encountered, an extra 0 is added





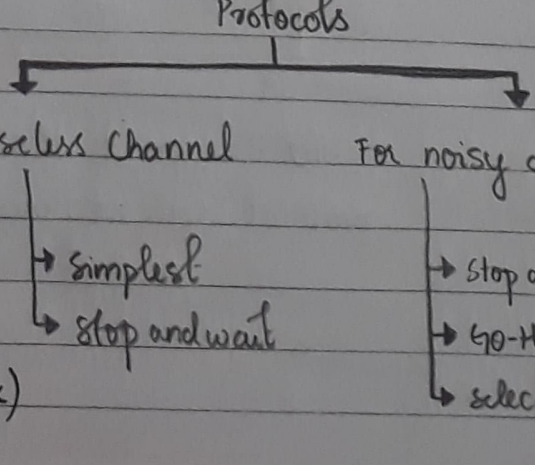
This means that if the flag like pattern 01111110 appears in the data, it will be changed to 011111010 (stuffed) and is not mistaken as a flag by the receiver.

* Flow and Error control:

- Flow control: It refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgement. Buffer is used for storing incoming data until they are processed.
- Error control: Error control in the data link layer is based on automatic repeat request (ARQ), which is the retransmission of data. Any time an error is detected in an exchange, specified frames are retransmitted.

* Protocols:

All the protocols are unidirectional i.e., from sender to receiver. Although special frames called acknowledgement (ACK) and negative acknowledgement (NAK)



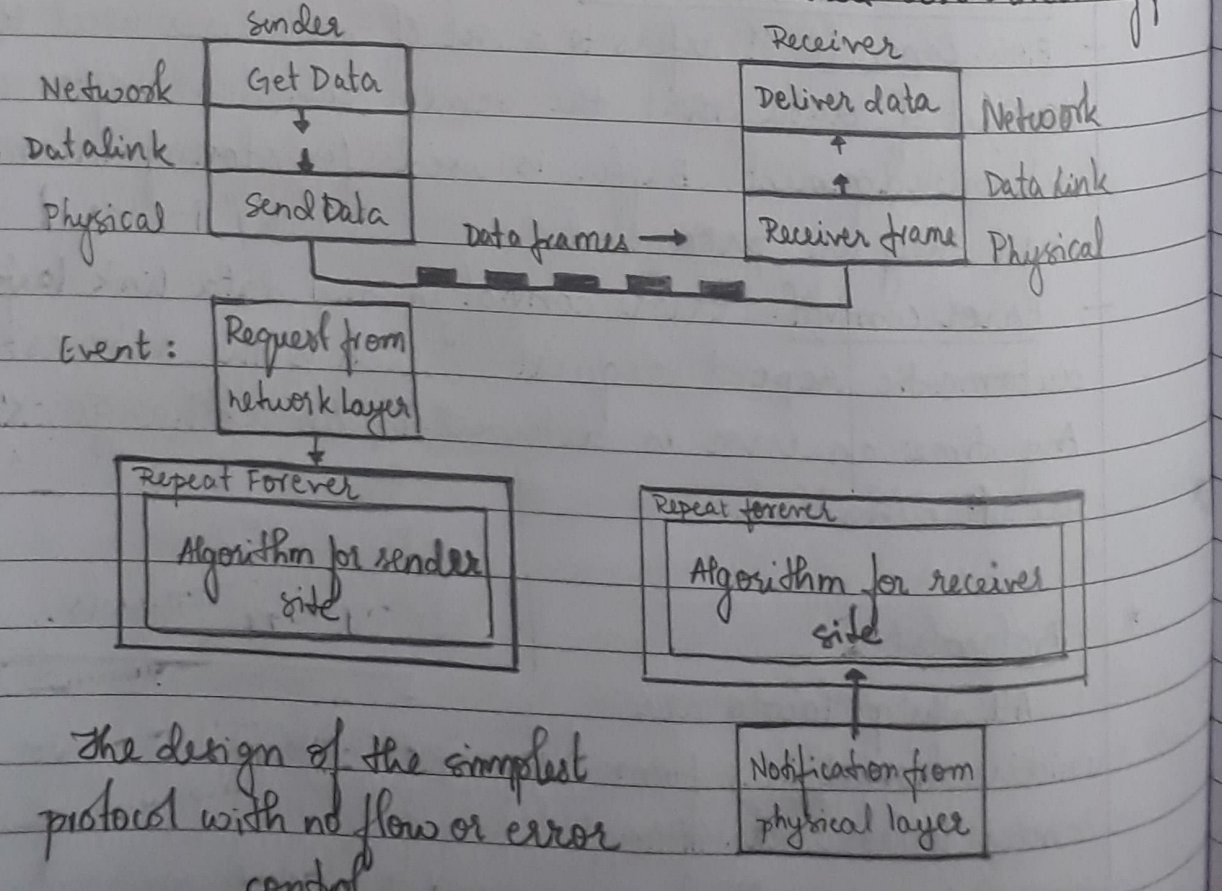
can flow in the opposite direction for flow and error control purposes, data flow in only one direction.

In real life network, the data link protocols are implemented as bidirectional: data flow in both directions. In these protocols the flow and error control information such as ACKs and NAKs is included in the data frames in a technique called piggybacking.

* Noiseless Channels:

An ideal channel is where no frames are lost, duplicated or corrupted. Error control is not used as the channel is a perfect noiseless channel.

- Simplest Protocol:
  - It has no flow or error control.
  - Unidirectional protocol: sender to receiver.
  - The receiver can never be overwhelmed with incoming frame



The design of the simplest protocol with no flow or error control

Algorithm at the sender site

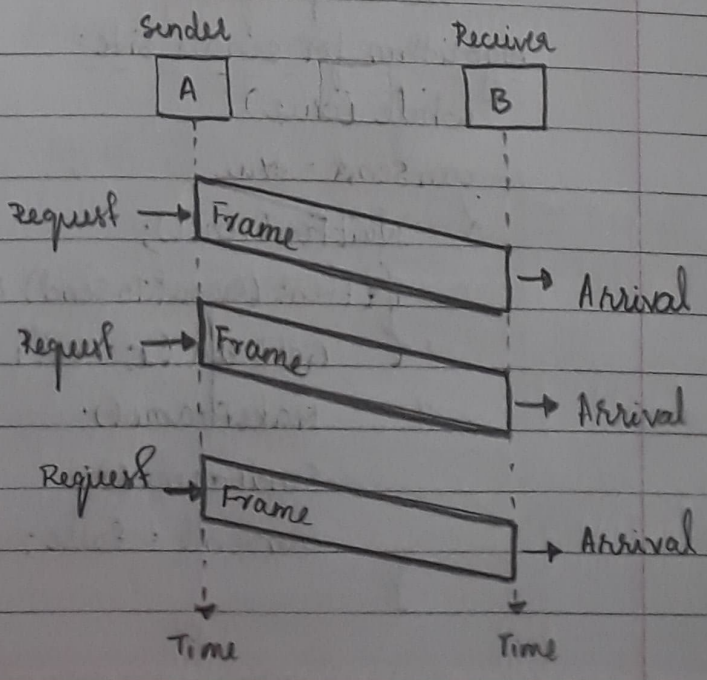
```
while (true)                          // Repeat forever
{
    WaitForEvent();                   // Sleep until event occurs
    if (Event (Request To Send))      // There is packet to send
    {
        GetData();
        MakeFrame();
        SendFrame();                  // Send the frame
    }
}
```

Algorithm at the receiver site

```
while (true)                          // Repeat forever
{
    WaitForEvent();                   // Sleep until event occurs
    if (Event (Arrival Notification)) // Data frame arrived
    {
        ReceiveData();
        ExtractData();
        DeliverData();                // Deliver data to network layer
    }
}
```

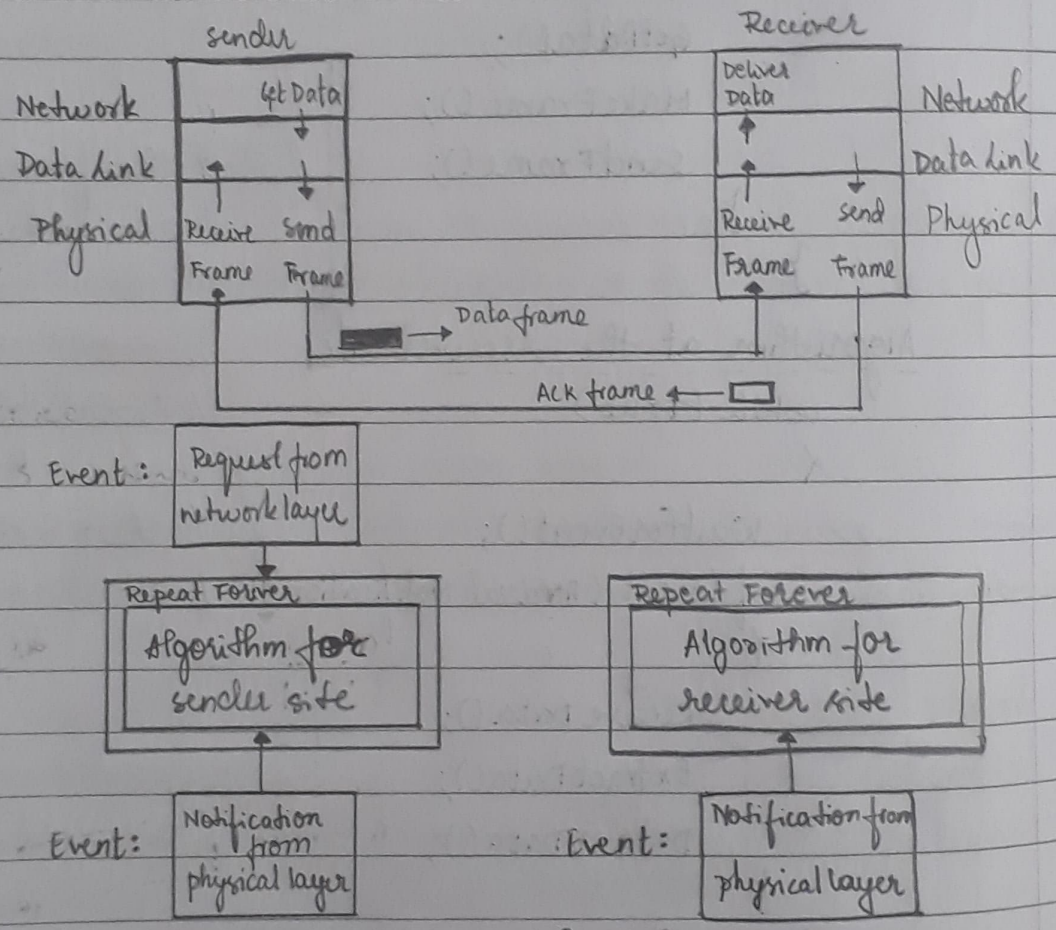
Example:

communication using simplest protocol. The sender sends a sequence of frames without thinking about the receiver. To send three frames, three events occur at the sender site and three events at the receiver site.



Stop and Wait Protocol:

- Sender sends one frame, stops until it receives confirmation from the receiver and then sends the next frame.
- Unidirectional communication for data frames but auxiliary ACK frames travel from the other direction.
- Flow control is added.



The design of stop-and-wait protocol

Algorithm for sender site:

```
while (true)                          // Repeat forever
canSend = true                        // Allow the first frame to go
{ WaitForEvent();                     // Sleep until an event occurs
  if (Event (RequestToSend) AND canSend)
  { GetData();
    MakeFrame();
    SendFrame();                      // Send the data frame
    canSend = false;                  // cannot send until ACK arrives
  }
```

```
    waitForEvent();                    // sleep until an event occurs
    if (Event (ArrivalNotification)    // An ACK has arrived
    {
        ReceiverFrame();               // Receive the ACK frame
        canSend = true;
    }
}
```

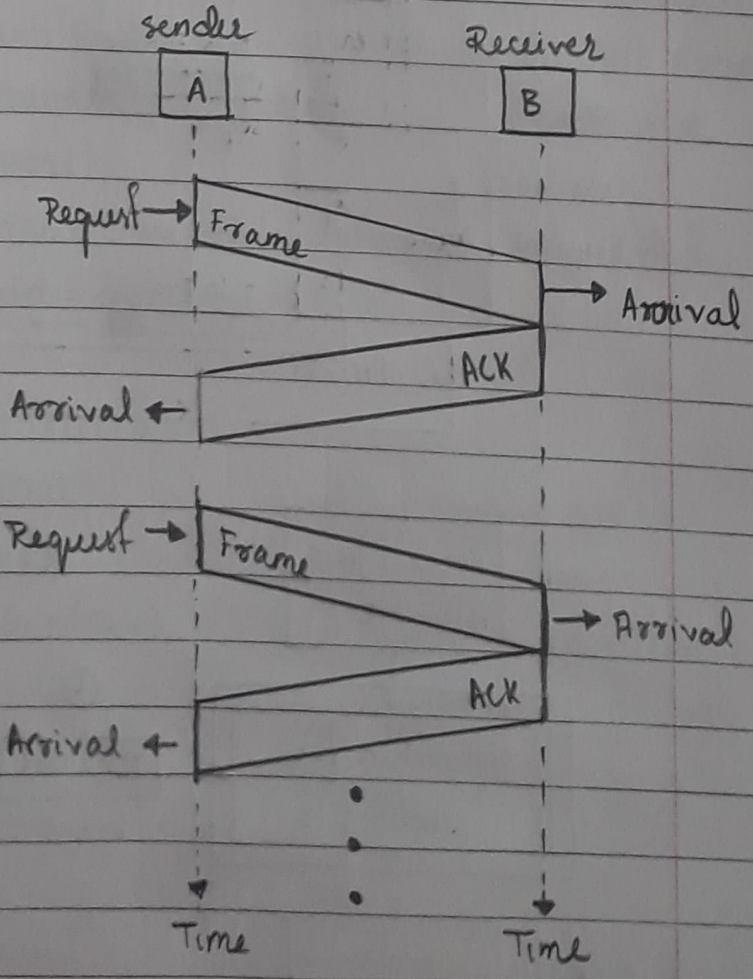
Algorithm for receiver side:

```
while (true)                           // Repeat forever
{
    waitForEvent();                    // sleeps until an event occurs
    if (Event (ArrivalNotification)); // Data frame arrives
    {
        ReceiverFrame();
        ExtractData();
        DeliverData();                 // Deliver data to network layer
        SendFrame();                   // send an ACK frame
    }
}
```

Example:

Communication using stop and wait protocol. The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame.

Here the sending two frames in the protocol involves the sender in four events and the receiver in two events.

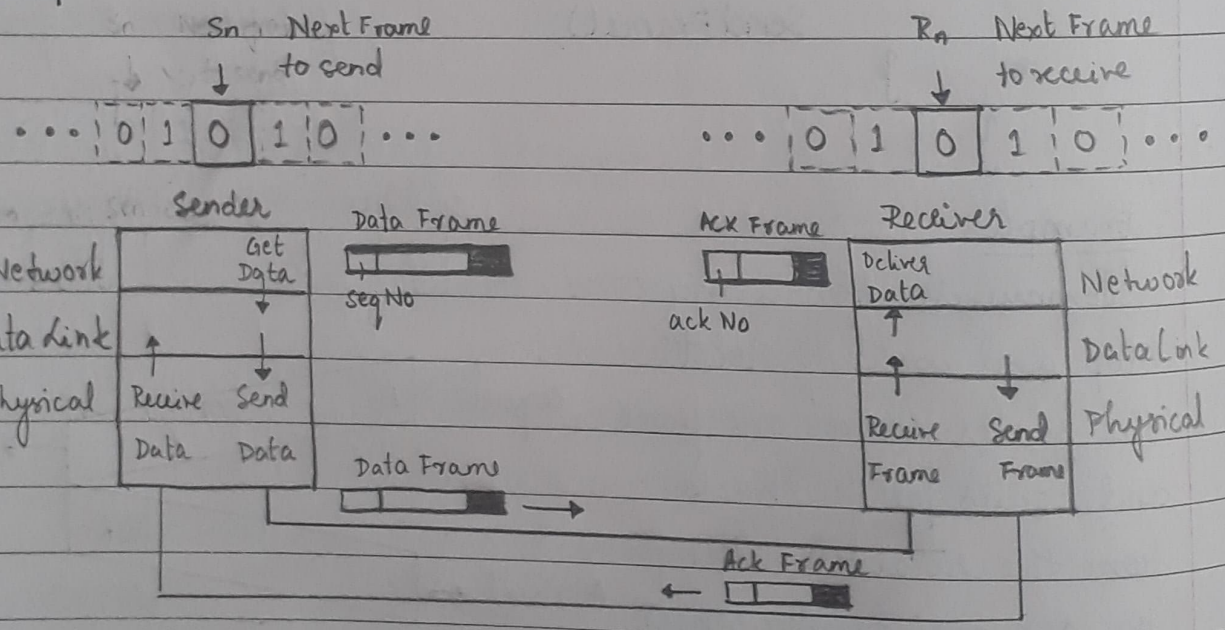


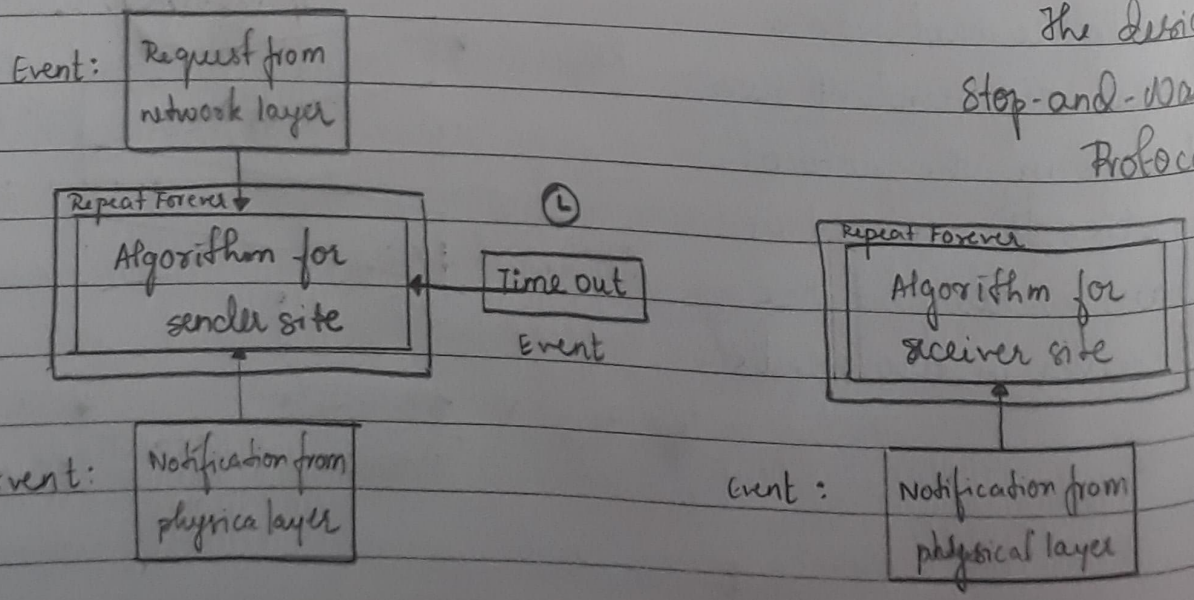
★ **Noisy Channels:**

Noiseless channels are non existent and thus we need to add error control to our protocols.

– **Stop-and-Wait Automatic Repeat Request**

- a simple error control mechanism is added to stop-and-wait protocol.
- error correction is done by keeping a copy of the sent frame and retransmitting the frame when the timer expires.
- we use sequence numbers to number the frames. The sequence numbers are based on modulo-2 arithmetic
- the acknowledgment number always announces in modulo-2 arithmetic of the sequence number of the next frame expected.





The design for Stop-and-Wait ARQ Protocol

Algorithm for sender site:

```
Sn = 0;                                  // Frame 0 should be sent first
canSend = true;                          // Allow the first request to go
while (true)                             // Repeat for ever
{   WaitForEvent();                      // sleep until an event occurs
    if (Event (RequestToSend) AND canSend)
    {   GetData();
        MakeFrame (Sn);                  // The seqNo is Sn
        StoreFrame (Sn);                 // keep copy
        SendFrame (Sn);
        startTimer ();
        Sn = Sn + 1;
        canSend = false;
    }
    WaitForEvent();                      // sleep
    if (Event (ArrivalNotification))     // An ACK has arrived
    {   ReceiverFrame (ackNo);           // Receive the ACK Frame
        if (not corrupted AND ackNO == Sn) // valid ACK
        {   Stoptimer();
            PurgeFrame (Sn-1);           // copy is not needed
            canSend = true;
        }
    }
    if (Event (TimeOut)                  // The timer expired
    {   startTimer();
        ResendFrame (Sn-1);              // Resend a copy check
    }
}
```

Algorithm for Receiver Site:

```
Rn = 0;                          // Frame 0 expected to arrive first
while (true)
{  WaitForEvent();               // sleep until an event occurs
   if (Event (Arrival Notification))  // Data frame arrives
   {  ReceiveFrame();
      if (corrupted (frame))
         sleep;
      if (seqNo == Rn)           // valid data frame
      {  ExtractData();
         DeliverData();          // Deliver data
         Rn = Rn+1;
      }
      SendFrame (Rn);            // Send an ACK
   }
}
```

Example:

Q1: Assume that in a stop-and-wait ARQ system, the bandwidth of the line is 1Mbps and 1 bit takes 20 ms to make a round trip. What is the bandwidth delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

— The bandwidth delay product is

$$= 1 \times 10^6 \times 20 \times 10^{-3} = 20{,}000 \text{ bits} //$$

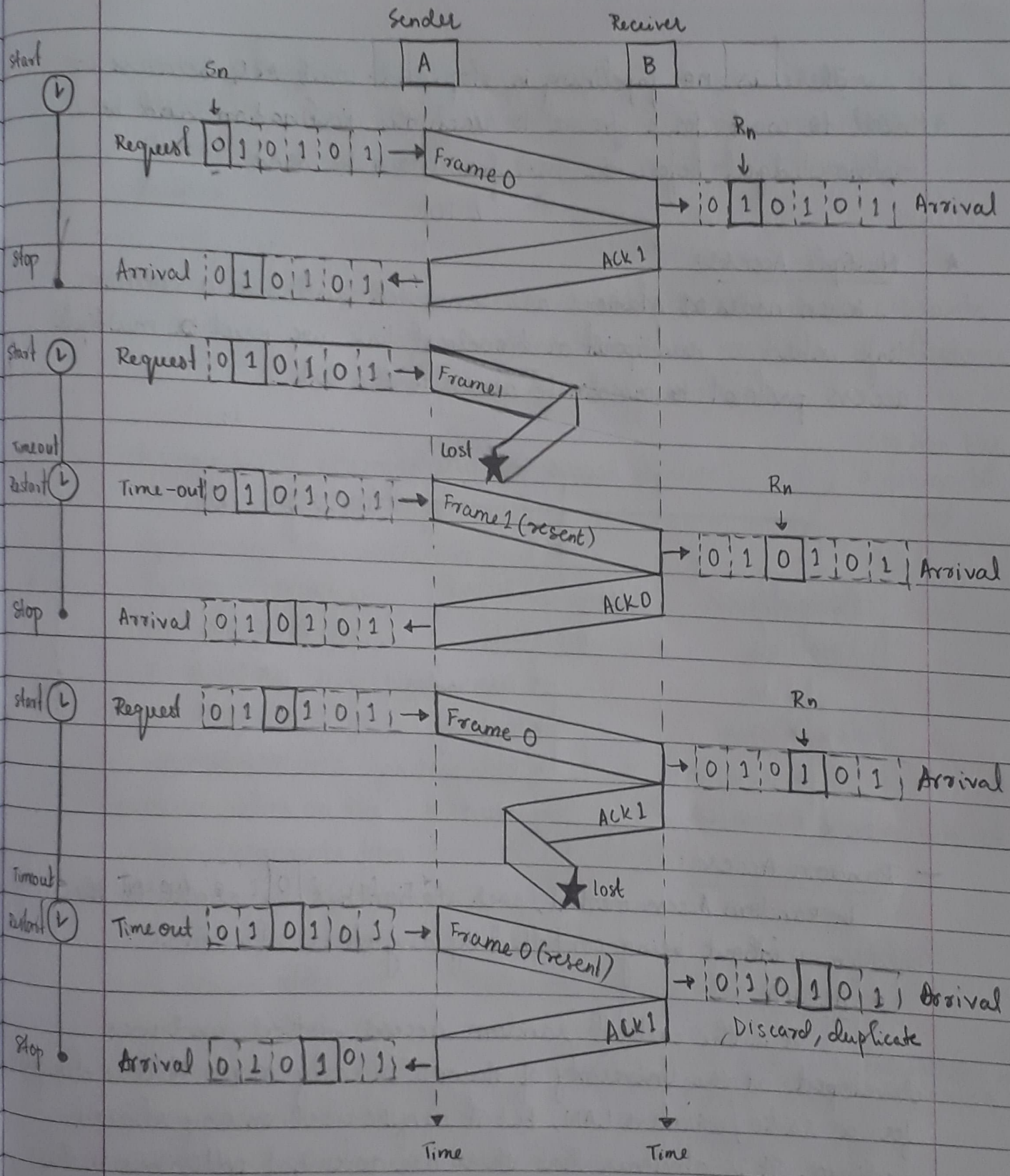
Utilization percentage of the link is

$$= \frac{1000}{20{,}000} = 5\% //$$

Q2: What is the utilization percentage of the link in the above question if we have a protocol that can send up to 15 frames before stopping and worrying about the acknowledgments?

— 15 frames or 15,000 bits during a roundtrip.

$$\therefore \text{Utilization} = \frac{15{,}000}{20{,}000} = 75\% //$$



Note:

The stop-and-wait ARQ protocol is inefficient if our channel is thick and long.

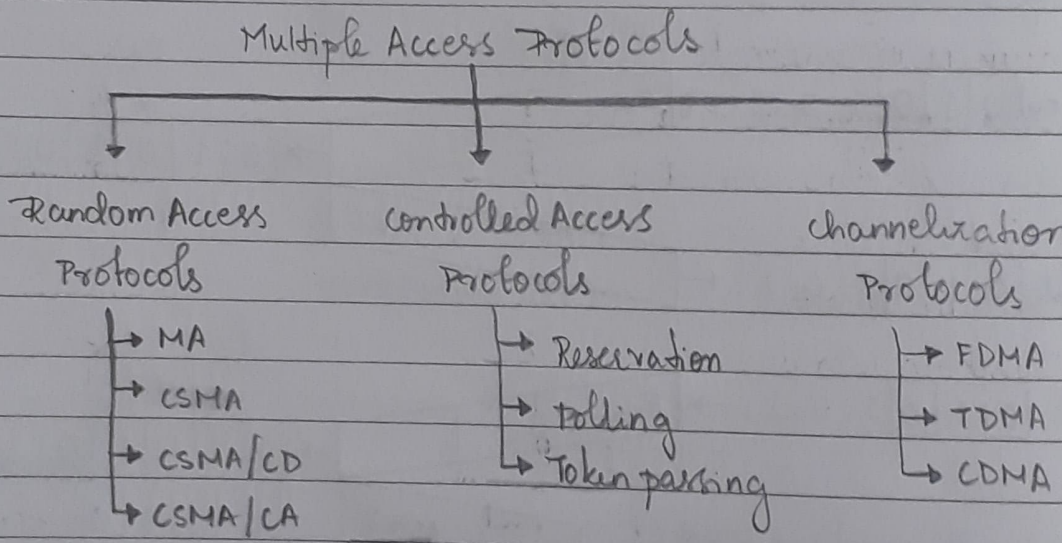
thick : large bandwidth long : round trip delay is long

The product of bandwidth and delay is called the bandwidth delay product.

There is no pipelining in stop-and-wait ARQ because we need to wait for a frame to reach the destination and be acknowledged before the next frame can be sent.

★ Multiple Access:

When nodes or stations are connected and use a common link called a multipoint or broadcast link we need a multiple access protocol to coordinate access to the link.

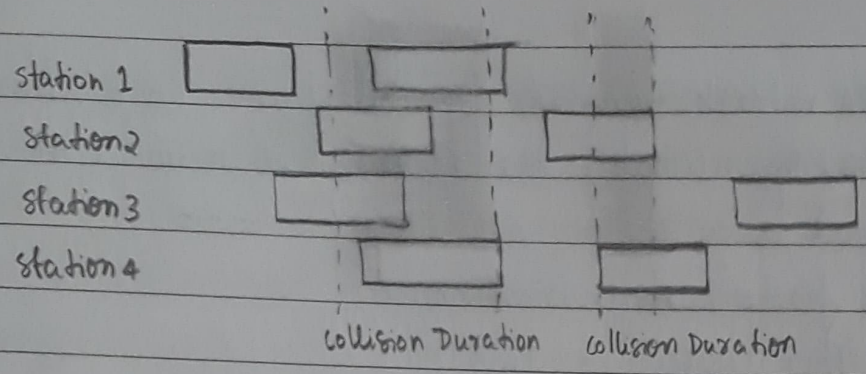


– Random Access:

In Random Access method, each station has the right to the medium without being controlled by any other station.

• ALOHA

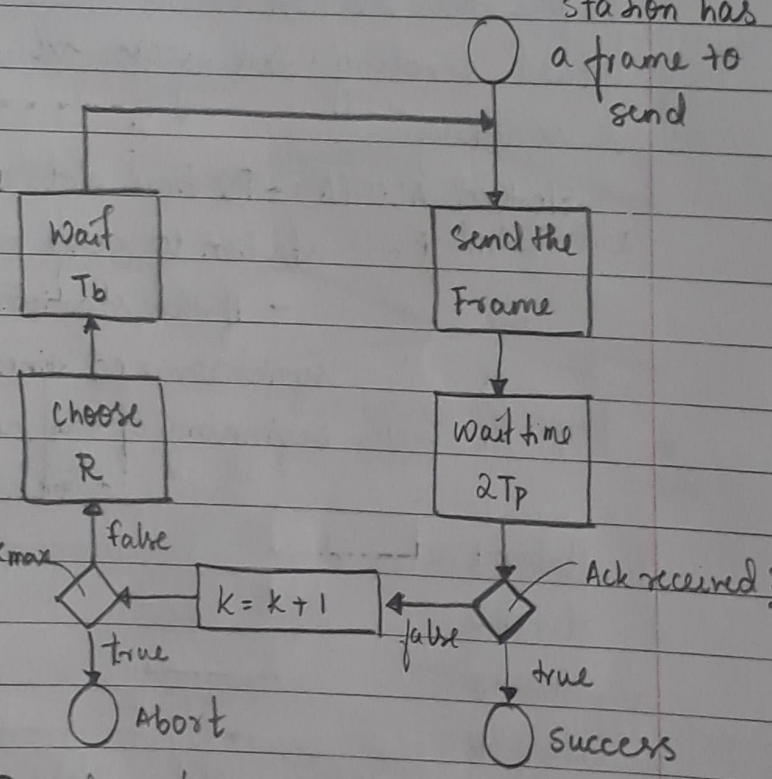
It is the earliest random access method, and was developed at the University of Hawaii in early 1970. It was designed for a radio/wireless LAN, but it can be used on any shared medium. It is obvious that there are potential collisions in this arrangement. The medium is shared between the stations. When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled.



Frames in a pure ALOHA network

Pure ALOHA: - Frames are sent whevere each stations wants.
- Only one channel to share: possibility of collisions between frames from different stations.

k: number of attempts
T_p: Max propagation time
T_{fr}: average transmission time
T_B: Back-off time
$R \times T_p$ or R_{fr}
R: Random number
0 to 2^{k-1}

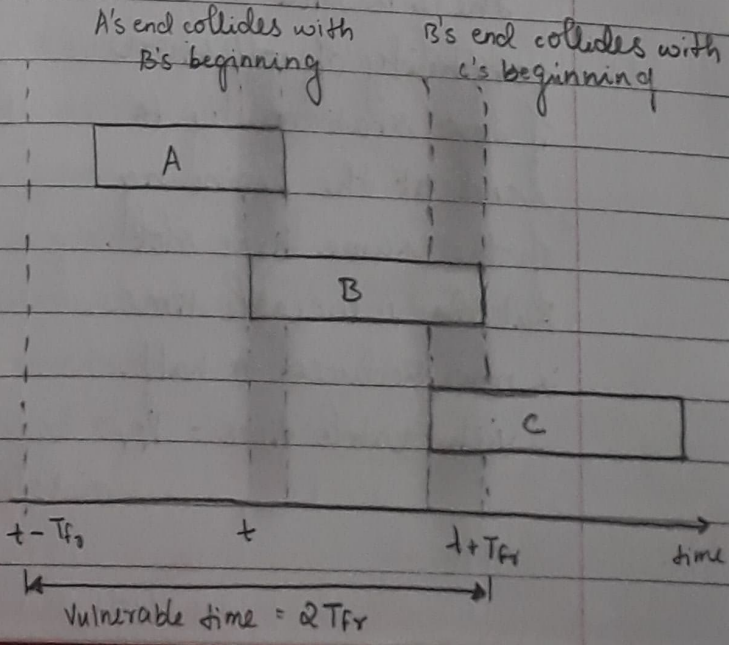


Procedure for pure ALOHA Protocol

The pure ALOHA protocol relies on the acknowledgments from the receiver. If the acknowledgment does not arrive after a timeout period, the station assumes that the frame has been destroyed and resends the frame.

Vulnerable time is where there is possibility of collision.

Vulnerable time = $2T_{fr}$



Example:

Q3: A pure ALOHA network transmits 200 bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

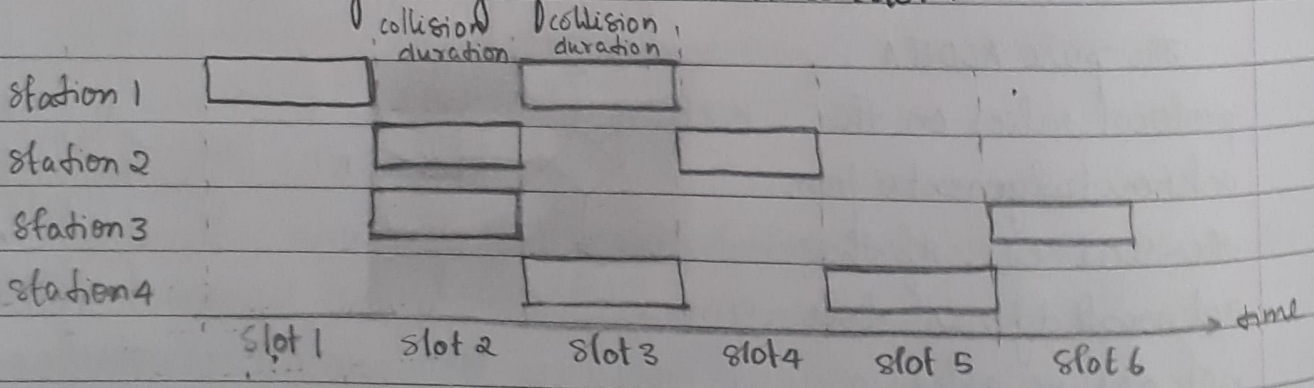
— Average frame transmission time

$$T_{fr} = \frac{200}{200 \times 10^3} = 1\text{ms}$$

Vulnerability time = $2T_{fr} = 2(1\text{m}) = 2\text{ms}$

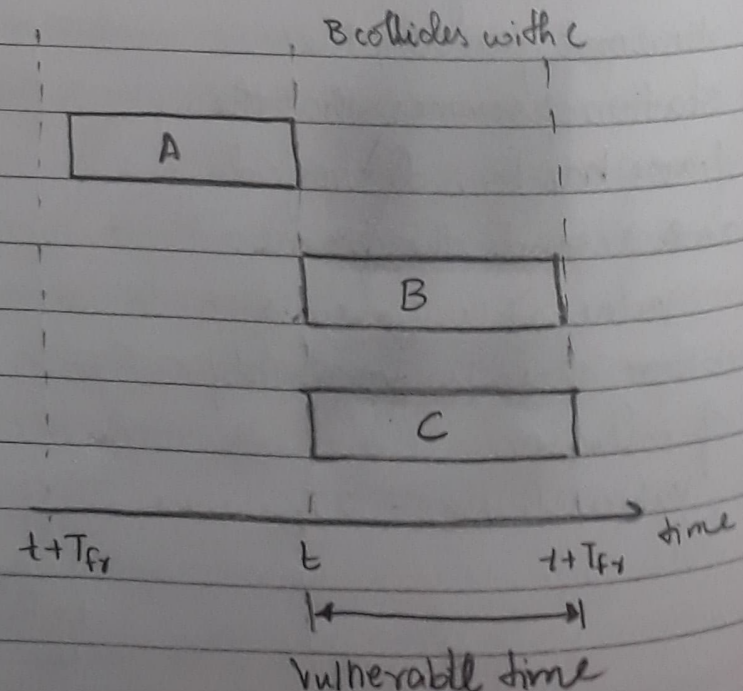
This means that no station should send later than 1ms before this station starts transmissions and no station should start sending during the one 1ms period that this station is sending.

Slotted ALOHA :- The time slots of T_{fr} are created and force the station to send only at the beginning of the time slot
- If the station misses the beginning of the synchronized time slot, it has to wait until the beginning of the next time slot.



There is still a possibility of collision if two stations try to send at the beginning of the same time slot. But the vulnerable time is now reduced to half.

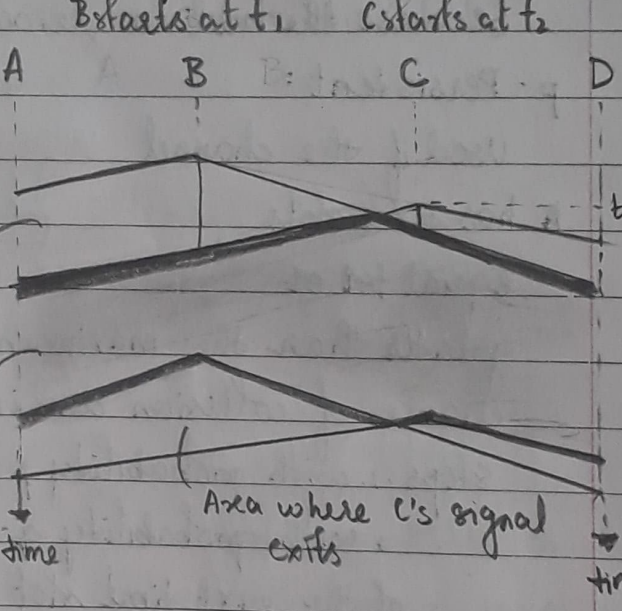
∴ Vulnerable time = T_{fr}



- CSMA: carrier Sense Multiple Access

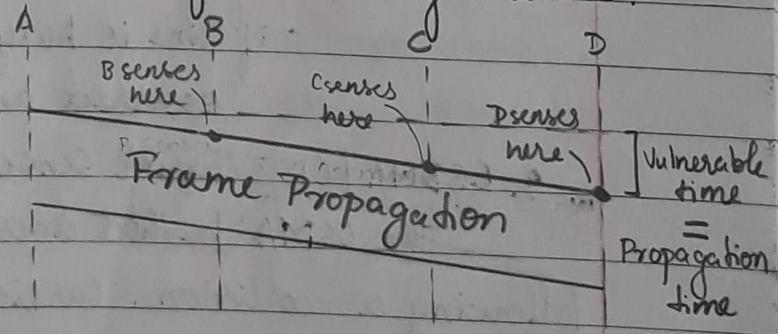
To minimize the chance of collision and, therefore increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. CSMA requires each station to first listen to the medium (check the state of medium) before sending. It is based on the principle "sense before transmit" or "listen before talk".

A station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received due to propagation delay.



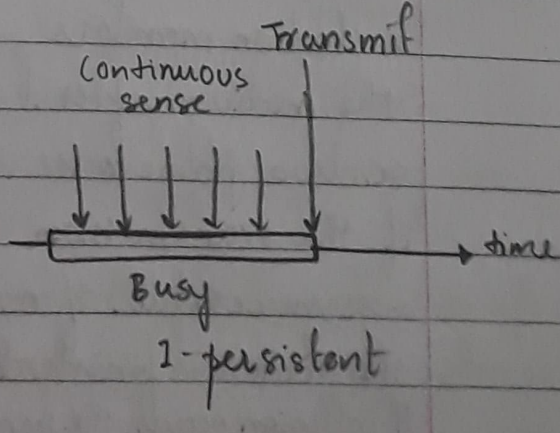
Here both, B and C frames are destroyed as they collide.

The vulnerable time for CSMA is the propagation time T_p. This is the time needed for a signal to propagate from one end of the medium to the other.

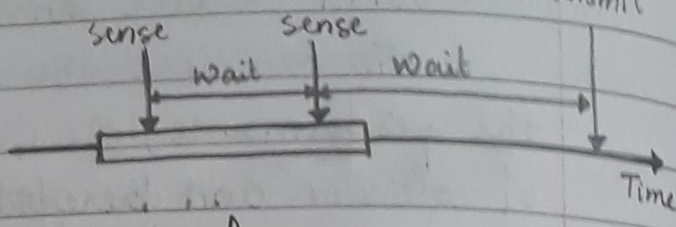


- Behaviors of three persistence methods:

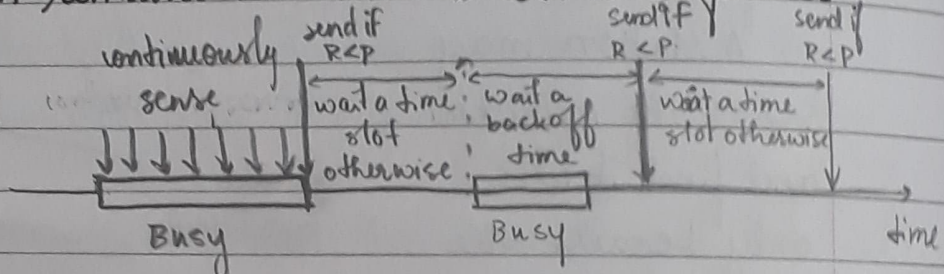
1-persistent: after the station finds the line idle, it sends its frame immediately. It has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.



Nonpersistent: if the line is idle, it sends immediately; if not it will wait for a random amount of time and then sense the line again. It reduces the chance of collision because it is unlikely two or more station will wait for the same amount of time and retry to send simultaneously. But efficiency is reduced because the medium remains idle when it can send frames.



p-Persistent: Used if the channel has time slots equal to or greater than the maximum propagation time. It reduces the chance of collision and improves efficiency.

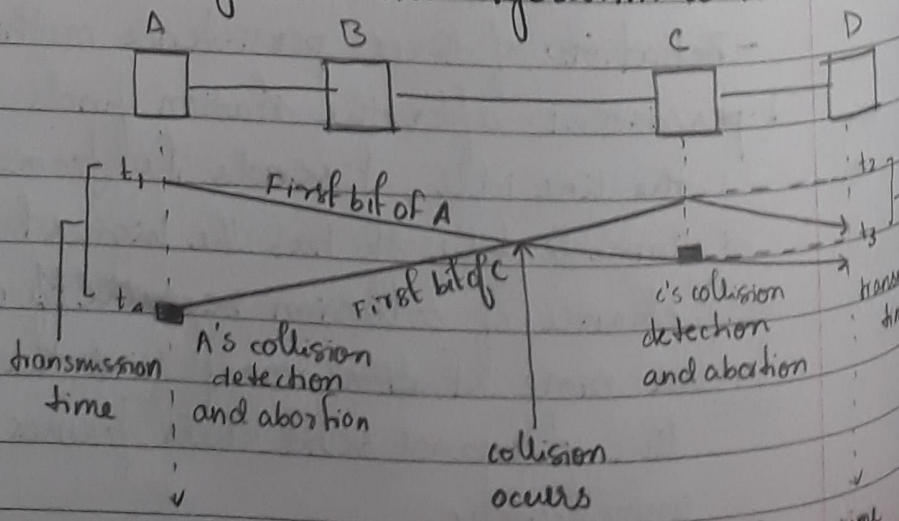


steps: 1. with probability p, the station sends its frame
2. with probability q=1-p, the station waits for the beginning of the next time slot and checks the line again
   - if line idle, it goes to step 1
   - if line is busy, it acts as though a collision has occurred and uses the backoff procedure.

- CSMA/CD: Carrier Sense Multiple Access with Collision Detection

The CSMA method does not specify the procedure following a collision. Carrier sense Multiple Access with collision detection (CSMA/CD) augments the algorithm to handle the collision.

In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If collision occurs, the frame is resent.



Collision of first bit in CSMA/CD

t_1: A starts sending the bits of its frame

t_2: C does not sense the first bit sent by A and starts sending its frame in both left and right direction.

t_3: collision occurs, then detects the collision on receiving the first bit of A's frame. Thus C immediately aborts transmission.

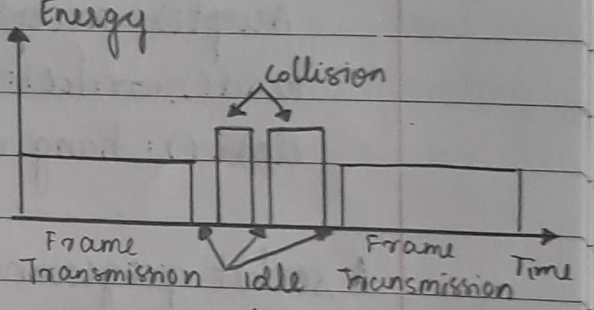
t_4: A detects collision on receiving the first bit of C's frame and immediately aborts transmission.

t_1-t_4: transmission time of A

t_2-t_3: transmission time of B

Energy level during transmission, idleness or collision

- At zero level: the channel is idle
- At normal level: a station has successfully captured the channel and sending its frame
- At abnormal level: there is a collision and level of energy is twice the normal level.

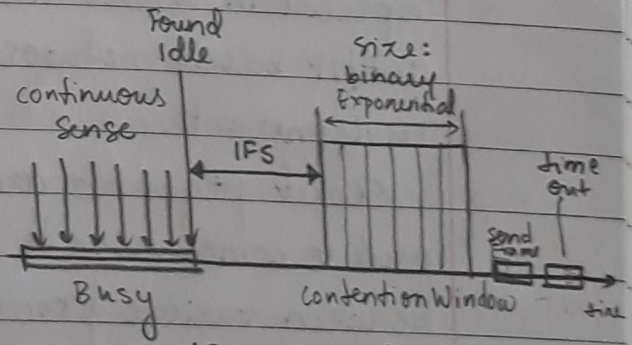


- CSMA/CA: carrier sense Multiple Access with collision Avoidance

carrier sense multiple access with collision avoidance (CSMA/CA) was invented for wireless networks, collisions are avoided through the use of CSMA/CA's three strategies:

- Interframe Space (IFS):

when an idle channel is found, the station does not send immediately as a distant station may have started transmitting. this wait period of time is called the interframe space. Even after IFS if the channel is idle then the station can send after waiting for a time equal to contention time.



- contention window: if the station finds the channel busy after IFS, it does not restart the timer, it stops the timer and restarts

it when the channel becomes idle.

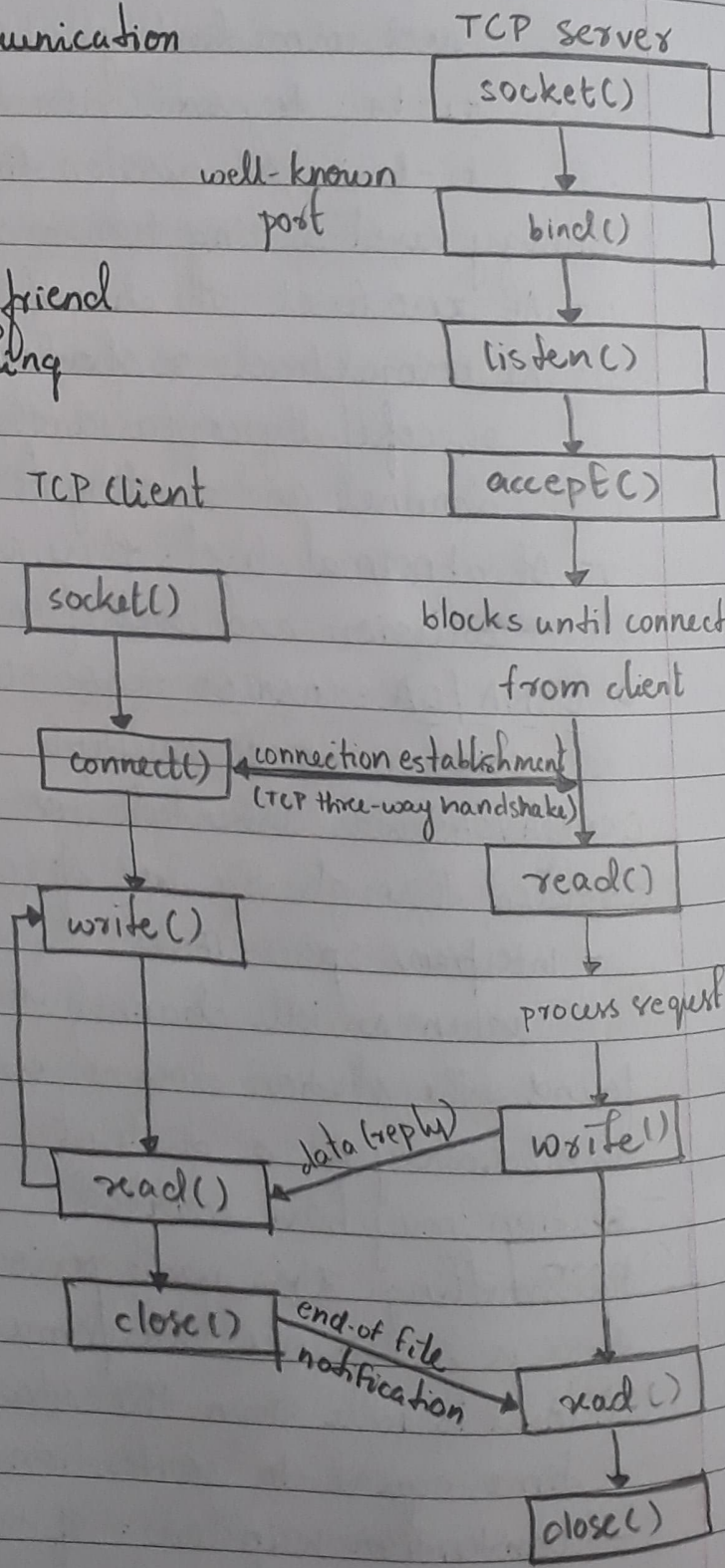
- Acknowledgement: Beside all the precautions collision may occur or data may be corrupted. Thus a positive acknowledgement and the time-out timer can guarantee that the receiver has received the frame.

* **Socket Programming:**

socket(): Endpoint for communication
Bind(): Assign a unique ID
listen(): wait for a chat
Connect(): Initialize chat
Accept(): accept chat from friend
read(), write(): start chatting
close(): hang up.

A socket programming interface provides the routines required for interprocess communication between applications, either on the local system or spread in a distributed, TCP/IP based network environment.

Socket programs are used to communicate between various processes usually running on different systems. It is mostly used to create a client-server environment.



UNIT - 3

Network Layer: Address Mapping Delivery and LANs

* **Address Mapping:**

A packet starting from a source host may pass through several different physical networks before finally reaching the destination host. The hosts and routers are recognized at the network level by their logical (IP) addresses and at the physical level by their physical addresses.

This means that delivery of a packet to a host or a router requires two levels of addressing: logical and physical. We need to be able to map a logical address to its corresponding physical address and vice versa. These can be done by using either static or dynamic mapping.

NOTE:

Physical Address:

- local address, unique locally but not necessarily unique universally. Ex: MAC.

Static Mapping:

- Involves the creation of a table that associates a logical address with a physical address. This table is stored in each machine on the network.

Ex: each machine that knows the IP address of another machine does not know its physical address but it can look it up in the table.

Dynamic Mapping:

- Each time a machine knows one of the two addresses, it can use a protocol to find the other one.

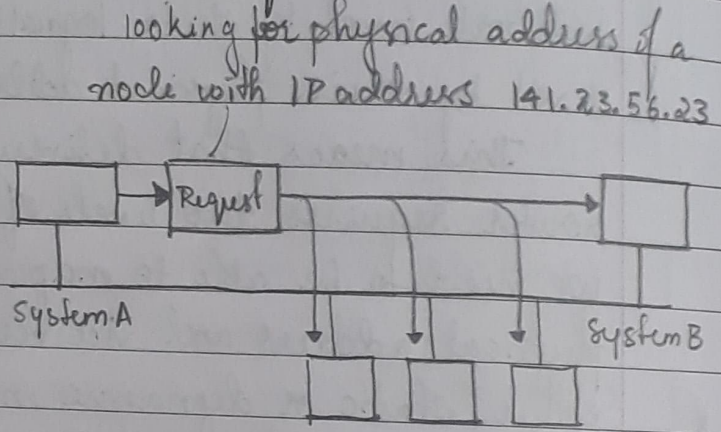
* Mapping Logical to Physical Address: ARP

• Address Resolution Protocol:

ARP protocol is used to map IP address to MAC address

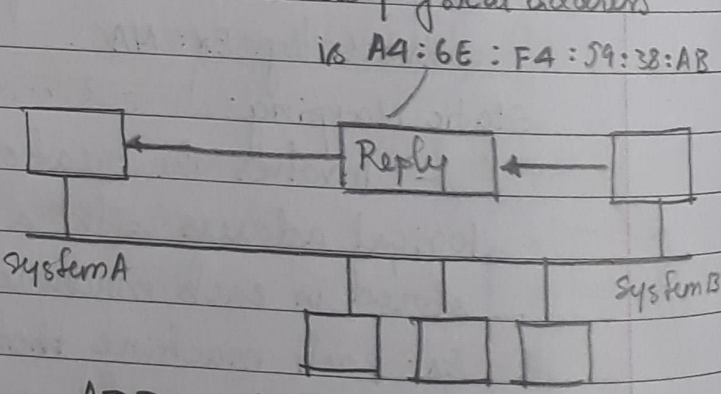
Anytime a host or a router has an IP datagram to send to another host or router it has the logical (IP) of the receiver. But the IP datagram has to be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver.

The host or the router sends an ARP query packet which includes the physical and IP addresses of the sender and IP address of the receiver. As the physical address of the receiver is unknown the query is broadcast over the network.



ARP request is broadcast

Every host or router on the network receives the ARP query packet but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and physical addresses.



ARP reply is unicast

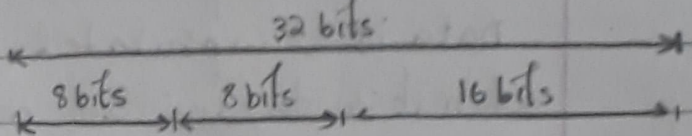
The packet is unicast and is directly sent to the inquirer by using the physical address received in the query packet

- Cache Memory:

Using ARP is inefficient if system A needs to broadcast an ARP request for each IP packet it needs to send to system B, ARP could instead broadcast the IP packet itself.

ARP can be useful if the ARP reply is cached because a system normally sends several packets to the same destination. A system that receives ARP reply stores the mapping in cache memory and keep it for 20 to 30 minutes unless there is no space in the cache. Before sending an ARP request, the system first checks its cache to see if it can find the mapping.

- ARP Packet Format:



Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender Hardware Address (Ex: 6 bytes for Ethernet)		
Sender Protocol Address (Ex: 4 bytes for IP)		
Target Hardware Address (Ex: 6 bytes for Ethernet)		Not filled in a request
Target Protocol Address (Ex: 4 bytes for IP)		

ARP Packet Format

Hardware Type: 16 bit field defining the type of the network on which ARP is running. (Ethernet)

Protocol Type: 16 bit field defining the protocol

Hardware Length: 8 bit field defining the length of the physical address in bytes (Ethernet - 6)

Protocol Length: 8 bit field defining the length of the logical address in bytes.

Operation: 16-bit field defining the type of packet (request/reply)

Sender Hardware Address: variable length field defining the physical address of the sender

Sender Protocol Address: Variable length field defining the logical address of the sender

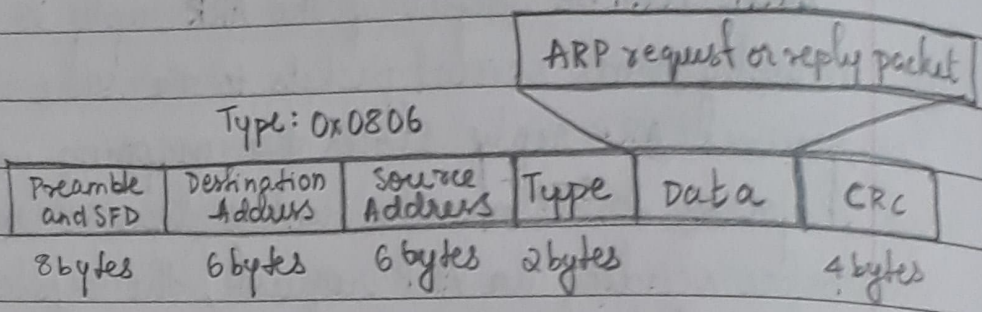
Target Hardware Address: variable length field defining the physical address of the target.

Target Protocol Address: variable length field defining the logical address of the target.

Keerthana Ashok

- Encapsulation:

An ARP packet is encapsulated directly into a data link frame.



Here it is encapsulated in an Ethernet frame.
Data is encapsulated with protocol information at each layer when it is transmitted across a network.

- Operation:

Steps on how ARP functions on a typical Internet.

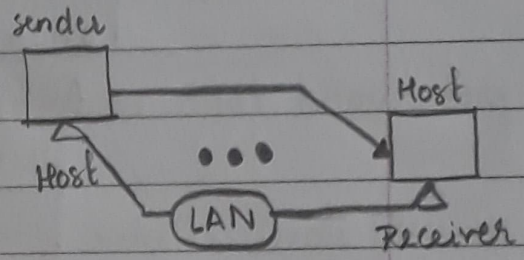
1. The sender knows the IP address of the target.
2. IP asks ARP to create an ARP request, filling in the sender's physical address, the sender's IP address and the target IP address. The target physical address is filled with 0's.
3. The message is passed to the data link layer where it is encapsulated in a frame by using the physical address of the sender as the source address and the physical broadcast address as the destination address.
4. Every host or router receives the frame as it contains a broadcast destination address. All machines except the target drop the packet as the target recognises its IP address.
5. The target machine replies with an ARP reply message that contains its physical address. The message is unicast
6. The sender receives the reply message. It knows the physical address of the target machine.
7. The IP datagram, which carries data from the target machine, is now encapsulated in a frame and is unicast to the destination.

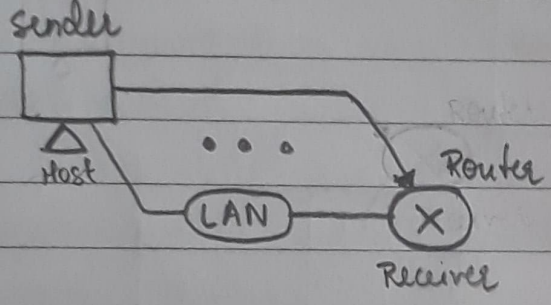
Keerthana Ashok

Four Cases using ARP:

CASE 1: A host has a packet to send to another host on the same network.

Target IP Address: Destination address in the IP datagram



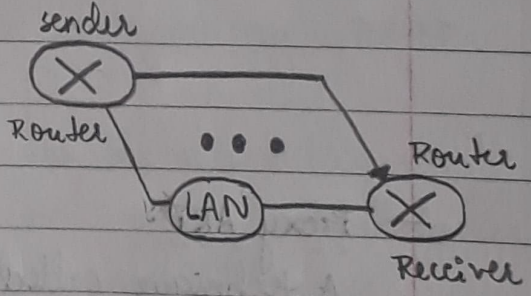


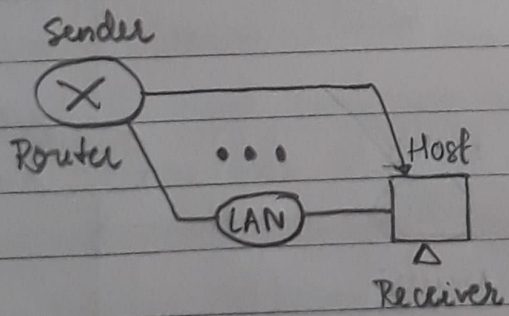
CASE 2: A host wants to send a packet to another host on another network. It must first be delivered to a router.

Target IP Address: IP address of a router.

CASE 3: A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.

Target IP Address: IP address of the appropriate router.



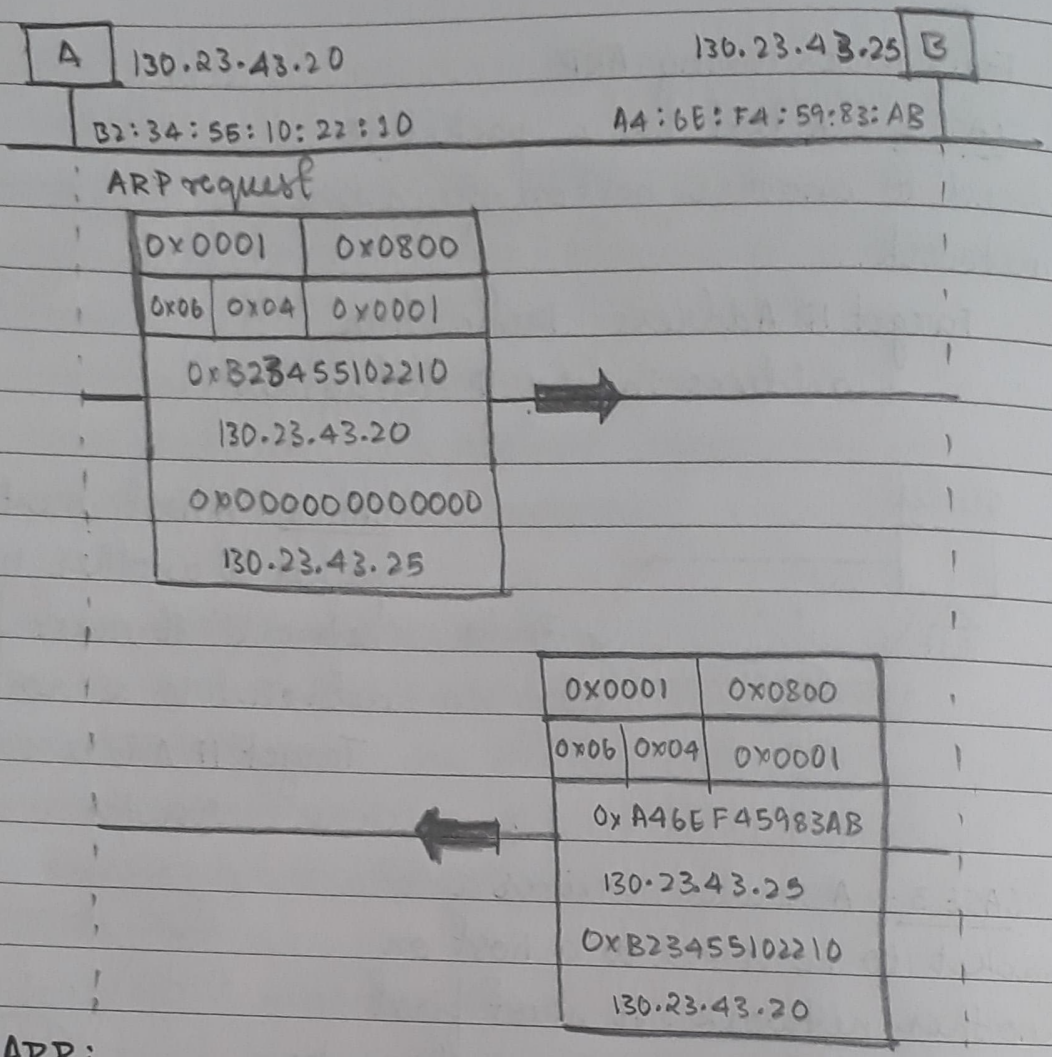


CASE 4: A router receives a packet to be sent to a host on the same network

Target IP Address: Destination address in the IP datagram.

Example 1

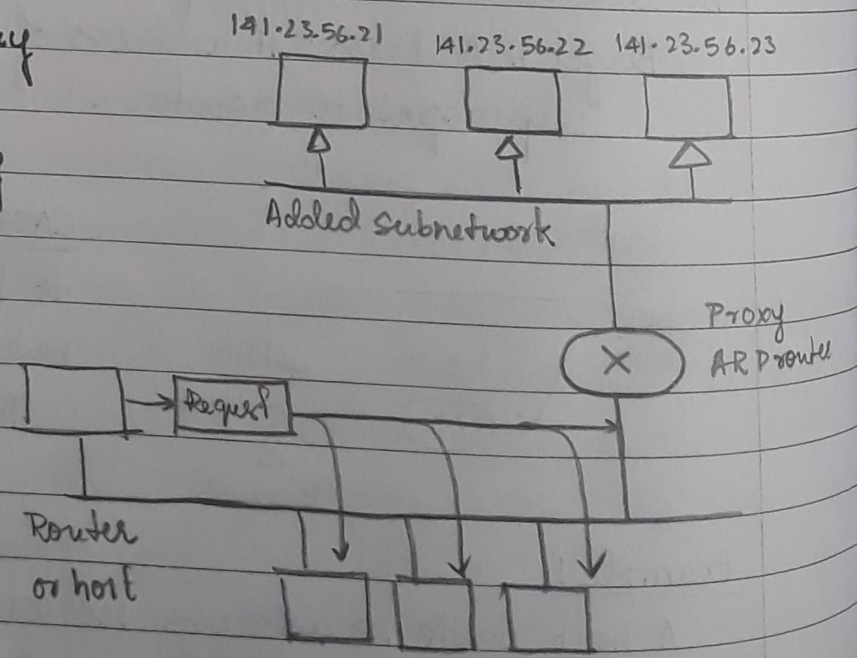
A host with IP address 130.23.43.20 and physical address B2:34:55:10:22:10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB (unknown for first host). The two hosts are on the same ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.



- Proxy ARP:

A technique called proxy ARP is used to create a subnetting effect. A proxy ARP acts on behalf of a set of hosts.

Whenever a router running a proxy ARP receives an ARP request looking for the IP address of one of these hosts, the router sends an ARP reply announcing its own physical address.



The proxy ARP router replies to any ARP request received for destinations: 141.23.56.21, 141.23.56.22 and 141.23.56.23

After the router receives the actual IP packet, it sends the packet to the appropriate host or router

* Mapping Physical to Logical Address : RARP and DHCP

• Reverse Address Resolution Protocol (RARP)

- It finds the logical address for a machine that knows only its physical address.
- To create an IP datagram, a host or a router needs to know its own IP address or addresses.
- The machine can get its physical address by reading its NIC and then use it to get the logical address by using RARP protocol.
- The RARP request is created and broadcast on the local network. Another machine on the local network that knows all the IP addresses will respond with a RARP reply.
- The requesting machine must be running a RARP client program; the responding machine must be running a RARP server program.
- Drawback: Broadcast is done at the data link layer. The physical broadcast address does not pass the boundaries of a network. This means that if an administrator has several networks or several subnets, it needs to assign a RARP server for each network or subnet. This makes RARP almost obsolete.

• Dynamic Host Configuration Protocol: (DHCP)

- It has been devised to provide static and dynamic address allocation that can be manual or automatic.
- Static Address Allocation: It is backward compatible with BOOTP which means a host running the BOOTP client can request a static address from a DHCP server which has a database that statically binds physical addresses to IP addresses.
- Dynamic Address Allocation: It has a second database with a pool of available IP addresses which makes it dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available IP addresses and

assigns an IP address for a negotiable period of time. When a DHCP client requests a DHCP server, it first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned. If the entry doesnot exist in the static database, the server selects an IP address from the available pool, assigns the address to the client and adds the entry to the dynamic database.

The dynamic aspect of DHCP is needed when a host moves from network to network or is connected and disconnected from a network. DHCP provides temporary IP addresses for a limited time.

The address assigned from the pool is temporary. The DHCP server issues a lease for a specific time and when the lease expires, the client must either stop using the IP address or renew the lease. The server has the option to agree or disagree with the renewal and if the server disagrees, the client stops using the address.

- Manual and Automatic Configuration: DHCP allows both manual and automatic configurations. Static addresses are created manually where as the dynamic addresses are created automatically.