

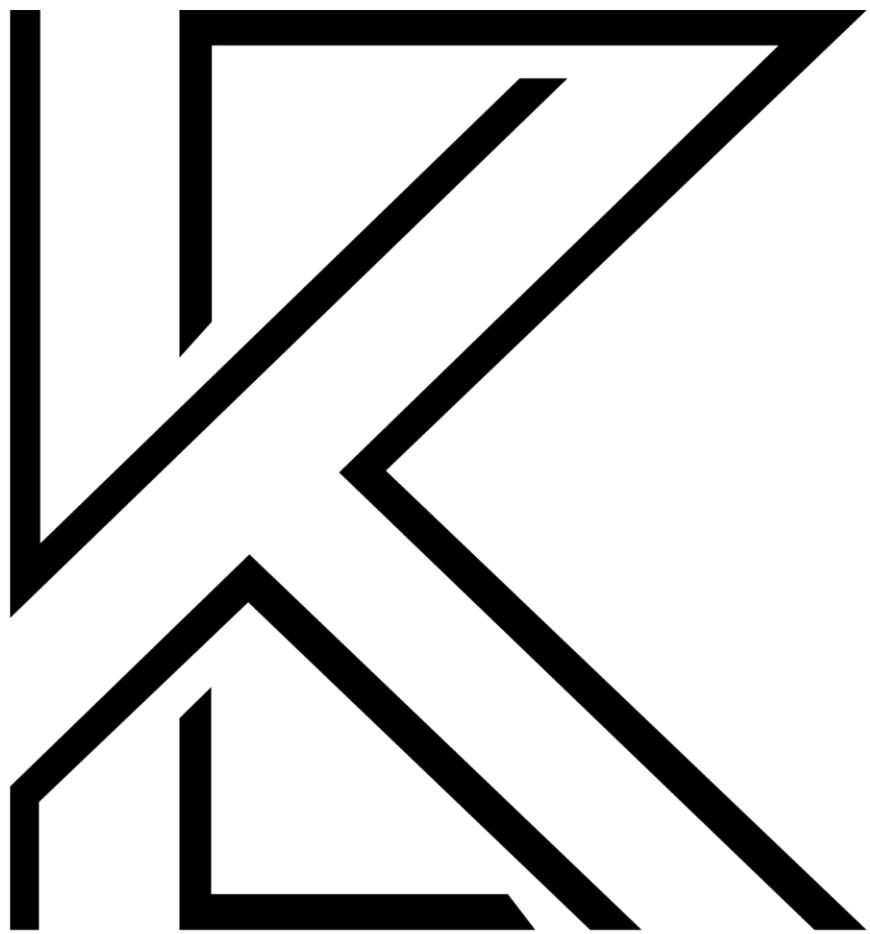
Computer communications and Networks

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا ﴿٥﴾

إِنَّ مَعَ الْعُسْرِ يُسْرًا ﴿٦﴾

*For indeed, with hardship comes ease.
Indeed, with hardship comes ease.*

[Surah Iushiraah Ayah 5-6]



Remember in your prayers.

INDEX

Section A:

Introduction

Internet ,Intranet and protocol	5
Human Communication vs network protocol.	6
Network Structure	7
Packet Switching-Circuit switching	10
Losses,Throughput and Traceroute	12
The OSI/ISO Model	15
CT 1 Question Solved	19
Application Layer	24
Processes	25
Web and HTTP	26
Cookies	29
E-mail	31
DNS	33
P2P	36
Transport Layer	38
TCP –UDP	39
rdt	46
Pipelining	52

Section B:

Network Layer	54
IPv4	56
IPv6	61
DHCP	63
NAT	64
Data Link Layer	65
MAC protocols	65
Slotted ALOHA	66
ARP	68
Networking Devices	69
PyQ's	73
Network Security	79
Cryptography	80
DES	81
CBC	82
RSA	83
#ash Function	86
Digital Certificates	86
Firewalls	87

Internet:

- The internet is a computer network that interconnects hundreds of millions of computing devices throughout the world.
- Internet can also be defined as -
 - 'an infrastructure that provides services to applications'
 - 'The internet is a vast network that connects computers all over the world' set of nodes connected by communication links.
- Through the internet people can share the information and from anywhere with an internet connection.

Intranet:

- An intranet is a private network contained within an enterprise that is used to securely share company information and computing resources among employees.

Intranet encourage communication within an organization. They let employees easily access important information, links, applications, forms and databases of company records.

Protocol:

- A protocol defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event.

A protocol is a set of rules that governs the communication between devices in a network. These rules define.

Format: The structure of the data being transmitted.

Order: The sequence in which messages are sent and received.

Actions: What the device do when they send, receive or encounter errors in messages.

Protocols are essential to ensure that devices from different manufacturers can communicate effectively over a network. They serve as the backbone of the internet and other computer networks by enabling reliable and efficient data exchange.

Significance of network protocols: Network protocols are vital for ensuring seamless communication between devices by defining rules for data exchange. They enable interoperability, reliability, and efficient data transfer across diverse devices and networks. Protocols ensure error detection, secure communication, and scalability, making them essential for the functioning of local and global networks like internet.

Human communication protocols VS Network protocols

Human protocols: Define rules for human communication, such as greetings, turn-taking, and responses, ensuring mutual understanding. Ensures orderly and effective human interaction.

Network protocols: Define rules for communication between devices in a network, including data format, transmission order and actions upon receiving messages.

Govern communication rules for network devices.

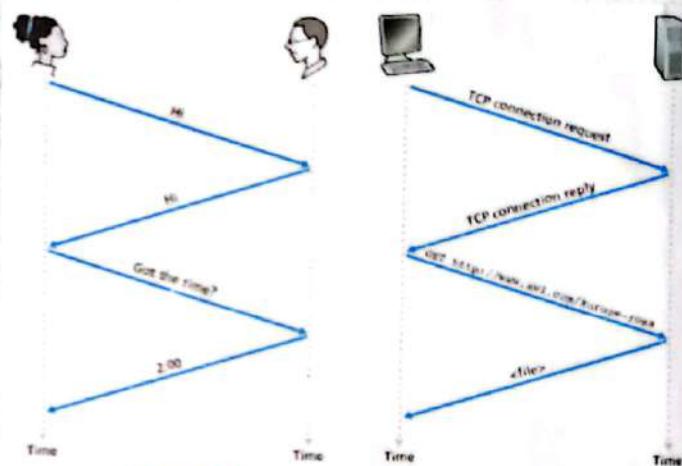


Figure 1.2 • A human protocol and a computer network protocol

Human Protocols	Network protocols
1. Entities are humans or groups.	1. Entities are computers or network devices.
2. Mediums are spoken language, gesture or text.	2. Mediums are wired or wireless digital networks.
3. Purpose is to enhance social interaction and knowledge sharing.	3. Purpose is the data exchange between devices.
4. Examples: saying Hello!, waiting for reply.	4. Examples: HTTP, TCP, SMTP, Ethernet protocols.

Network Structure:

The network structure is divided into:

The network edge: Mainly the devices that we use on a daily basis.

- Hosts: Devices at the edge of a network, including.

- Clients: End user devices like laptops, PC and smartphones.

- Servers: Devices providing services, often housed in data centers.

Access networks and physical media:

Connects hosts to the network core via communication links.

- wired: Ethernet cables, DSL, fibre optics.

- wireless: WiFi, cellular (4G, 5G), satellite communication.

Network core: The backbone of the network consisting of:

- Interconnected routers: Direct and forward data between networks.

Network of Networks: A global infrastructure enabling seamless communication. (e.g. the internet)

The core network is the part of the network that connects the different parts of the access network. It is a mesh of interconnected networks.

Write some network core devices?

Router, switches, load balancers, firewalls, NIC (Network Interface cards), MUX / Demux, etc.

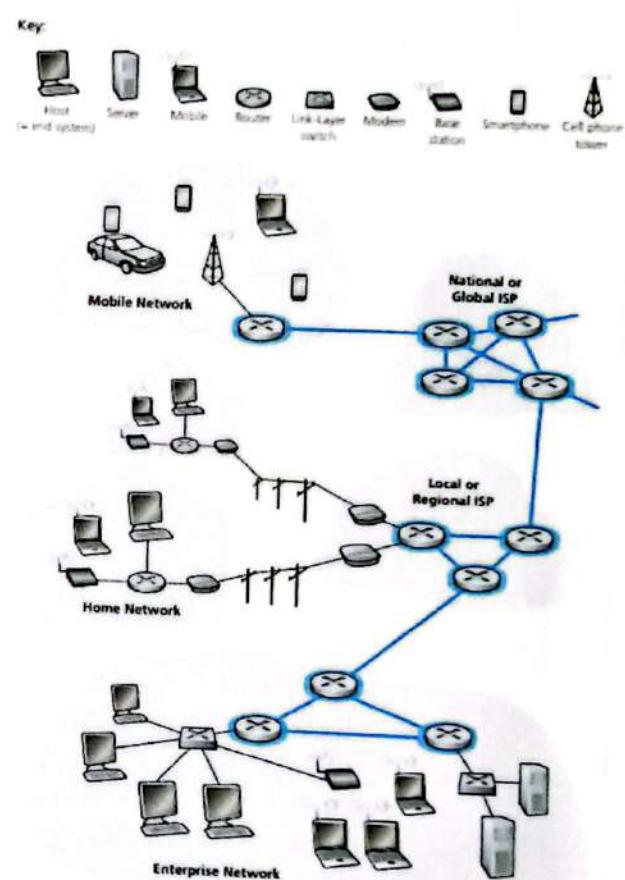
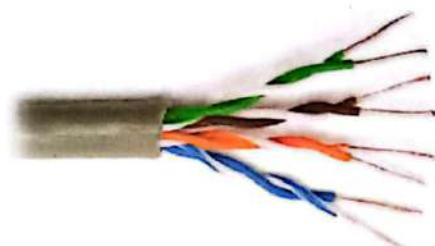


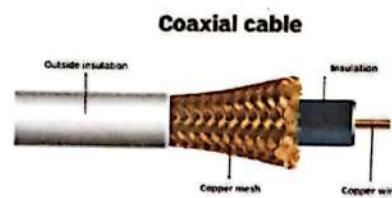
Figure 1.10 • The network core

Short Notes on Twisted pair, co-axial cable, fiber optic and Terrestrial Microwave link.

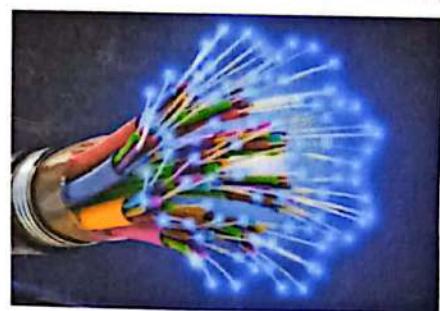
 Physical Media refers to tangible materials like twisted pair, copper wires, co-axial cables and fiber-optic cables that guide signal in a network.



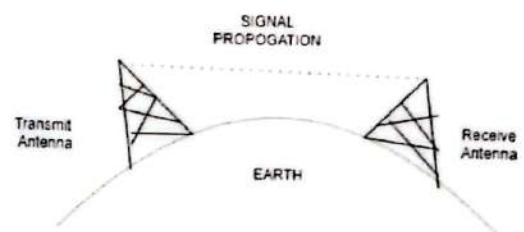
1. Twisted pair cable: Twisted pair cable consists of two insulated copper wires twisted together to reduce electrical interference. It is widely used in LANs and telephone networks. Data rates range from 10 Mbps to 10 Gbps depending on wire quality and distance, Category 6 cables enable high-speed connections for up to 100 meters.



2. Co-axial cable: Comprising two concentric copper conductors with shielding, co-axial cable supports high-speed connection data rates and shared communication. Commonly used in cable TV and internet systems, with digital signal transmitted in specific frequency bands.



3. Fiber optic: This medium transmits light pulses, offering bit rates upto hundreds of Gbps. Fiber optic is immune to electro-magnetic interference, has low attenuation over long distances, and is ideal for long-haul and backbone networks.



Wireless media uses electromagnetic waves, such as radio and satellite signals to transmit data through the air without physical connections.

4. Terrestrial Microwave Link: Operating in the radio spectrum, terrestrial microwave links provide wireless communication over short, local or wide areas. They offer mobility and long distance coverage but are subject to interference and fading.

□ Connecting End Systems to edge routers.

■ Residential Access Networks: Use DSL, cable modems, or fiber for home connections with bandwidth ranging from 10Mbps to 1Gbps. Typically shared access with other users in the area.

Institutional Access Networks: Use Ethernet or WiFi often linked to high speed fiber backbone with bandwidth up to 10 Gbps. Provides dedicated or shared access based on organizational (school/company) setup.

Mobile Access Networks: Utilize 4G/5G cellular networks with bandwidth of 10Mbps to over 1Gbps. Always shared, as multiple devices connect to the same base station.

□ Cable and Home Network.

■ Cable Network: A cable network uses Hybrid Fiber-Coaxial (HFC) technology to connect homes to the ISP. The network transmits data and TV signals over shared cables using Frequency Division Multiplexing - FDM. A splitter separates video signals and internet data, which are then sent through a cable modem to the home network. Signals travel to cable head end and are processed by a Cable Modem Termination System (CMTS). Offers up to 30Mbps downstream and 2Mbps upstream, but bandwidth is shared among users.

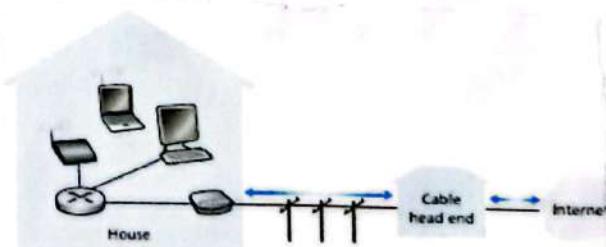


Figure 1.9 • A typical home network

Home Network: A Home network connects devices via a router with a firewall and NAT for security. Devices can connect through wired Ethernet (100Mbps - 1Gbps) or Wi-Fi. Internet access is provided by the cable or DSL-Digital subscriber line modem, which communicate with the central office or headend.

Packet switching:

Packet switching divides data into small packets, which are sent independently over a shared network. Each packet may take different paths and is reassembled at the destination.

Circuit switching:

Circuit switching establishes a dedicated communication path between two endpoints for the duration of the session, reserving resources exclusively for the connection.

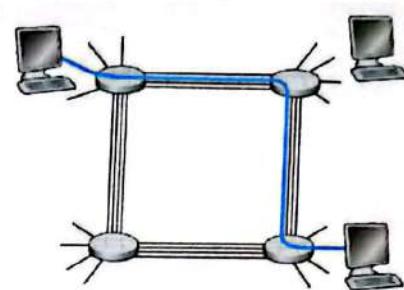


Figure 1.13 • A simple circuit-switched network consisting of four switches and four links

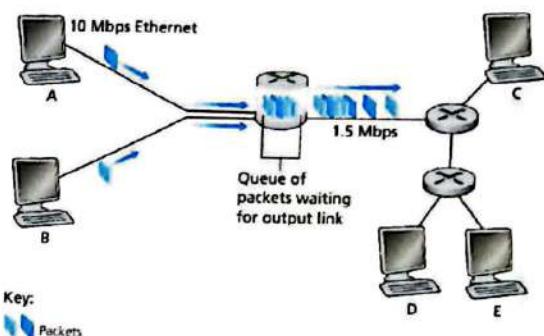


Figure 1.12 • Packet switching

Packet switching

1. No dedicated path in connection, data is sent in packets.
2. Efficiency is high, as bandwidth is shared dynamically.
3. Less reliable because packets can arrive out of order or be lost.
4. Scales better for large networking.
5. It supports more users with store and forward process.
6. It is implemented in network layer.
7. Less-reliable best effort service.
8. Examples: Internet (email, browsing)

Circuit switching

1. Requires a dedicated path for communication.
2. Efficiency is low, as resources are reserved even if idle.
3. Reliable because data is transmitted in order.
4. Limited scalability due to resource allocation.
5. Less users support.
6. It is implemented in Physical Layer.
7. More reliable but waste of resources.
8. Telephone Networks

■ How packet switching increases capacity?

■ In packet switching bandwidth is shared dynamically among users. For instance, in a 10Mbps link, if one user sends at 2Mbps and another at 3Mbps, the remaining 5Mbps is available for other users.

Unlike circuit switching, where each user gets a fixed allocation regardless of usage; packet switching re-allocates unused bandwidth in real time, maximizing network utilization.

Example: In VoIP, packets are sent only during speech, leaving capacity free for others during silence.

■ Delay and Types of Delay.

■ Delay refers to the time it takes for a packet to travel from the source to the destination across a network. It is the summation of various types of delays experienced at each node along the path.

Types:

1. Processing Delay: Time taken by a router to examine a packet's header, determine the appropriate path and check for bit level errors.
2. Queueing Delay: The time a packet spends waiting in the queue to be transmitted. Depends on traffic load, ranges from 11 to m sec.
3. Transmission Delay: Time to push all bits of a packet onto the link, calculated as,

$$Tx \text{ delay} = \frac{L}{R} \text{ sec}$$

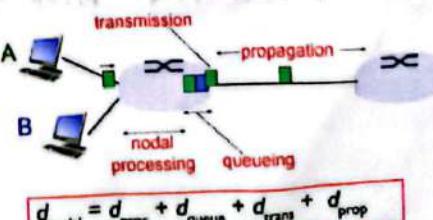
where

L = packet size(bits)

R = link rate (bit/sec)

4. Propagation Delay: Time for a signal to travel through the physical link to the next node. calculated as,

$$P_{delay} = \frac{d}{S} \text{ ms}$$



$$d_{\text{total}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

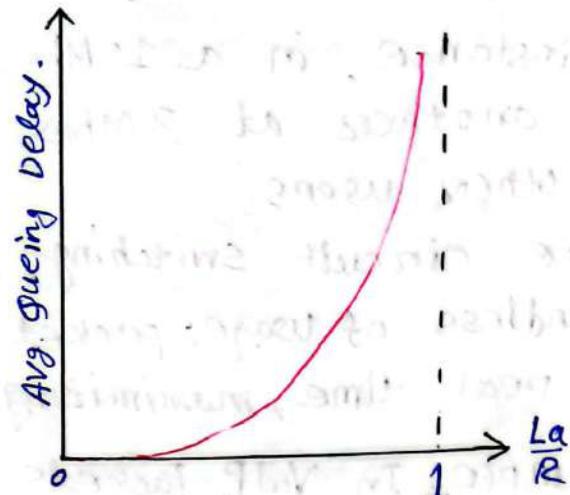
Losses

Losses in computer networks occur when packets arrive at a queue with finite capacity, and the queue is full. These packets are dropped, causing data loss. The likelihood of packet loss increases as the traffic intensity $\frac{La}{R}$ approaches or exceeds '1', where packets arrive faster than they can be processed. Proper system design ensures traffic intensity remains below '1' to minimize delays and loss.

$\frac{La}{R} \approx 0$ avg. queuing delay small.

$\frac{La}{R} < 1$ avg. queuing delay large.

$\frac{La}{R} > 1$ more work arriving than can be serviced, avg. delay infinite.

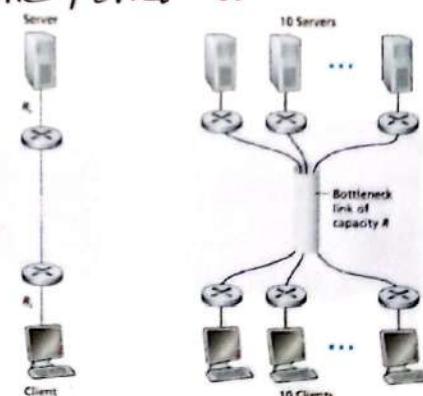


a = avg. packet arrival rate.

Throughput:

Throughput refers to the actual rate at which data is successfully transferred from one point to another in a network.

The rate at which bits are transferred between sender and receiver is called throughput?



Traceroute:

Traceroute is a diagnostic tool used to trace the path that packets take from a source to a destination in a network.

1. Packet sending: Traceroute sends specially crafted packets toward the destination. Each packet has a Time-to-Live (TTL) value starting from 1, which increments with each subsequent packet.

2. Router Behaviour: When a router receives a packet with TTL=1, it decrements TTL to 0, discards the packet and sends an ICMP message "Time Exceeded" back to the source, along with its address.

3. Source Records Information: The source records the time it took to receive the ICMP message and the address of the router.

4. Repeats for all Hops: The process repeats for each hop until the packet reaches the destination, which responds with an acknowledgment.

Effect of Link Bandwidth on Throughput and Maximum Throughput between sender and receiver.

The link bandwidth determines the maximum data rate a channel can support. Higher bandwidth allows more data to be transmitted per unit of time, increasing potential throughput.

However, the actual throughput is limited by factors like network congestion, packet loss and protocol overhead.

The maximum throughput between a sender and receiver is minimum of the bandwidth along the path and the receiver's processing capability. It is calculated as,

$$\text{Max Throughput} = \min(\text{BW}, \text{Reactor's Rate}).$$

Layering in Networking:

Layering in networking is the process of dividing the network design into distinct functional levels, called layers, where each layer handles specific tasks and provides services to the layer above while relying on the layer below. This approach simplifies complexity, ensures modularity and supports interoperability.

Why Layering is needed?

Layering reduces complexity: Layering organizes network functions into smaller manageable sections. Each layer performs specific tasks and uses services from the layer below it. This modular approach simplifies troubleshooting, upgrades and management.

Structural Responsibilities: It provides a structural way to discuss and implement system components. It separates the responsibilities amongst the network layers reducing duplication of efforts and errors.

What are the reasons for dividing Networks into Layers.

The reasons for dividing networks into layers are discussed below:

Abstraction: Layers abstract complex operations, allowing easier focus on specific functions.

Compatibility: Layering enables interoperability between hardware and software.

Error isolation: Faults in one layer don't affect others, aiding debugging.

Scalability: Networks can grow without major redesigns.

Flexibility: Layers can evolve independently, supporting innovation.

The Internet Protocol Stack:

The internet protocol stack is a five layer networking model used in the internet. It includes:

1. Application Layer: facilitates communication between network applications. Data unit is messages. The used protocols are HTTP, SMTP, FTP etc.

2. Transport Layer: Ensures reliable or unreliable delivery of application messages. Data unit is segments. Used protocol is TCP.

3. Network Layer: Routes packets between devices across networks. Data unit is datagrams. Used protocol is IP.

4. Link Layer: Transfers frames over a specific physical link between adjacent nodes. Data unit is frames. Protocols used are Ethernet, Wi-Fi.

5. Physical Layer: Transmits raw bits through physical media (e.g. copper, fiber). Data units are bits. Protocols depend on hardware and transmission media.

Application
Transport
Network
Link
Physical

a. Five-layer Internet protocol stack

The OSI/ISO Reference Model.

The open system Interconnection -OSI model is a conceptual framework that standardizes the functions of a communication systems into seven layers. Described below:

1. Application Layer: Interfaces with users enabling services like file transfer and email. protocols used in this layer are HTTP, SMTP, FTP. Data unit is messages. Devices are PC's.



2. Presentation Layer: Handles data formatting, encryption and compression. protocols used are SSL, JPEG. Devices are Application software and data unit is here also messages.

b. Seven-layer ISO OSI reference model

OSI Model	Function	TCP/IP Layers
7 APPLICATION	» End user layer » HTTP, FTP, RPC, SSH, DNS	APPLICATION
6 PRESENTATION	» Syntax layer » SSL, SSH, IMAP, FTP, MIME, XML, JPEG	
5 SESSION	» Synch & send to port » APIs, Sockets, WinSock	
4 TRANSPORT	» End-to-end connections » TCP, UDP	TRANSPORT
3 NETWORK	» Packets » IP, ICMP, IPSec, IGMP	NETWORK
2 DATA LINK	» Frames » Ethernet, PPP, Switch, Bridge	NETWORK INTERFACE
1 PHYSICAL	» Physical structure » Coax, Fiber, Wireless, Hubs, Repeaters	

3. Session Layer: Manages sessions and synchronization between the end users. protocols used are NetBIOS, RPE. Devices are computer servers. and data unit is still messages.

4. Transport Layer: Provides reliable data transfer with error checking capability. protocols used are TCP, UDP. Devices used are routers, firewalls. Here data unit is changed to segment.

5. Network Layer: Routes data packets between devices. protocols used are IP, ICMP, devices used are outers and switches. Data units are changed to packets.

6. Data Link Layer: Ensures reliable frame transfer between nodes. protocols used are Ethernet, Wi-Fi. Devices used in this layer are NICs, Bridges. Data units are frames.

7. Physical Layer: Transmits raw bits over physical media. protocols used are RS-232, IEEE 802.3. Devices are cables, hubs. Data units are bits.

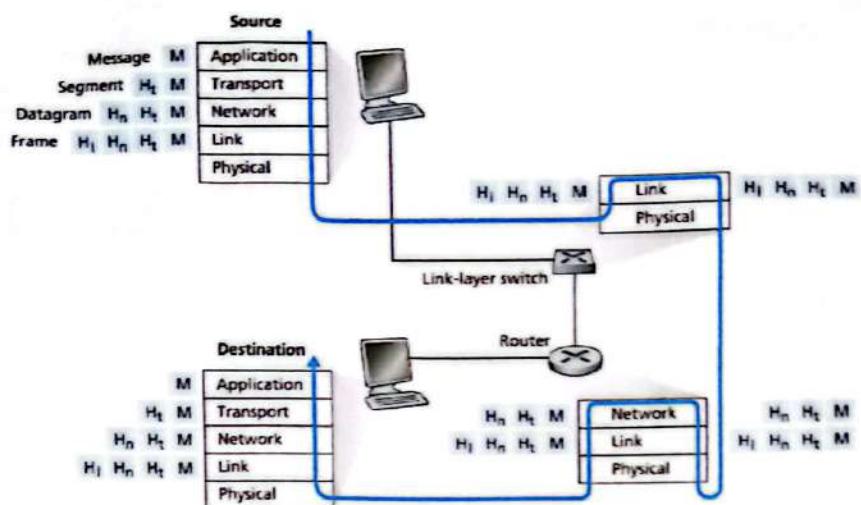


Figure 1.24 • Hosts, routers, and link-layer switches; each contains a different set of layers, reflecting their differences in functionality

■ Draw a block diagram of Internet structure in terms of networks of networks.

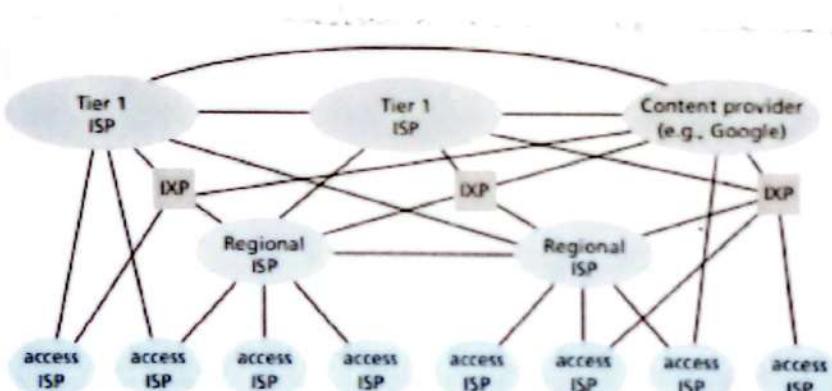


Figure 1.15 • Interconnection of ISPs

■ Write some Network core - Devices

■ Some Network core devices are:

1. Routers: Directs data packets between networks using IP addresses.
2. Switches: Connects multiple devices in a network and manage data transfer based on MAC addresses.
3. Firewalls: Monitor and control incoming/outgoing traffic to protect the network.
4. Gateways: Acts as translators between different network protocols or architectures.
5. Network Hubs: Basic devices for connecting multiple devices.
6. Wireless Access points (APs): Provide wireless connectivity within a network.

A 64 Kb file is transmitted through 1 Mbps link. If the nodal processing delay is negligible and the queuing delay is 50 ms. Then find the total delay, the length of the physical link is 100 km.

We have,

$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop.}}$$

NOW, Transmission delay,

$$\begin{aligned} d_{\text{trans}} &= \frac{L}{R} \\ &= \frac{64 \text{ kb}}{1 \text{ Mbps}} \\ &= \frac{64 \times 10^3}{10^6} \\ &= 64 \times 10^{-3} \text{ sec} \end{aligned}$$

Given,

File size, $L = 64 \text{ kb}$.

Tx. speed, $R = 1 \text{ Mbps}$.

Distance, $d = 100 \text{ km}$.

speed, $s = 2 \times 10^8 \text{ [Assumed]}$

Propagation Delay,

$$\begin{aligned} d_{\text{prop}} &= \frac{d}{s} \\ &= \frac{100 \times 10^3}{2 \times 10^8} \\ &= 50 \times 10^{-5} \text{ sec.} \\ d_{\text{queue}} &= 50 \times 10^{-6} \end{aligned}$$

Total delay.

$$\begin{aligned} d_{\text{nodal}} &= d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop.}} \\ &= 0 + 50 \times 10^{-6} + 64 \times 10^{-3} + 50 \times 10^{-5} \\ &= 0.06455 \text{ sec.} \end{aligned}$$

Ans.

Calculate the time required to send a file of 74072 Bytes from host 'J' to 'K' over any circuit switched network?

Assume all links are 1.635 Mbps.

Each link uses FDM with 27 channels frequencies.
534 ms to establish end to end circuit.

Each circuit has transmission rate,

$$\frac{1.635 \times 10^6 \text{ bps}}{27} = 60555.55556 \text{ bps}$$

$$= 60.55555 \text{ Kbps.}$$

so it takes. $\frac{74072 \times 8}{60.55555 \times 10^3} = 9.7856 \text{ sec}$ to transmit the file. To this we add 534 ms the establishment time, giving. $9.7856 + 0.534 = 10.3196 \text{ sec}$

How long does it takes to send a file of 640,640 bits from host A to host B over a circuit switched network.

Assume

All links are 1.63 Mbps.

Each link uses FDM with 27 channel frequencies.

500 ms to establish end-to-end connections.

Each circuit has transmission rate,

$$\frac{1.63 \times 10^6}{27} = 60370.37 \text{ bps.}$$

So it takes $\left(\frac{640,640}{60370.37} + 0.5 \right) = 11.11 \text{ sec}$

Department of Electronics and Telecommunication Engineering

Course No. ETE 315

Time: 20 minutes

CT 01

Marks: 20

ID:

1. A packet of size L bits is sent over a path consisting of N links. Each link has a transmission rate of R bps and a propagation delay of d_{prop} seconds. Assume the packet experiences negligible queuing and processing delays.
 - (a) Derive the total end-to-end delay experienced by the packet.
 - (b) If $L = 1200$ bits, $N = 5$, $R = 1$ Mbps, and $d_{prop} = 2$ ms per link, compute the total delay for the packet.

ChatGPT



Solve the Problem



Solution to Problem 1

(a) Derive the total end-to-end delay

For a packet of size L bits sent over N links:

- **Transmission Delay (per link):**

$$d_{\text{trans}} = \frac{L}{R}$$

- **Propagation Delay (per link):**

$$d_{\text{prop}} = d_{\text{prop}}$$

The total end-to-end delay for N links:

$$\text{Total Delay} = N \times d_{\text{prop}} + (N \times d_{\text{trans}})$$

Substitute d_{trans} :

$$\text{Total Delay} = N \times d_{\text{prop}} + N \times \frac{L}{R}$$

$$\text{\text{Total Delay}} = N \left(d_{\text{prop}} + \frac{L}{R} \right)$$

(b) Calculate the total delay for given values



Message



ChatGPT


$$\text{Total Delay} = N \left(d_{\text{prop}} + \frac{L}{R} \right)$$

(b) Calculate the total delay for given values

- $L = 1200 \text{ bits}$, $N = 5$, $R = 1 \text{ Mbps}$,
 $d_{\text{prop}} = 2 \text{ ms}$

1. Transmission Delay (per link):

$$d_{\text{trans}} = \frac{L}{R} = \frac{1200}{1 \times 10^6} = 0.0012 \text{ seconds} = 1.2 \text{ ms}$$

1. Total Delay:

$$\text{Total Delay} = N (d_{\text{prop}} + d_{\text{trans}})$$

$$\text{Total Delay} = 5 \times (2 + 1.2) = 16 \text{ ms}$$

Answer: The total end-to-end delay is **16 ms**.

Solution to Previous Question 2



Message



2. Consider a file of size F bits that is transferred from a sender to a receiver over a path with N links, where each link has a transmission rate R_i bps (where $i=1,2,\dots,N$). Assume that the file experiences no delay other than transmission delays.
- (a) Derive an expression for the effective throughput of the file transfer in terms of R_i for the N links.
- (b) If $R_1=2$ Mbps, $R_2=4$ Mbps, $R_3=1$ Mbps, calculate the effective throughput for transferring a file of size 10 MB.

≡ ChatGPT



(a) Effective Throughput Derivation

The effective throughput for file transfer over N links is determined by the **bottleneck link**, i.e., the link with the lowest transmission rate. Let R_{eff} represent the effective throughput:

$$R_{\text{eff}} = \min(R_1, R_2, \dots, R_N)$$

(b) Calculate Effective Throughput for Given Values

- $R_1 = 2, \text{Mbps}$, $R_2 = 4, \text{Mbps}$, $R_3 = 1, \text{Mbps}$

$$R_{\text{eff}} = \min(2, 4, 1) = 1 \text{ Mbps}$$

File size

$$F = 10, \text{MB} = 10 \times 8, \text{Mb} = 80, \text{Mb}.$$

Transfer time:

$$T = \frac{F}{R_{\text{eff}}} = \frac{80}{1} = 80 \text{ seconds}$$

Answer:

- Effective Throughput: **1 Mbps**

- Transfer Time: **80 seconds**.



Message



✓ Notation for the prev yr.

Application Layer:

The application layer is the top layer of the network model where users and network processes interact directly. It provides services like email, file transfer, and Web browsing. This layer defines protocols such as HTTP, FTP, DNS and SMTP to enable communication between software applications over a network. It ensures proper data formatting, encryption and session management for user-facing tasks, acting as a bridge between network services and end-user applications.

- Provide network services to end-user applications. (e.g: web browsers, email, clients)
- Enable communication between different software applications across networks.
- Data formatting, translate data into a format understandable by the receiver.
- Resource sharing: Manage access to shared network resources (e.g. files, printers).
- Authentication and authorization, Verify user identity and permissions.
- Session Management: Establish, maintain, and terminate communication sessions.
- Error handling: Detect and recover from errors at the application level.
- Directory services: Help applications find other services and resources on the network.
- Data encryption and security: Ensure confidentiality and integrity of transmitted data.

Some network apps:

- Social networking
- Web
- e-mail
- Text-messaging
- multi-user network games.
- Streaming stored video.
- P2P file sharing.
- Voice over IP (VoIP)
- Internet search
- Remote login.

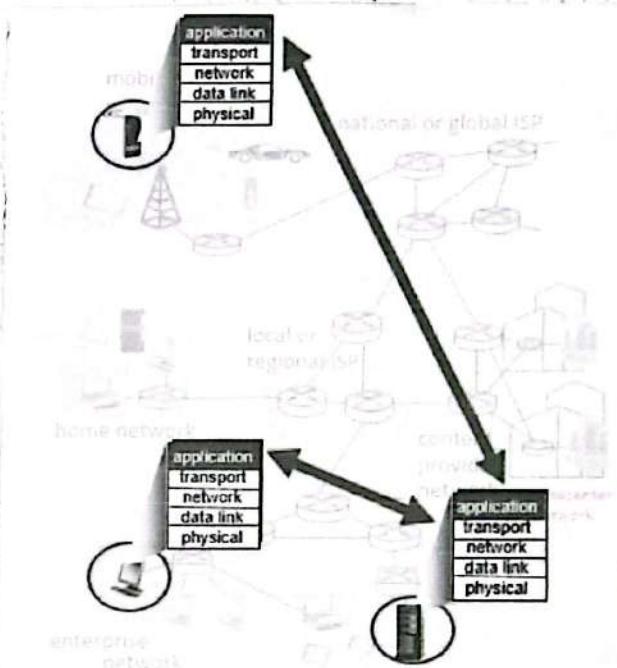
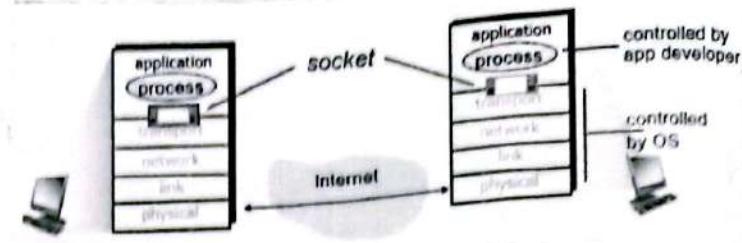


Figure: Application Layer in networking.

Processes:

A process in a program running on a device (host) that performs a specific task and can communicate over the network.



Processes interact by exchanging data and messages, typically using a unique port number for identification.

✓ Client process:

- Initiates communication

- Requests services or data from the server

- Usually runs temporarily during the interaction.

Server process:

- Waits passively for incoming requests.

- Provides services or resources to clients.

- Continuously runs, ready to accept multiple client connection.

Socket:

A socket is an endpoint for sending and receiving data between processes over a network. It combines an IP address and a port number to uniquely identify each process. 192.168.10.16:443

IP

HTTPS

port #

In process management, socket allows multiple applications to communicate simultaneously without confusion.

The OS manages sockets by assigning a unique port number to each active socket, tracking them in a socket table. It ensures correct delivery of data to the right process based on the socket's IP address and port pair (192.168.10.16:443), maintaining organized and efficient communication.

A process ID (PID) uniquely identifies a running process inside the OS. Port numbers identify network communication endpoints (sockets), not process directly. PID manages the overall process while port numbers manage network connections.

Web and HTTP:

The web is a system of interconnected documents and resources accessed via internet, using browsers.

HTTP - Hyper Text Transfer Protocol is the communication protocol used by browsers and servers to exchange web data like HTML pages, images and videos. HTTP defines how messages are formatted and transmitted, and how web servers and browsers should respond. HTTP is the web application layer protocol, using a client server model. Browsers (clients) request and display objects from web servers. A web page consists of a base HTML file and multiple reference objects, each addressable by a URL.



- HTTP runs over TCP, with clients initiating connections (typically using port 80).

- It is stateless protocol, meaning servers do not retain information about past client interconnections, simplifying design but losing session summary.
- Unless extra mechanism (like cookies) are used, HTTP messages are exchanged during a TCP session, which is closed after the transfer.

HTTP connections are mainly two types:



Fig 1: HTTP overview.

Non-Persistent HTTP

- One TCP connection per object.
- TCP connection closed after each object transfer.
- Requires multiple connections for one web page (base HTML + each object).
- High overhead due to frequent TCP setup/teardown (TCP handshake).

Persistent HTTP

- Single TCP connection for multiple objects.
- Connection stays open to send/receive multiple HTTP requests/responses.
- Reduces latency and TCP overhead.
- Two types:
 - Without pipelining: Requests sent sequentially.
 - With pipelining: Multiple requests sent without waiting for responses.

Non-persistent HTTP



WWW.cuet.ac.bd
/etc/home.index



Client

→ Server

- HTTP client creates TCP connection to WWW.cuet.ac.bd on port 80.
- Client sends HTTP request for /etc/home.index.
- Server accepts the connection, notifying client.
- Server sends HTTP response with home.index file.
- Client displays HTML.
- Server closes TCP connection.

If requests from client is sent for 10 JPEG images, then server responds for each JPEG repeating the above steps.

Non-persistent HTTP Response time.

RTT - Round Trip Time is the Time for a small packet to travel from client to server and back.

In non-persistent HTTP, each object requires

- One RTT to establish TCP connection (TCP handshake).
- One RTT for HTTP request and first byte of HTTP response.
- Plus the object transmission time.

So, Total time to get one object = 2 RTT's + object Transmission Time

Non-persistent HTTP issues:

- Requires 2RTT's per object
- OS overhead for each TCP connection.
- Browsers often open multiple parallel TCP connections to fetch referenced objects in parallel.

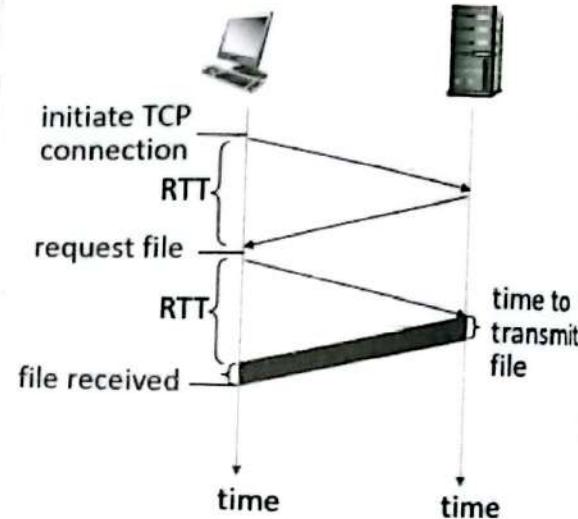
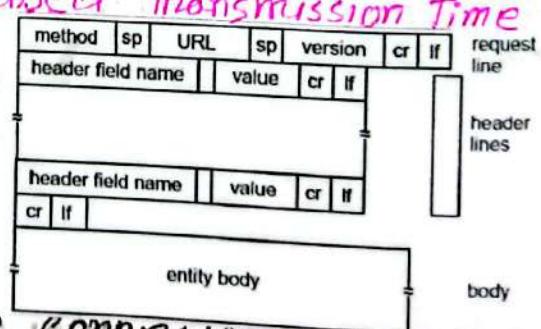


Fig: 2 RTT + object transmission.



connections to fetch

While in persistent HTTP, server leaves connection open after sending responses and uses one RTT for all the referenced objects.

HTTP

- GET: Requests data from a server (e.g. a web page). Data is sent in the URL.
- POST: Sends data to the server (e.g. form submission). Data is in the body, not the URL.
- HEAD: same as GET, but only retrieves headers (no body). Used to check resources.
- PUT: Uploads data to the server to create or update a resource at a specific URL.

HTTP Response Status Codes:

Status code appears in 1st line in server to client response message

- 200 OK! Request was successful; the requested content is included in the response.
- 301 Moved Permanently: The resource has been moved to a new URL provided in the response.
- 400 Bad Request: The Server could not understand the request due to invalid syntax.
- 404 Not Found: The requested resource is not available on the server.
- 505 HTTP version not supported! The server does not support the HTTP version.

Maintaining user/server state: Cookies

HTTP is stateless, so cookies are used to maintain state across multiple requests.

Step 1: Server sends a set cookie header in the HTTP response with a unique ID.

Step 2: Browser stores this cookie in a local cookie file.

Step 3: for the future requests to the same site, browser includes the cookie ID in the cookie header.

Step 4: Server checks cookie ID against the back-end database to identify the user.

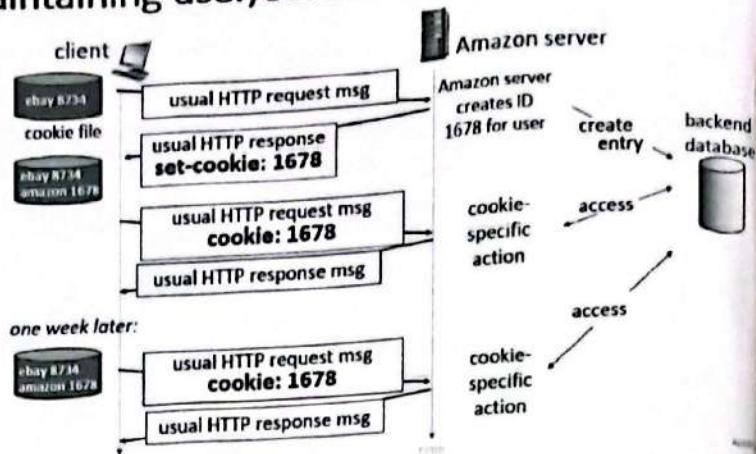
- User authentication, shopping cart tracking, personalized content and session management are maintained via cookies.
- Cookies help simulate a 'memory' between client-server exchanges by carrying state in messages.

Real-life example of cookies:

- Susan visits Amazon.com for the first time.
- Amazon sends a set-cookie: ID=789XYZ in its HTTP response. Susan's browser saves this cookie.
- On her next visit, her browser sends cookie: ID=789XYZ in the HTTP request.
- Amazon uses the ID to retrieve Susan's shopping cart and preferences from its database.

Result: Susan sees her saved cart and personalized recommendations

Maintaining user/server state: cookies



Web cache and proxy server:

A web cache (proxy server) is an intermediary system stores copies of web content to reduce access time and bandwidth usage.

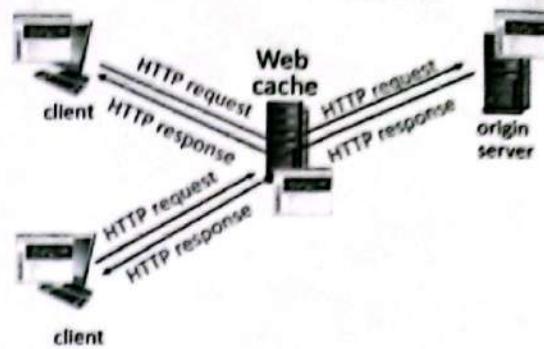


Fig: Web cache / Proxy server working.

- User configures browsers to send requests to the cache server.
- If requested object is in the cache → returned to client directly from web cache.
- If not in cache → cache fetch from origin server, stores it and serves it to client.
- Act as both client (to origin) and server (to user).

Benefits:

- Reduces client response time
- Lower bandwidth usage.
- Improves performance for slower servers.
- Reduces access link utilization and improves delay from minutes to milliseconds.

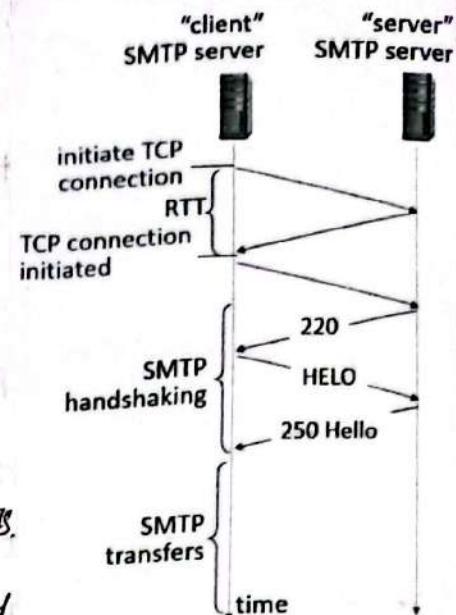
E-mail:

User agent (Mail client): Used to compose, send and read emails (e.g., outlook, gmail app).

Mail servers: Store emails in mailboxes (incoming) and message queues (outgoing).

SMTP - Simple Mail Transfer Protocol

- Used for sending emails between mail servers.
- Runs over TCP Port 25 for reliable delivery.
- Works in three phases: Handshaking, message transfer and closure.
- Uses command/response interaction (text commands & 5 fixed code)



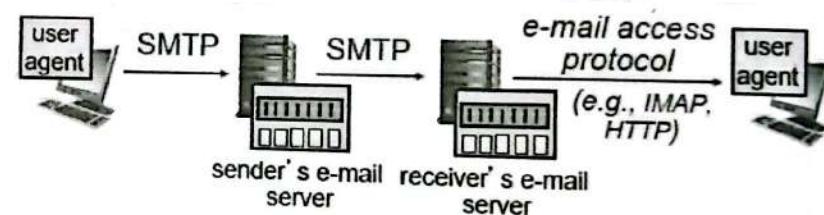
POP3 or IMAP protocols are used by user agents to retrieve emails from mail servers.

Ensures email delivery from sender \rightarrow server \rightarrow recipient's server \rightarrow recipient.

Email sending-receiving scenario:

Alice Sends email to Bob:

- Alice uses a User Agent (UA) to write an email to Bob.
- Her UA sends it via SMTP to her mail server.
- The mail server opens a TCP connection with Bob's mail server.
- The message is transferred and placed in Bob's mail box.
- Bob opens his UA to read the message.



E-mail Retrieval:

- SMTP handles delivery from sender to recipient's server.
- IMAP or POP3 is used by the recipient to retrieve the email.
- Webmail (Gmail etc) uses **HTTP + SMTP + IMAP/POP3**.

✓ IMAP & POP3:

IMAP: Internet Message Access Protocol allows users to read email directly from the mail server without downloading them. Messages stay on the server unless explicitly deleted.

POP3: Post Office Protocol Version 3: Downloads emails from the mail server to the local device and usually deletes them from the server afterward.

IMAP

- Emails stay on the server
- Access mail from multiple devices.
- Synchronizes folders in real time
- Requires constant internet connection.
- Better for managing large mailboxes

POP3

- Emails are downloaded and deleted from the server.
- Typically used on a single device.
- No folder synchronization.
- Can read offline once downloaded.
- Simple, less server storage required.

Q Why SMTP is used in email rather than HTTP?

A SMTP is specially designed for sending email between mail servers. It works with push based delivery, ideal for mail routing and queuing.

SMTP uses TCP port 25 for reliable message transfer.

HTTP is not designed for mail delivery but for retrieving web content (pull-based).

SMTP supports mail commands, queuing, and retries, essential for email flow.

Email clients use SMTP to send emails and IMAP/POP3 to receive them.

Thus SMTP ensures reliable and structured email transfer.

Domain Name System:

DNS translates human readable domain names (like google.com) into IP addresses.

DNS: a distributed, hierarchical database

Services:

- Host to IP address mapping.
- Host aliasing
Canonical and alternative names.
- Mail server aliasing.
- Load distribution (via multiple IPs for one name.)

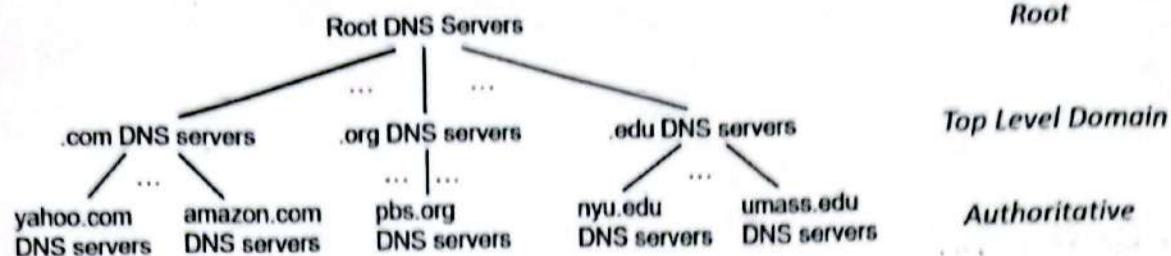


Figure: DNS Hierarchy.

DNS Hierarchy:

- Root DNS server
- Top level domain (TLD) servers (e.g. .com, .edu)
- Authoritative DNS servers (hold actual domain info).

✓ This notation indicates Prev Yr.

ICANN - Internet Corporation for Assigned Names and Numbers.
Manages IP address allocation, domain name system, root servers and coordinates unique identifiers globally for a stable, secure internet.

Why centralized DNS is used?

- Single point of failure: If the central server crashes, all domain name resolution fails.
- Traffic volume: One server can't handle billions of queries globally.
- Geographical distance: Central server far from users causes high latency.
- Maintenance challenges: Difficult to update and manage a single global DNS Database.
- Scalability issues: Centralized system can't scale with the growing Internet.
- Lack of redundancy: No backups or distribution increases risk of total outage.

Iterated query:

In an iterated query, the DNS resolver asks a DNS server for a domain name.

If the server doesn't know the answer, it does not contact other servers $\xrightarrow{2 \leftarrow 3}$. Instead, it

replies with a referral: the address of another DNS server. The resolver then repeats the query to the referred server.

This process continues step by step, until the resolver reaches the server with the final IP address.

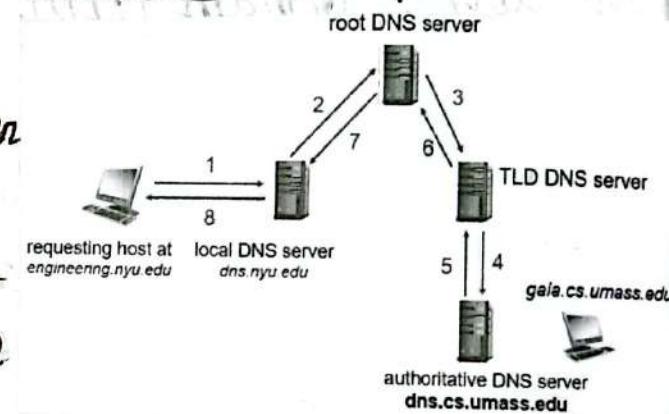
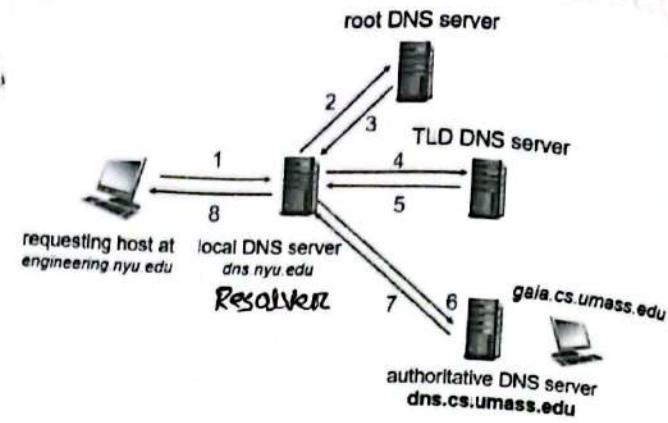
Resolvers handle all query iterations; servers just give hints.

Recursive query:

In a recursive query, the DNS resolver asks a DNS server to resolve a domain name completely. The server takes full responsibility to return the final IP address. If needed

it contacts other DNS servers on behalf of the resolver.

The client waits for the final response.



Q How DNS updates records?

- DNS records are updated through zone file changes on the authoritative DNS server. Admins edit the zone file and update the Start of Authority (SOA) record with a higher serial number. DNS servers then use zone transfer (AXFR/DNSFR) or dynamic updates (as per RFC 2136) to synchronize data. Changes propagate gradually due to TTL values cached across DNS Resolvers.

⌚ This symbol Indicates Prev Yr.

Difference between TLD and Authoritative DNS:

TLD DNS Server	Authoritative DNS Server
Stores address of authoritative servers for domains like .com , .org	Stores actual IP address of specific domain names (e.g., example.com)
Responds with referral to authoritative DNS	Responds with the final IP address of the domain
Handles domain extensions	Handles domain-specific records



Q: What are DDoS and Spoofing attacks? How do they occur, how severe are they, and how can they be defended against?

DDoS (Distributed Denial-of-Service):

A DDoS attack floods a server/network with massive fake traffic using compromised systems (botnets), overloading resources and denying access to legitimate users. It's highly disruptive and can cripple services for hours or days.

Spoofing Attack:

Spoofing involves faking identity—like IP or email address—to trick systems or users. It enables phishing, man-in-the-middle, and unauthorized access attacks.

Severity:

Both are severe; they can cause service outages, financial loss, and data breaches.

Defense:

- Use firewalls, anti-DDoS services (e.g., Cloudflare)
- Implement rate-limiting, filtering, and anomaly detection
- Deploy authentication and validation protocols (e.g., SPF, DKIM for emails)
- Regularly monitor and update security systems

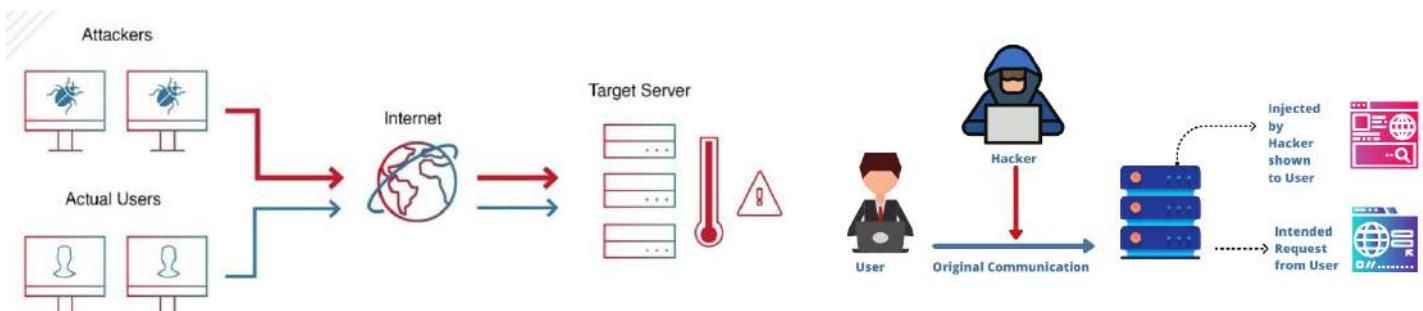


Figure 1: D-Dos attack

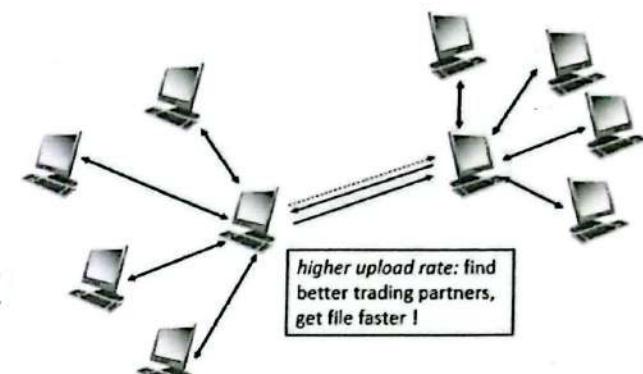
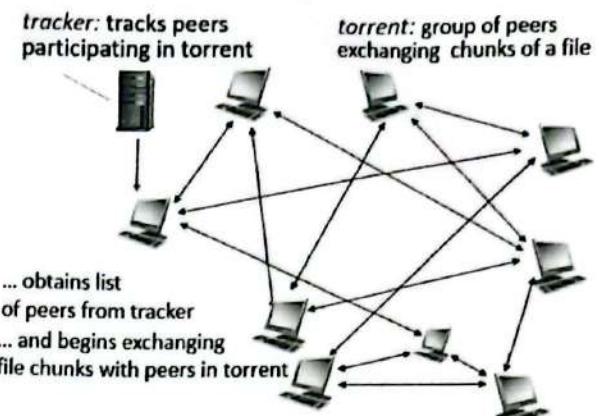
Figure 2: Spoofing attack

P2P, Client Server and Hybrid Architecture?

Client Server: A central server provides resources or services. Client request data from the server. Example: Web sites.

P2P architecture: All nodes(peers) act as both client and servers. Sharing resources directly without a central server. Example: Bit Torrent.

Hybrid Architecture: Combines client server and P2P. A central server helps co-ordinate or locate peers, but data is exchanged peer-to-peer. Example Skype or modern file sharing system.



Advantages

Client server:

- Centralized control and management.
- Easy to backup and update.
- Secure and reliable.

Dis-advantages

- Single point of failure.
- Higher server cost.
- Scalability under heavy load arises.

P2P

- No central server needed.
- More scalable with increased users.
- Reduces load on individual systems.

- Hard to manage and monitor.
- Security risks due to open sharing.
- Inconsistency performance.

Hybrid

- Combines strength of both models.
- Efficient resource sharing and co-ordination.
- More resilient to failure.

- More complex to design and maintain.
- Still partially dependent on a central server.

Zoom uses a Hybrid Architecture combining Client-Server and Peer-to-Peer (P2P) models.

 **Explanation (to the point):**

- For 1-on-1 calls, Zoom often uses P2P communication to reduce latency and server load.
- For group meetings, it uses Client-Server, routing audio/video through Zoom's cloud servers for control, stability, and scalability.
- This hybrid approach ensures low latency, reliability, and centralized control when needed.

Q: For P2P file sharing, there is no notion of client and server sides of communication session. Do you agree? Why or why not?

 Yes, I agree.

- In P2P file sharing, all peers act as both clients and servers.
- Each node can request (download) and provide (upload) files.
- No centralized server exists to control or manage file sharing.
- This decentralized approach enhances scalability and fault tolerance.

Transport Layer:

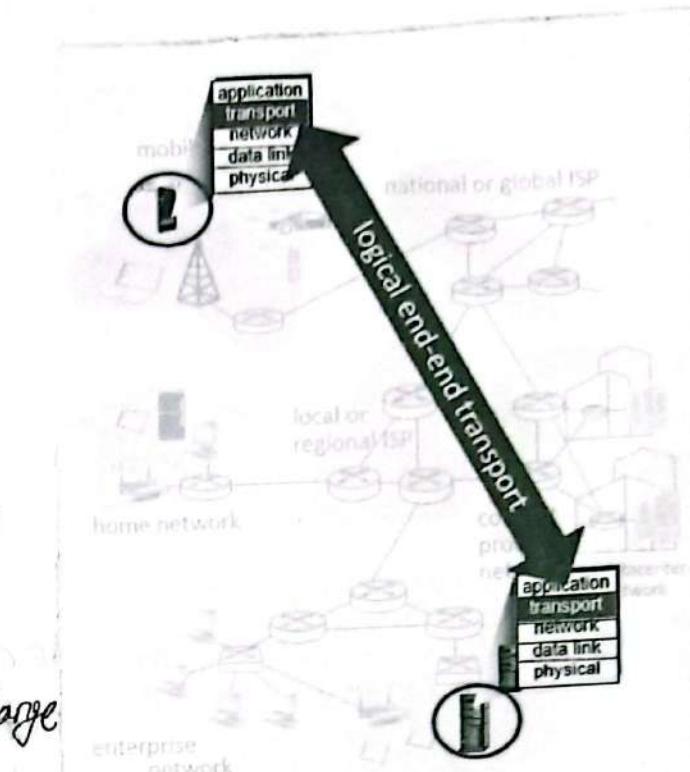
The transport layer is the fourth layer in the OSI and TCP/IP models that ensures reliable data transfer between devices over a network. It manages end-to-end communication, flow control, error checking and data segmentation.

Services and functionality:

- Process-to-process delivery: Ensures data reaches the correct application (using port numbers).
- Segmentation and reassembly: Breaks large data into smaller segments and reassembles them at the destination.
- Error detection and recovery: Ensures reliable delivery using checksums and acknowledgments.
- Flow control: Prevents overwhelming the receiver using mechanisms like sliding windows.
- Connection control: Supports both connection-oriented and (TCP) connectionless (UDP) services.
- Multiplexing/Demultiplexing: Manages multiple communication sessions simultaneously.

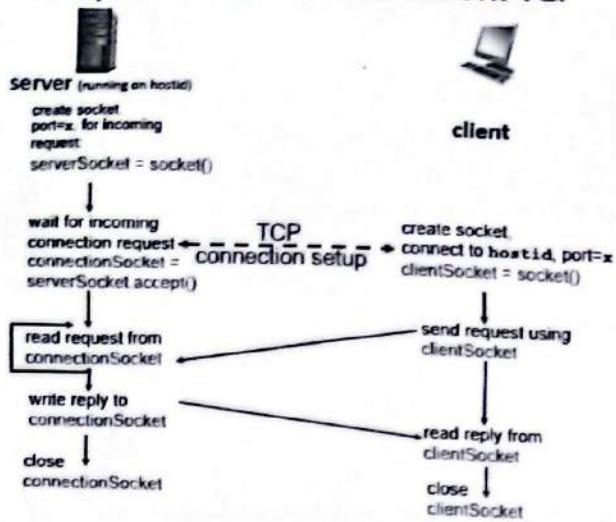
Q What transport services does an application need?

- Application typically need transport services such as reliable data transfer, error detection, flow control, multiplexing and timely delivery. Depending on the use case, some may also require connection-oriented communication (e.g. for file transfer) or low latency connectionless services (e.g. for live video or VoIP). Application choose between TCP (reliable) and UDP based on these needs.

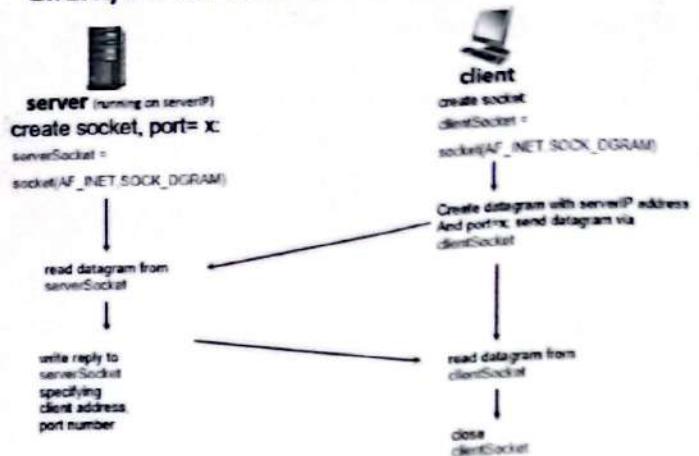


TCP & UDP:

Client/server socket interaction: TCP



Client/server socket interaction: UDP



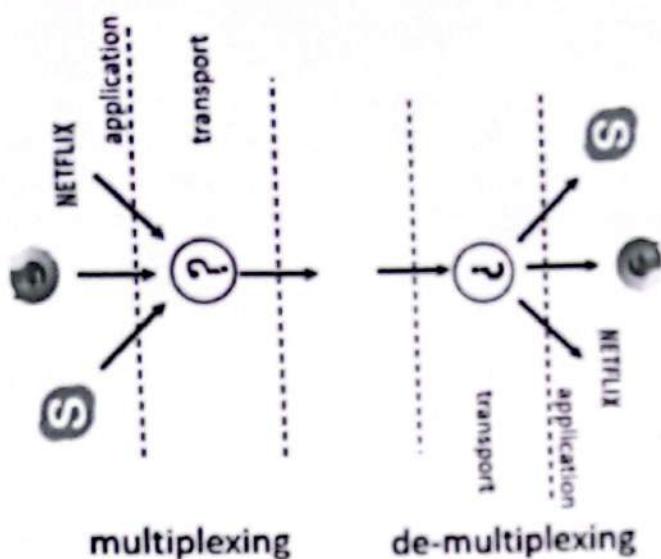
Transmission Control Protocol

- Connection oriented protocol that establishes a connection before data transfer
- Provides reliable data transfer with error checking and retransmission.
- Ensures ordered delivery of data packets by sequence number
- Slower due to overhead of reliability, flow and congestion control.
- Used for applications like web browsing, email, file transfer
- Uses more resources due to connection management.
- Examples: HTTP, FTP, SMTP

User Datagram protocol

- Connectionless protocol that sends data without setting up a connection.
- Provides unreliable data transfer, no guarantee of delivery.
- No guarantee of order in delivery.
- Faster with minimal overhead.
- Used for real-time applications like streaming, VoIP, gaming.
- Lightweight and uses fewer resources.
- Example: DNS, DHCP, live video

Multiplexing and Demultiplexing in TCP



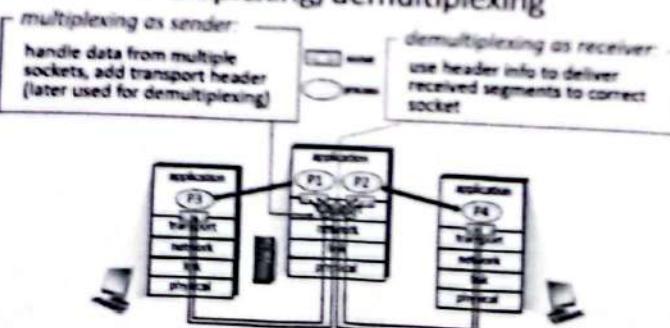
Multiplexing

- Happens at the sender side
- Combines data from multiple applications.
- Adds headers with source/destination IP: Port numbers.
- Sends all data through a single TCP connection.

De-multiplexing

- Happens at the receiver side
- Uses destination port numbers to deliver data to correct application.
- TCP uses socket pair (source IP: Port #) for identification.
- Ensures data is sent to the correct process among multiple active processes.

Multiplexing/demultiplexing



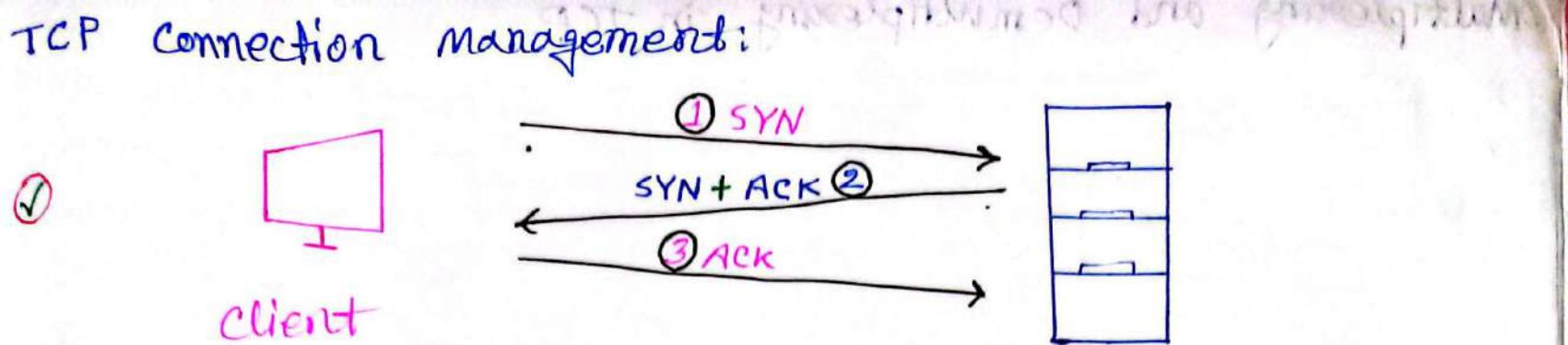


Fig: 3-Way handshaking in TCP connection.

- ① Client sent a SYN to initiate connection
- ② Server replies with a SYN+ACK
- ③ Client sends an ACK to complete the handshake

NOW both sides are ready to exchange data.

TCP Connection Termination:

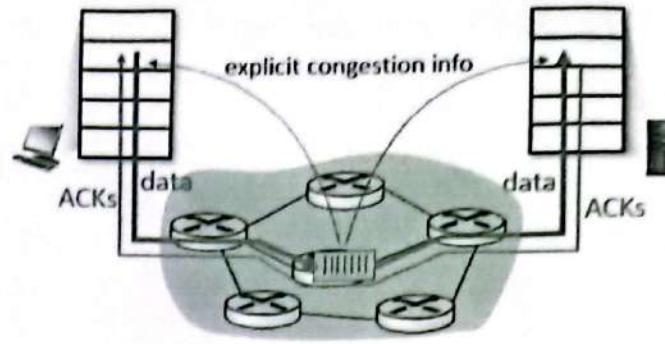
- Either side can initiate closing by sending a **FIN** (finish connection)
- The receiver acknowledges with an **ACK** and can send its own **FIN**.
- final **ACK** confirms the termination.

Q: 'TCP is connection oriented transport layer protocol' - justify.

A: TCP is called a connection-oriented transport layer protocol because it establishes a reliable connection between sender and receiver before data transmission. This is done using a three way handshake process (**SYN - SYN+ACK - ACK**) to set up communication parameters and ensure both parties are ready. Once the connection is established, TCP provides sequencing, error detection, flow control and retransmission to ensure reliable data transfer. After communication ends, TCP gracefully terminates the connection.

TCP Congestion Control:

✓ TCP controls congestion using a mechanism called congestion control, which adjusts the data transmission rate based on network conditions it uses.



- Slow Start: Increases congestion window exponentially until loss occurs or threshold is reached.

Congestion Avoidance: Grows congestion window linearly after slow start to avoid overload.

Fast Retransmit and Fast Recovery: Detects packet loss via duplicate ACKS, reduces congestion window quickly, then gradually increases to recover.

MSS \rightarrow Maximum Segment Size

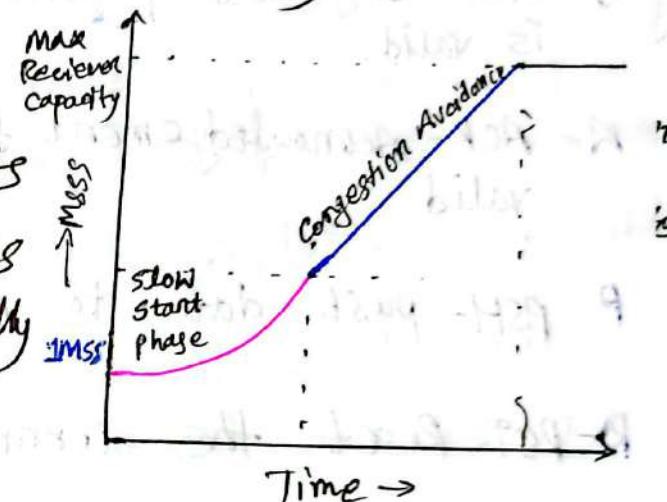


fig: TCP Congestion Control graph MSS vs Time

TCP Flow Control:

TCP uses flow control to prevent the sender from overwhelming the receiver. Achieved using a sliding window mechanism.

- Receiver advertises a window size indicating how much data it can accept.
- Sender limits the number of unacknowledged bytes to this window size.
- If the receiver buffer is full, it advertises window size = 0 (pause transmission)
- Sender resumes sending when window size > 0 is received.
- Ensures efficient data transfer and avoids buffer overflow.

✓ TCP Segment Structure:

A TCP segment includes:

- Source/Destination port.
- Sequence number.
- Acknowledgement number.
- Header length, flags, window size.
- Checksum, urgent pointer, options and data.

✓ Flags (Control Bits):

U - URG - Urgent pointer field is valid.

A - ACK - Acknowledgement field is valid.

P - PSH - push data to receiving application.

R - RST - Reset the connection.

S - SYN - Synchronize sequence numbers (start connection).

F - FIN - Finish terminate the connection.

④ How TCP sets sequence and acknowledgement numbers?

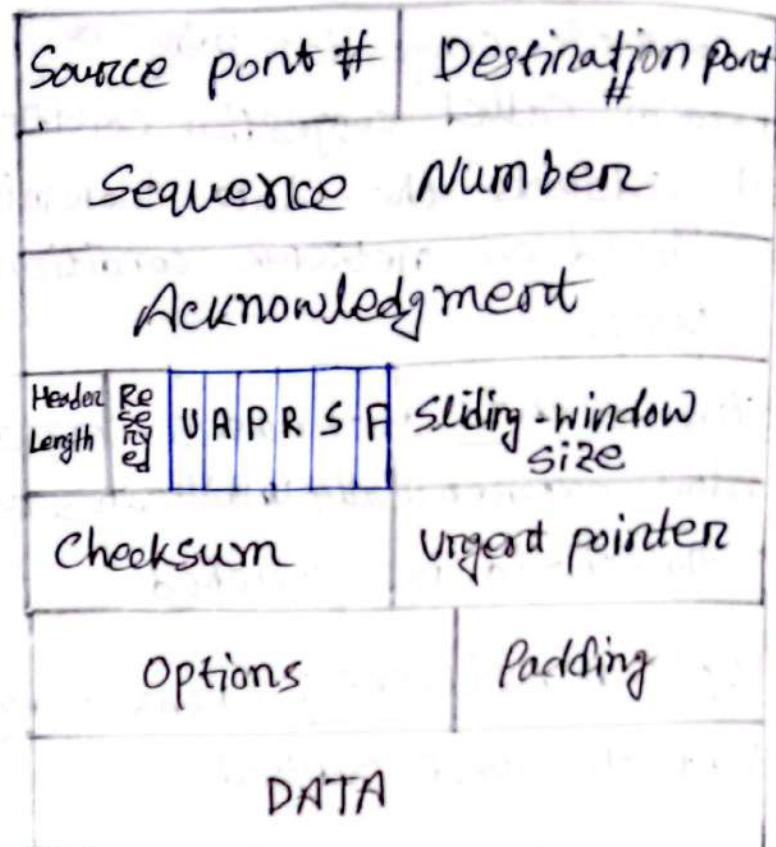
④ When A TCP connection starts, each side selects an initial sequence number (ISN) randomly.

SYN segment: Sender sets the ISN and sends it.

Receiver replies with (SYN-ACK) sets its own ISN, and acknowledges sender's ISN+1.

From then on each byte of data sent increments the sequence number. Acknowledgement number always indicates the next byte expected.

TCP ensures reliable data delivery using these numbers to track data send and receive.



TCP Retransmission Scenario

1. Lost ACK Scenario:

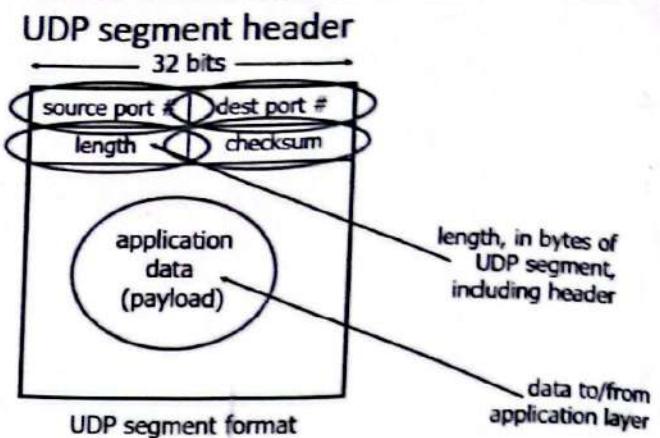
- Sender sends a segment and waits for ACK.
- Receiver gets the segment and sends ACK but ACK get lost.
- Sender times out, assumes segment lost, retransmits the same segment.
- Receiver gets duplicate segment and resends the ACK, but does not store duplicate data, due to sequence number.

2. Premature Timeout:

- Network delay causes ACK to arrive late.
- Sender times out before ACK arrives, retransmits the segment.
- Receiver gets duplicate segment and responds with duplicate ACK.
- Causes unnecessary retransmission, wastes bandwidth but maintains reliability.

UDP Segment Header:

- UDP header size is 8 byte fixed.
- Fields (2 bytes).
 - Source port #: Identifies sending application
 - Destination port #: Identifies receiving application
 - Length: Total length of UDP + data.
 - Checksum: Used for error detection of header and data.



✓

■ Why voice and video traffic is often sent over TCP rather than UDP today's internet?

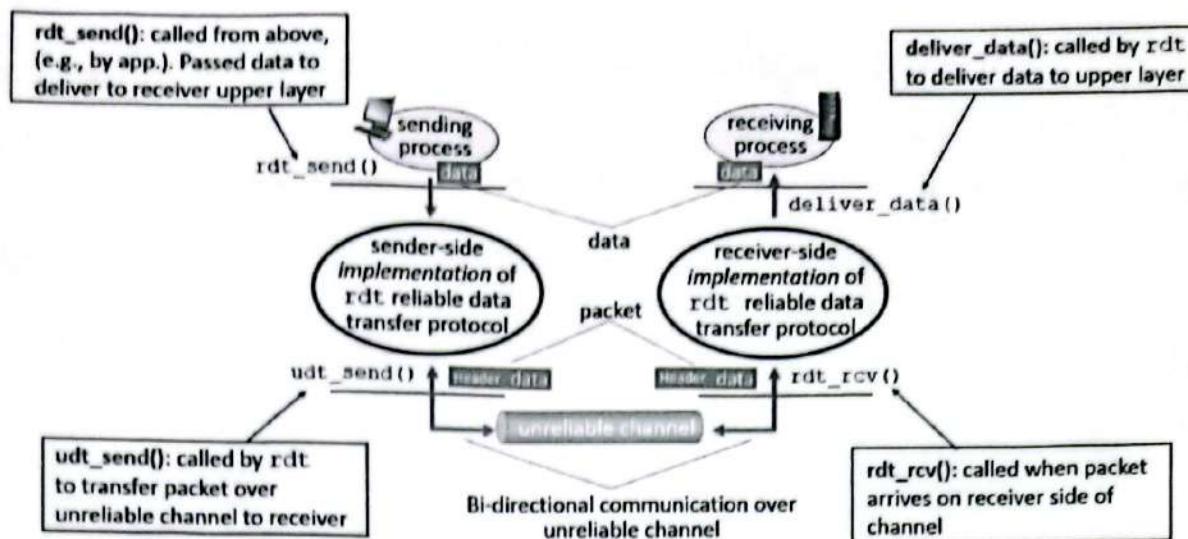
- Firewall/NAT traversal: TCP is more likely to pass through firewalls and NATs.
- Congestion control: TCP manages congestion better, avoiding network overload.
- Reliability: Many platforms prioritize reliability over latency.
- HTTPS usage: Video platforms use HTTPS.
- Simpler deployment: TCP is universally supported, reducing compatibility issues.

✓

■ Why HTTP, FTP, SMTP and POP3 run on TCP rather than UDP?

- Reliability: TCP ensures data delivery, essential for file transfer and email.
- Ordered delivery: These protocols data in correct sequence.
- Error checking & correction: TCP handles retransmission on loss.
- Connection-oriented: Needed for session management in HTTP etc.
- Flow control: Prevents overwhelming the receiver.

Reliable data transfer protocol (rdt): interfaces



Rdt ensures accurate and complete delivery of data over an unreliable channel. It handles errors such as packet loss, duplication and bit corruption using mechanisms like acknowledgments (ACKs), negative Acknowledgments (NAKs), checksum and re-transmissions.

Protocols like rdt 1.0, 2.0, 3.0 build increasing levels of reliability. RDT is fundamental in transport protocols like TCP to guarantee ordered, error-free communication between Sender and receiver.

Rdt 2.0

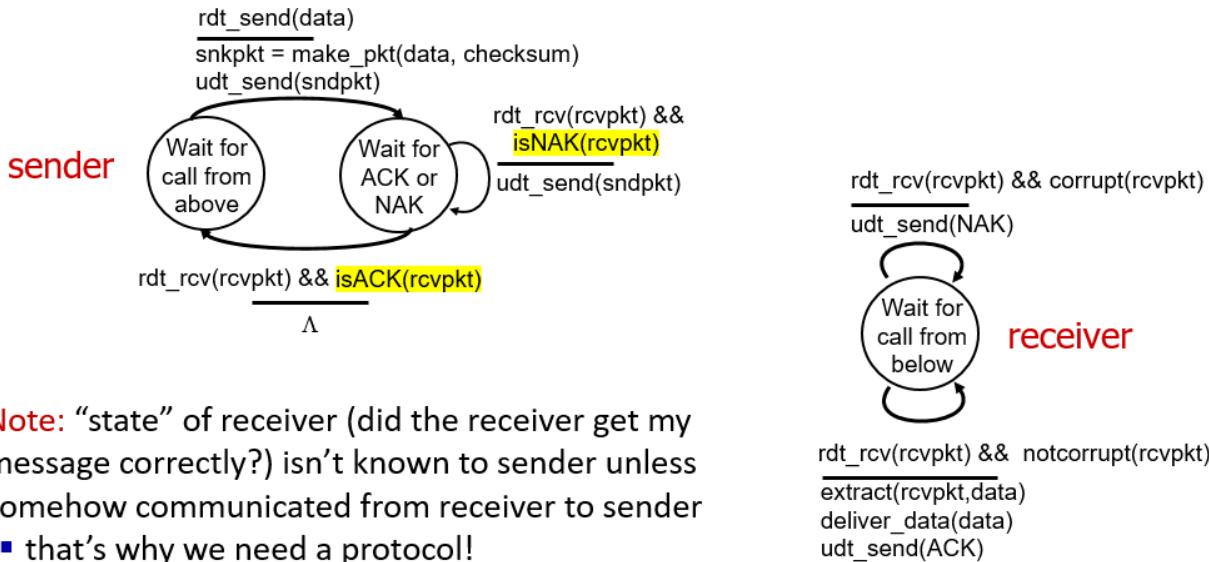
Ensures reliable data transfer over channels prone to bit errors.

Mechanism:

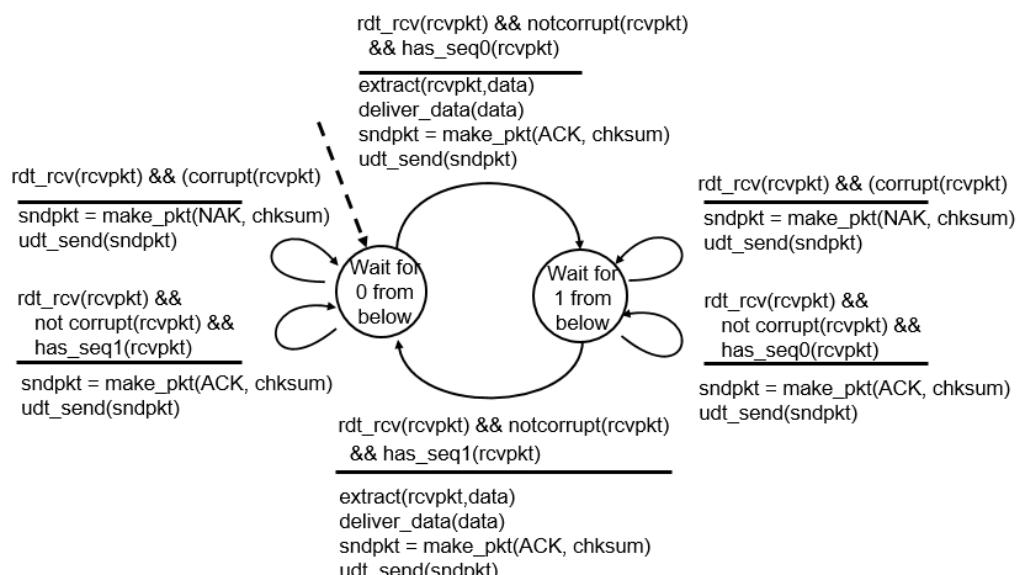
- Sender transmits a packet with a checksum.
- Receiver checks for errors, using the checksum.
- If the packet is error-free receiver sends an ACK.
- If errors are detected, receiver sends a NAK.
- Upon receiving a NAK, sender re-transmits the packet.

✓ This symbol Indicates Prev Yr.

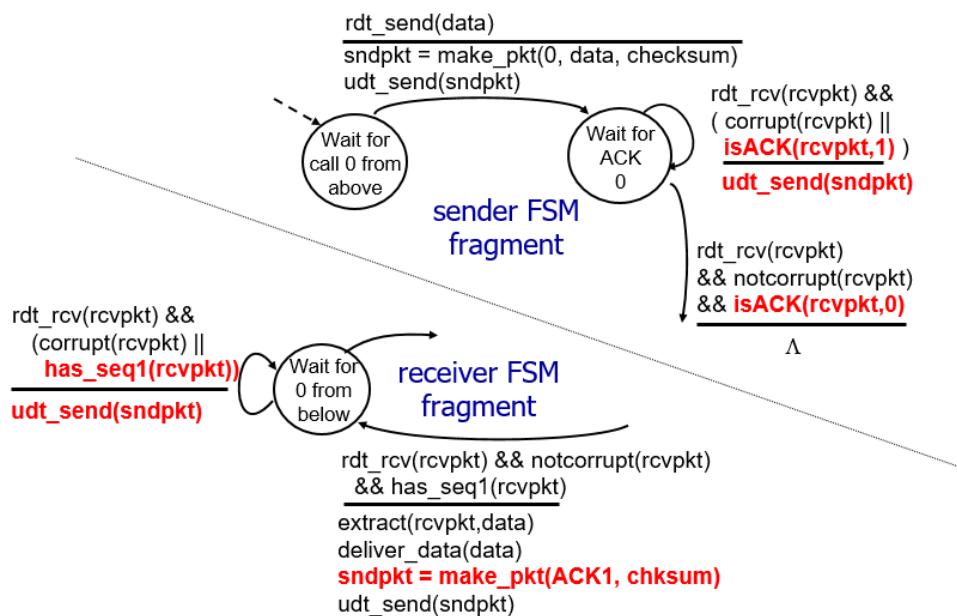
✓ rdt2.0: FSM specification



rdt2.1: receiver, handling garbled ACK/NAKs



rdt2.2: sender, receiver fragments



Error Scenario of rdt 2.0

If an ACK or NAK is corrupted during transmission

- Sender cannot determine whether the packet was received correctly.
 - To be safe sender re-transmit the packet.
 - This may lead to duplicate packets at the receiver.
 - Receiver must handle duplicates appropriately to maintain data integrity.

The limitation led to development of RDT 2.1, which incorporates sequence numbers to distinguish between new and duplicate packets.

Rdt 3.0:

Handles packet loss and corruption over unreliable channels.

Introduces timeouts: sender waits for ACK within a certain time.

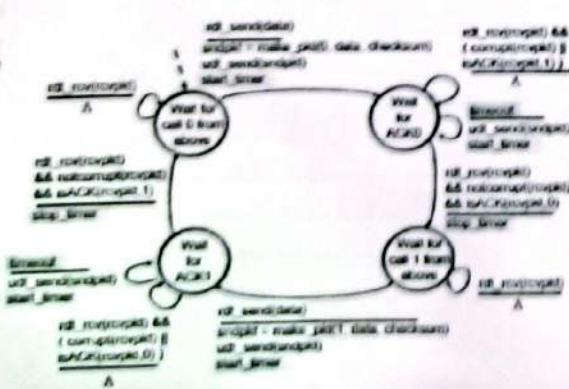
If ACK not received before timeout, sender retransmits the packet.

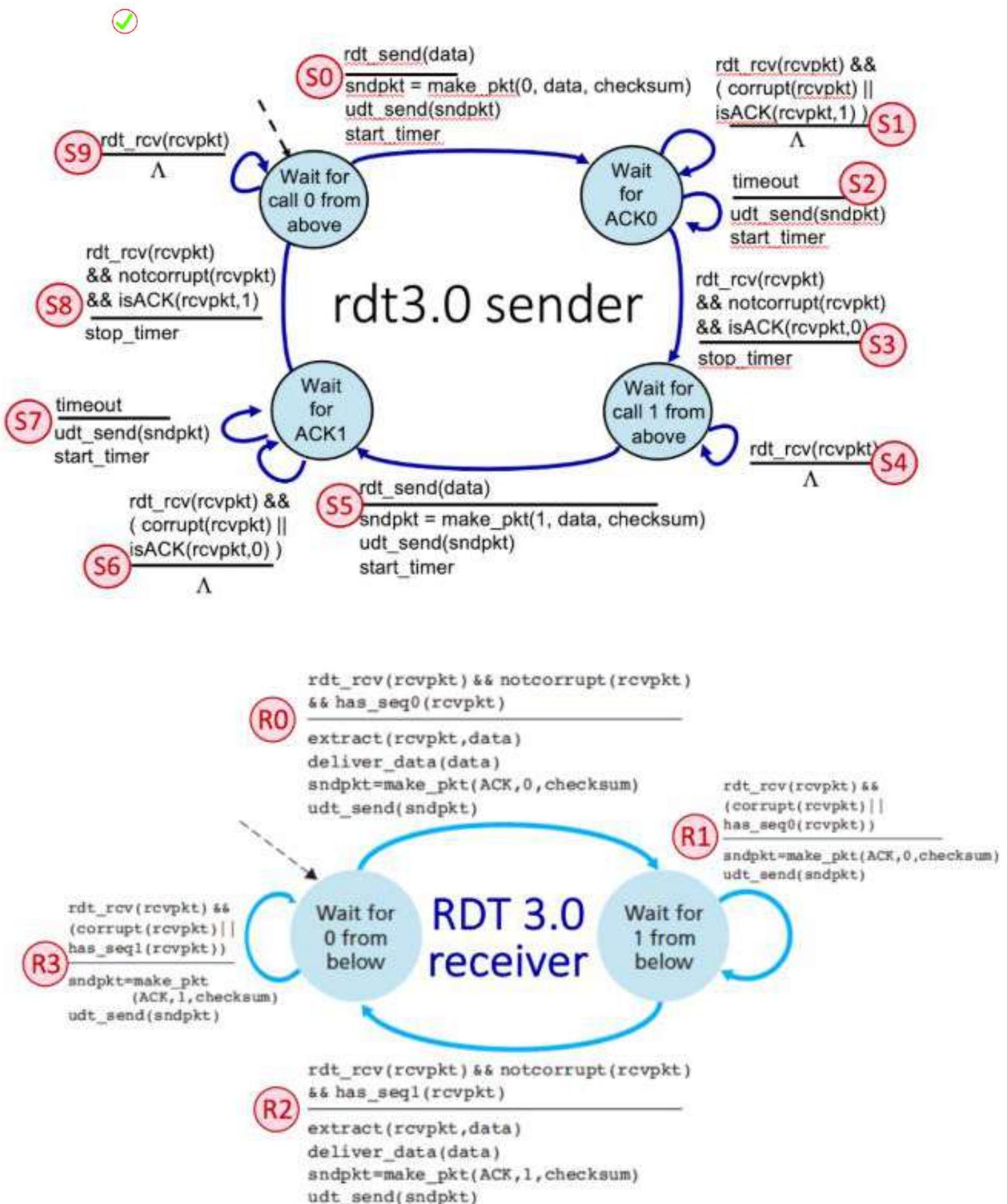
Use sequence numbers to identify the duplicates.

Receiver sends ACK for correct packet; ignores duplicates.

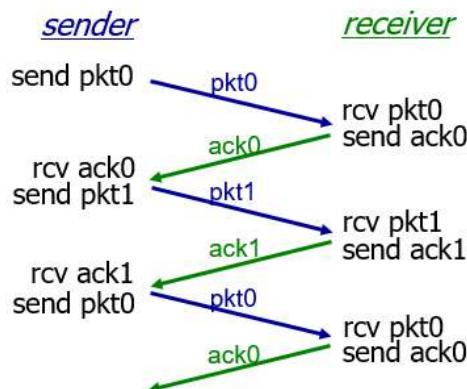
Handles scenarios with lost packets or lost ACKs. Provides stop-and-wait reliability; sender sends one packet at a time and waits.

rdt3.0 sender

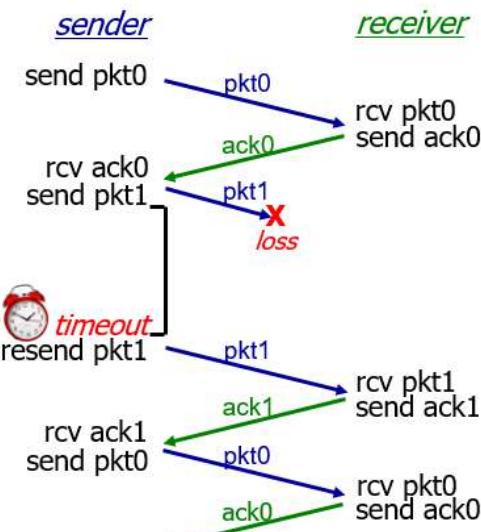




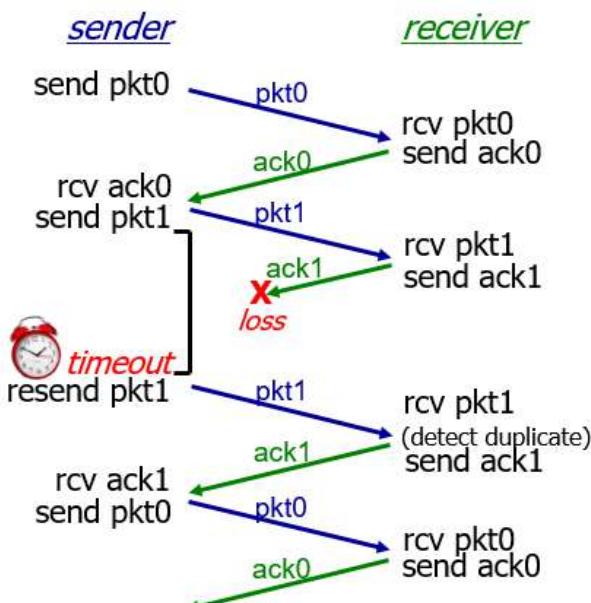
rdt3.0 in action



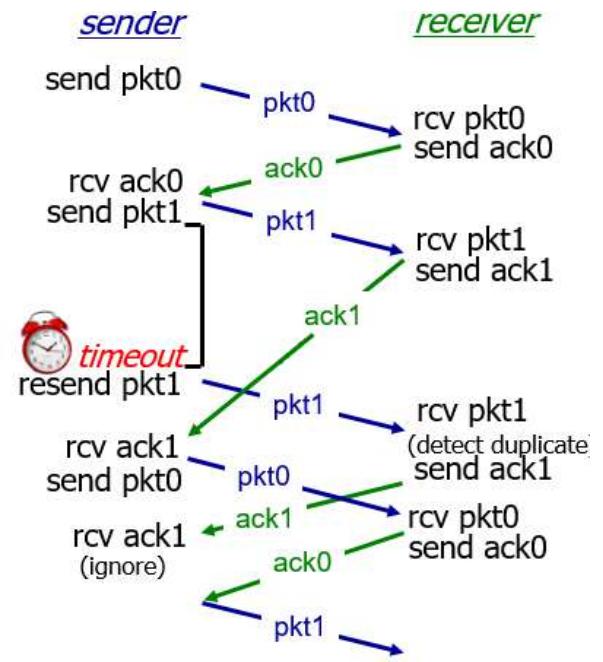
(a) no loss



(b) packet loss



(c) ACK loss



(d) premature timeout/ delayed ACK

Performance of rdt3.0 (stop-and-wait)

- U_{sender} : **utilization** – fraction of time sender busy sending
- example: 1 Gbps link, 15 ms prop. delay, 8000 bit packet
 - time to transmit packet into channel:
$$D_{\text{trans}} = \frac{L}{R} = \frac{8000 \text{ bits}}{10^9 \text{ bits/sec}} = 8 \text{ microsecs}$$

RDT Summary Table

RDT Version	Problem Addressed	New Mechanism Introduced
RDT 1.0	No errors	Basic send/receive logic
RDT 2.0	Bit errors in channel	Checksums, ACK, NAK for error detection
RDT 2.1	Corrupted ACK/NAK	Sequence numbers to detect duplicates
RDT 2.2	NAKs unnecessary	Only ACKs used; duplicates handled via sequence #
RDT 3.0	Packet loss (no ACK received)	Timer and retransmission (timeout mechanism)

✓ Question: In RDT protocol, why do we need to add sequence numbers?

Answer (to the point):

Sequence numbers are needed to detect **duplicate packets** and **ensure correct ordering**. When acknowledgments (ACKs) or packets are lost or corrupted, the sender may retransmit the same packet. The receiver uses sequence numbers to distinguish **new data** from **retransmissions**, thus ensuring reliable and **in-order delivery** of data.

✓ Question: Why is Internet checksum used in internet communication?

Answer (to the point):

Internet checksum is used to **detect errors** in transmitted data (e.g., bit flips) during communication. It ensures **data integrity** by allowing the receiver to verify whether the received data matches what was sent. If the checksum doesn't match, the data is considered **corrupted** and is discarded or retransmitted.

✓ Question: Briefly explain the reliable data transfer scenario.

Answer (to the point):

Reliable data transfer ensures that data is delivered **accurately and in order** over an unreliable network. It handles issues like **packet loss**, **bit errors**, and **duplicate data**. Protocols like rdt (reliable data transfer) use techniques such as:

- **Acknowledgment (ACK)** and **Negative Acknowledgment (NAK)**
- **Sequence numbers** to detect duplicates
- **Timers** for retransmission of lost packets
- **Checksum** for error detection

Examples include rdt 2.0 (error detection + ACK/NAK), rdt 3.0 (adds timeout-based retransmission).

Pipelining:

Pipelining is the technique where multiple packets (or segments) are sent by the sender without waiting for individual acknowledgements from the receiver.

Necessary:

- Increases network utilization and throughput.
- Reduces idle time for sender and receiver.
- Essential for high bandwidth and high delay network.
- Prevents underutilization caused by stop-and-wait protocols.
- Enables continuous data flow, boosting efficiency.

■ Pipelining increases utilization by a factor - prove.

■ In stop-and-wait utilization,

$$\frac{\text{Transmission time}}{\text{Transmission time} + \text{Round Trip Time}} = \text{Utilization}$$

$$\text{or, } U_{SW} = \frac{L/R}{L/R + RTT}$$

In pipelining with n packets (see fig.)

utilization become

$$U_{\text{pipelined}} = \min\left(n \times \frac{L/R}{L/R + RTT}\right)$$

Thus pipelining increases utilization by a factor of n .



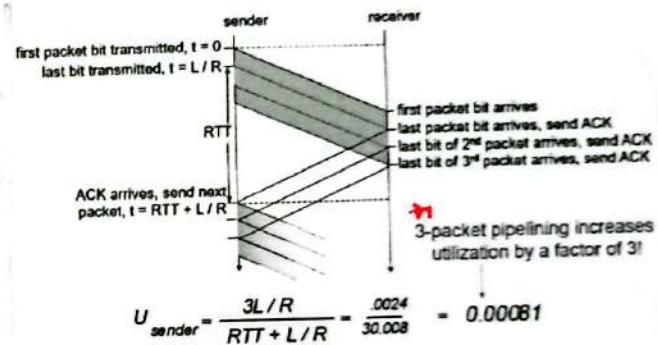
No-Back-N:

A reliable data transfer protocol that uses a sliding window and sends multiple packets without waiting for individual ACKs.

Working:

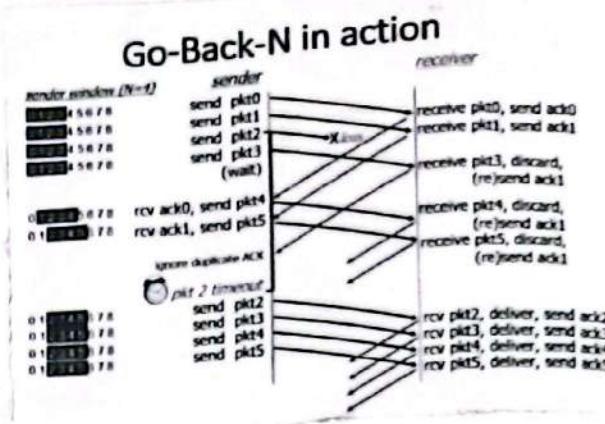
- Sender can transmit up to N acknowledge packets.
- Receiver only accepts in-order packets.
- If a packet is lost or corrupted, receiver discards that and all following packets.
- Sender goes back and retransmits the lost packet and all subsequent packets in the window.

Pipelining: increased utilization



Example:

1. Sender sends packet 0~3 (see fig).
2. Receiver gets 0,1,3 but Pkt 2 is lost.
3. Receiver discards pkt 3 and sends ACK for pkt 1
4. Sender retransmit pkt 2 and 3.



Advantages: Simpler than Selective Repeat.

Disadvantages: Wastes bandwidth when many packets after loss are discarded.

Selective Repeat:

A reliable data transfer protocol that allows multiple packets to be sent and acknowledged independently.

Working:

- Sender keeps a window of multiple unacknowledged packets.
- Receiver individually acknowledges each correctly received packets.
- Out-of-order packets are buffered by the receiver.
- Only lost or corrupted packets are re-transmitted, not the whole window.

Example:

1. Sender sends packet 0~3
2. Receiver receives packet 0,1,3 but pkt 2 is lost.
3. Receiver ACKs 0,1 and 3.
4. Sender retransmit pkt 2 only.
5. Receiver inserts pkt 2 in-order.

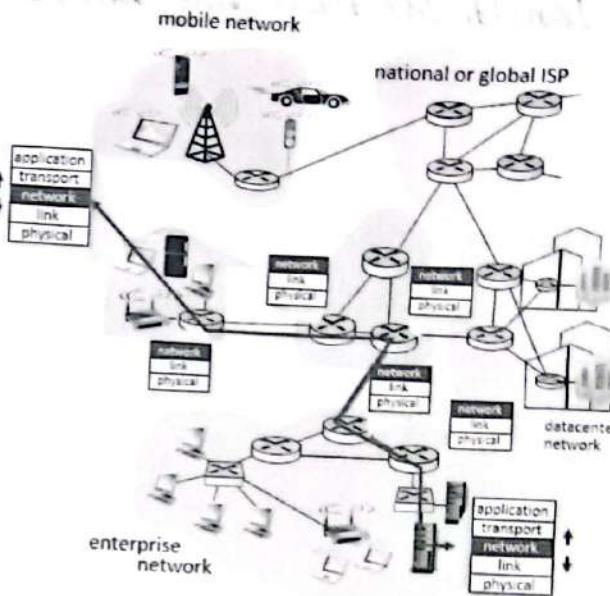
Advantages: Efficient use of Bandwidth, fewer retransmission.

Network Layer

The network layer is the third layer of the OSI model that manages the delivery of data packets across multiple networks from the source to the destination.

Services and functionalities:

- Logical Addressing: Assigns IP addresses to uniquely identify devices.
- Routing: Selects best path for data to travel across networks.
- Packet Forwarding: Moves packets from source to destination node-by-node.
- Fragmentation/Reassembly: Breaks large packets into smaller ones and reassembles them.
- Error Handling & Diagnostics: Handles delivery failures and provides diagnostics (e.g. ICMP).
- Connectionless Communication: Usually sends data without establishing a connection (e.g. IP).
- Traffic Control: Helps manage congestion in routing path.
- QoS Support: Prioritizes specific traffic based on service quality.



IP addressing is a method to assign a unique identifier to each device connected to a network, enabling devices to send and receive data. It ensures accurate delivery of data by identifying source and destination devices on the internet or local networks. Types include:

- IPv4: 32 bit address, e.g.: 192.168.1.1

IPv6: 128 bit address, e.g.: 2001:0db8:85a3::8a2e:0370:7334.

Required Definitions:

1. **IPv4 address:** a 32-bit number, usually written in dotted decimal form, that uniquely identifies an interface of some computer
2. **Host Address:** another term for IP address
3. **Network:** a group of hosts, all of which have an identical beginning position of their ip addresses.
4. **Network Number:** a 32-bit number that represent a network and it can't be assigned as ip address of a host
5. **Network address:** another term for the network number.
6. **Broadcast address:** a 32-bit number that is used to address all hosts in the network. It can't be assigned as an ip address of a host.
7. **Subnet:** a group of hosts, all of which have an identical portion of their ip addresses, a subnet differs from a network in that a subnet is a further subdivision of a network.
8. **Subnet number:** a 32-bit number that represent a subnet. It can't be assigned as ip address of host.
9. **Subnet address:** another term for the subnet number.
10. **Subnet broadcast address:** a 32-bit number that is used to address all hosts in the subnet. It can't be assigned into a host's IP address.
11. **Sub-netting:** the process of subdividing networks into smaller subnets.
12. **Subnet mask:** A 32-bit combination used to describe which portion of an address refers to the subnet and which part refers to the host.
13. **Network mask:** 32-bit number. The mask is used by computers to calculate the network number of a given IP address by performing a Boolean AND operation of the address and mask.
14. **Address mask:** another term for a mask
15. **Interface:** A network connection.

IPv4 Addresses:

An IP address is a 32-bit number, usually written in dotted decimal form, that uniquely identifies an interface of some computer.

The address is made up of 32-binary digits which can be divisible into a network portion and host portion with the help of a subnet mask. The 32 bits are broken into four octets ($1 \text{ octet} = 8 \text{ bits}$). Each octet is converted to decimal and separated by a period (.). The value in each octet ranges from 0 to 255 decimal or 00000000 to 11111111 binary.

10. 1. 23. 19
00001010 00000001 00010111 00010011

IP Address classes:

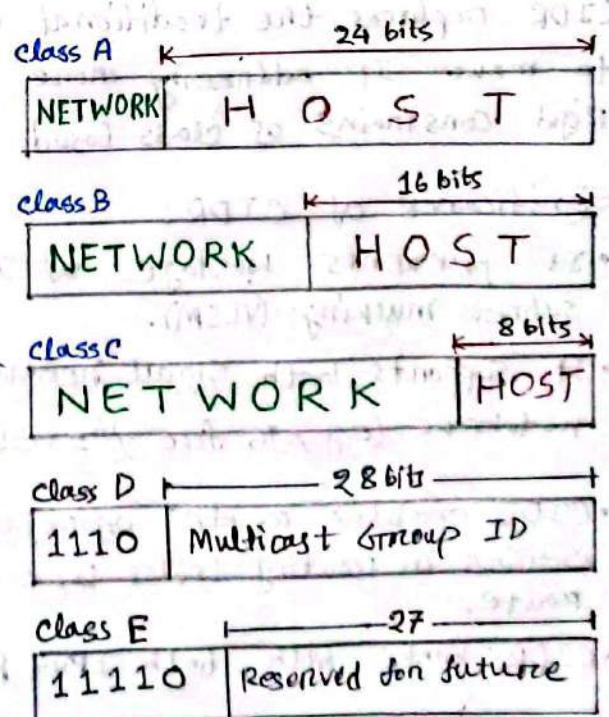
From an IP address, its class can be determined from the three high order octet bits. These address's octet are broken down to provide an addressing scheme that can accommodate large and small networks.

There are five different classes of networks, A to E. This text focuses on addressing classes A to C, since class D and E are reserved.

Class A: The first octet is the NETWORK portion and octet's 2, 3, 4 (next 24 bits) are for the network manager to divide into subnets and hosts. Class A addresses are used for networks that have more than 65536 hosts, actually $2^8 - 2 = 16777214$ max.

Class B: The first two octets are the network portion, octet's 3 & 4 (16 bits) are for local subnets and hosts. Class B networks that have between $256 \times (2^8 - 2) = 65534$ hosts.

Class C: Three first three are the network portion, and the last part of the 4 octets are the host portion that can handle 254 hosts.



Network Masks:

A network mask helps to know which portion of the address identifies the network and which portion of the address identifies the node.

Class A, B & C has default masks, also known as natural masks as shown below.

Class A: 255.0.0.0

Class B: 255.255.0.0

Class C: 255.255.255.0

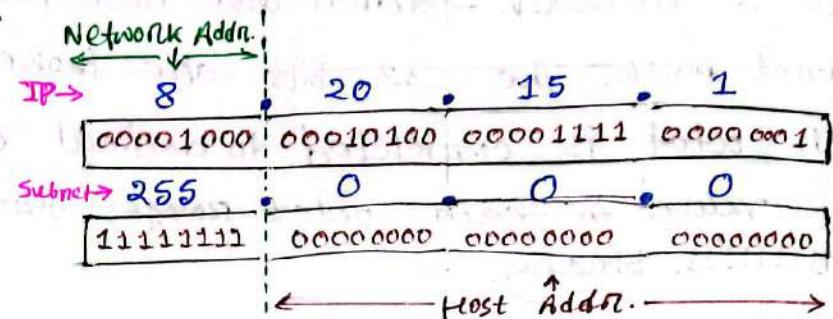


Fig: The mask helps to identify network and node/host part.

CIDR Notations:

Classless Inter Domain Routing is a method for representing IP address and their associated network prefixes. It is written as an IP address followed by a forward slash (/) and a number, which indicates the number of bits in the network prefix (subnet mask).

For Example: 192.168.1.0/24 indicates, the network address is 192.168.1.0 and /24 means the first 24 bits are used for the network portion, and the remaining 8 bits are for the host addresses.

Why CIDR is used?

CIDR replaces the traditional class based IP addressing system. Class A, B, C to make IP addressing more flexible and efficient. It eliminates the rigid constraints of class based fixed subnet masks.

Significance of CIDR:

- It prevents wastage of IP addresses by allowing variable length subnet masking (VLSM).
- It supports both small networks (e.g. /30 for $2^2 = 4$ addresses) and large networks (e.g. /16 for $2^{16} = 65536$ addresses).
- CIDR enables route aggregation (supernetting), reducing the number of entries in routing tables by summarizing multiple IP addresses into single route.
- CIDR works with both IPv4 & IPv6 address ensuring long term relevance.

Subnetting:

Subnetting is the process of dividing a large network into smaller, more manageable subnetworks, or subnets.

It helps improve network efficiency, enhances security, and optimizes the use of IP addresses by allocating only the required number of addresses to each subnet.

In Subnetting: $192.168.13.10/16$

First 16 bits for Network address and the rest (32-16) 16 bits for host allocation.

- Each network portion of the IP address is extended using bits from the host portion (subnet mask).
- Each subnet operates as an independent network.

Experimenting classes from CIDR Notation:

IP address with CIDR	Subnet Mask	Class Names	First octet
$10.168.1.14/8$	$255.0.0.0$	A	10
$190.168.1.15/16$	$255.255.0.0$	B	190
$192.168.1.16/24$	$255.255.255.0$	C	192

Exercise 1: Find the class of each address

- a. $00000001\ 00001011\ 00001011\ 11101111$
 - Convert the first octet to decimal: $00000001 = 1$.
 - Class A: 0-127.
- b. $11000001\ 10000011\ 00011011\ 11111111$
 - Convert the first octet to decimal: $11000001 = 193$.
 - Class C: 192-223.
- c. $14.23.120.8$

Class	First octet value	Subnet mask
A	0-127	8
B	128-191	16
C	192-223	24
D	224-239	-
E	240-255	-

 - The first octet is 14.
 - Class A: 0-127.
- d. $252.5.15.111$
 - The first octet is 252.
 - Class E: 240-255.

Exercise -2

b. $192.168.10.0/27$

- Subnet Mask: $255.255.255.224$ (5 bits for subnetting)
- Number of Subnets: $2^3 = 8$ subnets.
- Number of Hosts/Subnet: $2^5 - 2 = 30$ hosts.
- Subnets:
 - $192.168.10.0 - 192.168.10.31$
 - $192.168.10.32 - 192.168.10.63$
 - $192.168.10.64 - 192.168.10.95$
 - $192.168.10.96 - 192.168.10.127$
 - $192.168.10.128 - 192.168.10.159$
 - $192.168.10.160 - 192.168.10.191$
 - $192.168.10.192 - 192.168.10.223$
 - $192.168.10.224 - 192.168.10.255$

Exercise 2: Subnet the following addresses

- $192.168.10.0/26$
 - Subnet Mask: $255.255.255.192$ (6 bits for subnetting)
 - Number of Subnets: $2^2 = 4$ subnets.
 - Number of Hosts/Subnet: $2^6 - 2 = 62$ hosts
 - Subnets:
 - $192.168.10.0 - 192.168.10.63$
 - $192.168.10.64 - 192.168.10.127$
 - $192.168.10.128 - 192.168.10.191$
 - $192.168.10.192 - 192.168.10.255$

Sample Exercise 2

Given the Class C network of 204.15.5.0/24, subnet the network in order to create the network in Figure 2 with the host requirements shown.

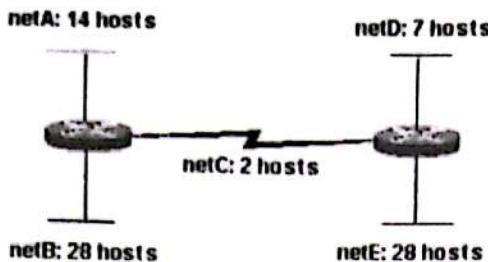


Figure 2

Looking at the network shown in Figure 3, you can see that you are required to create five subnets. The largest subnet must support 28 host addresses. Is this possible with a Class C network? And if so, then how?

You can start by looking at the subnet requirement. In order to create the five needed subnets you would need to use three bits from the Class C host bits. Two bits would only allow you four subnets (22).

Since you need three subnet bits, that leaves you with five bits for the host portion of the address. How many hosts does this support? $2^5 = 32$ (30 usable). This meets the requirement. Therefore, you have determined that it is possible to create this network with a Class C network.

An example of how you might assign the subnetworks is:

netA: 204.15.5.0/27 host address range 1 to 30	excluding network address 0 and broadcast address 31.
netB: 204.15.5.32/27 host address range 33 to 62	excluding network addr. 32 and broadcast Addr. 63.
netC: 204.15.5.64/27 host address range 65 to 94	excluding network addr. 64 and broadcast Addr. 95.
netD: 204.15.5.96/27 host address range 97 to 126	excluding network addr. 96 and broadcast Addr. 127.
netE: 204.15.5.128/27 host address range 129 to 158	excluding network addr. 128 and broadcast Addr. 159.

D Determine the followings for a given IP address. 175.231.232.116/24
converting the ip to bit level. (the subnet in bit level).

175	231	232	116
1111111111111111111111111111000000			

1. Subnet: 255 . 255 . 255 . 224 (from 3(111) bit subnet bits)

2. Block size = $256 - 224$ (last octet) = 32 or 2^{5-0} available ip addresses.

3. Subnet bits are 3. thus NO of subnets are $2^3 = 8$.

4. Number of hosts = $2^5 - 2$ no. of 1's in last octet. = $2^5 - 2 = 30$

5. For Each subnet's the Network IP, Broadcast IP and first and last valid IP addresses are determined below.

Since the first three octet's subnet value is 255.255.255, the four octet part and their corresponding values are shown in the table.

P.T.O.

	subnet 1	subnet 2	subnet 3	subnet 4	subnet 5	subnet 6	subnet 7	subnet 8
Subnet ID	.0	.32	.64	.96	.128	.160	.192	.224
First valid Host.	.1	.33	.64	.97	.129	.161	.191	.225
Last valid Host	.30	.62	.94	.126	.158	.190	.222	.254
Broadcast Address.	.31	.63	.95	.127	.159	.191	.223	.255

- ③ The ip '175.231.232.116' falls in which subnet?
 ⇒ The last octet's decimal value is **[.116]** that falls under Subnet-4 (.97 ~ .126).

Required Formulae:

1. **Subnet Mask:** convert the prefix ($/n$) to binary with **n** ones (1) and the rest as zeros, then convert to dotted decimal format.
2. **Network Bits:** Equal to prefix length $/n$.
3. **St Bits:** $32 - n$; where n is prefix length.
4. **Subnet Bits:** Total bits used for Subnetting = n - default Network bits.
 $(e.g. /27 \rightarrow [27 - 24 = 3])$
class C
default bits
5. **Hosts per subnet:** $2^{\text{host bits}} - 2$ (subtracting Network & Broadcast Address).
6. **Number of Subnets:** $2^{\text{subnet bits}}$ ($e.g. /27 \rightarrow 2^3 = 8 \text{ subnets}$).
7. **First valid IP:** Network address + 1
8. **Broadcast address:** Network address + $2^{\text{host bits}} - 1$
9. **Last valid IP:** Broadcast Address - 1

For Verification of the Calculated Result, Visit:

<https://www.subnet-calculator.com>

Segment Structure:

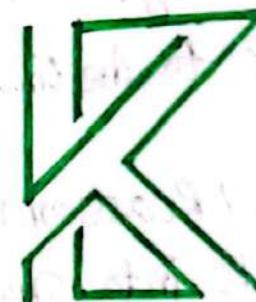
IPv4:

- Version (4 bits): Indicates IP version 4.
- Header length (4 bits): Length of the header.
- Types of service (8 bits): QoS specification.
- Total Length (16 bits): full packet size header + data
- Identification, flags, fragment offset: for fragmentation and re-assembly.
- TTL (8 bits): Limits packet lifetime.
- Protocol (8 bits): Specifying upper layer protocol.
- Header checksum: Error checking of header.
- Source IP (32 bits) & Destination IP (32 bits)
- options (if any) + padding.

IPv6:

- Version (4 bits): Indicates IP version 6.
- Traffic class (8 bits): priority/QoS.
- Flow label (20 bits): for real time traffic flow ID.
- Payload length (16 bits): Length of the data part.
- Next header (8 bits): specifies upper layer protocol like (TCP/UDP).
- Hop limit (8 bits): similar to TTL in IP V4.
- Source (128 bits) & Destination (128 bits) Addresses.

IPV4 Header								
32 bits								
VER 4 bits	HEC 4 bits	Service Type 8 bits		Total Length 16 bits		Fragment offset 13 bits		
Datagram Identification 16 bits		Flags 3 bits		Header Checksum 16 bits				
Time To Live 8 bits	Protocol 8 bits		Source IP Address 32 bits		Destination IP Address 32 bits		Options	



IPV6 Header			
Version (4-bits)	Priority/Traffic class (5-bits)	Flow Label (20-bits)	
Payload Length (16-bits)		Next Header (8-bits)	Hop Limit (8-bits)
Source IP Address (128-bits)			
Destination IP Address (128-bits)			
Extension Headers (1.....n) (128-bits)			
Data/Payload			

What's missing (compared with IPv4):

- no checksum (to speed processing at routers)
- no fragmentation/reassembly
- no options (available as upper-layer, next-header protocol at router)

① Transition from IPv4 to IPv6:

IPv4 has limited (32-bit) has limited ~43 billion addresses, IPv6 offers vastly more. Growing Internet device led to IPv4 depletion.

IPv6 benefits: Larger address space, improved routing, built-in security (IPsec), and simplified headers.

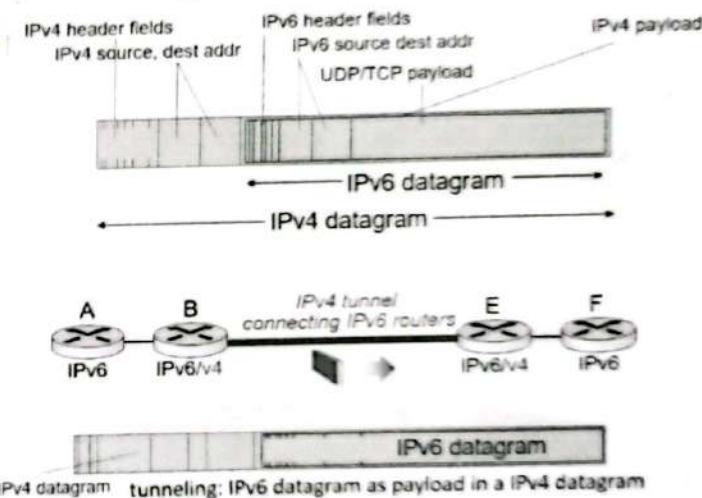
- Dual stack: Devices run both IPv4 and IPv6 simultaneously to ensure compatibility.
- Tunneling: IPv6 packets are encapsulated in IPv4 for networks that only support IPv4.
- Translation NAT64/ DNS64: Converts IPv6 to IPv4 and vice versa to enable communication.
- Gradual Deployment: Full IPv6 adoption is ongoing; both protocols co-exist.

② Tunneling:

Tunneling is a method of encapsulating packets of one network (IPv6) protocol within another protocol (IPv4).

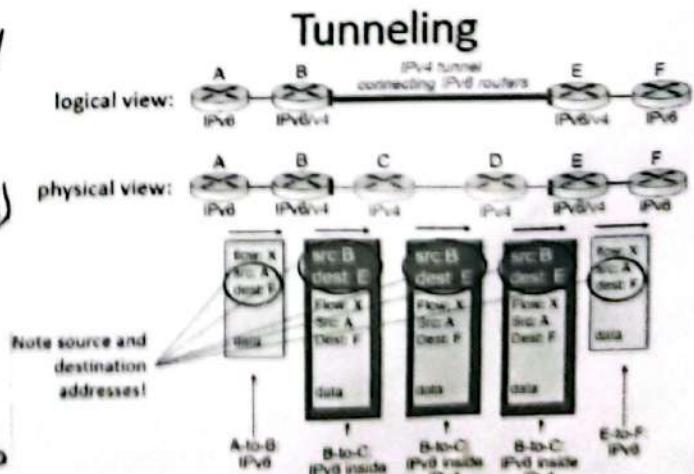
Purpose:

- Enables transmission of IPv6 packets over IPv4 only networks.
- Supports incompatible protocol communication.
- Ensures secure communication over untrusted networks (e.g. VPNs).



Mechanism:

- The original packet (e.g. IPv6) is wrapped inside another packet (e.g. IPv4).
- Sent through the tunnel (e.g. via internet).
- At the destination the encapsulated packet is unwrapped-decapsulated and forwarded normally.
- Common techniques 6to4, Teredo, ISATAP and CoRERF



DHCP:

Dynamic Host Configuration protocol automatically assigns IP addresses and other network configurations (like DNS, gateway) to devices on a network.

DHCP Registration Process: DORA

Discover: client broadcasts a DHCP Discover message to find available DHCP servers.

Offer: DHCP server responds with a DHCP Offer containing an available IP and configuration info.

Request: Client replies with a DHCP Request message, accepting the offer.

Acknowledgment: Server sends a DHCP ACK confirming the lease of IP and setting.

Why DHCP is used in any network where hosts are frequently changing?

- Automatic IP assignment - no manual configuration needed.
- Efficient management of limited IP addresses via leasing.
- Reduces errors from manual setup.
- Faster device onboarding - ideal for networks with many mobile or temporary devices.
- Centralized control - admin can manage IP settings from one place.

DHCP client-server scenario

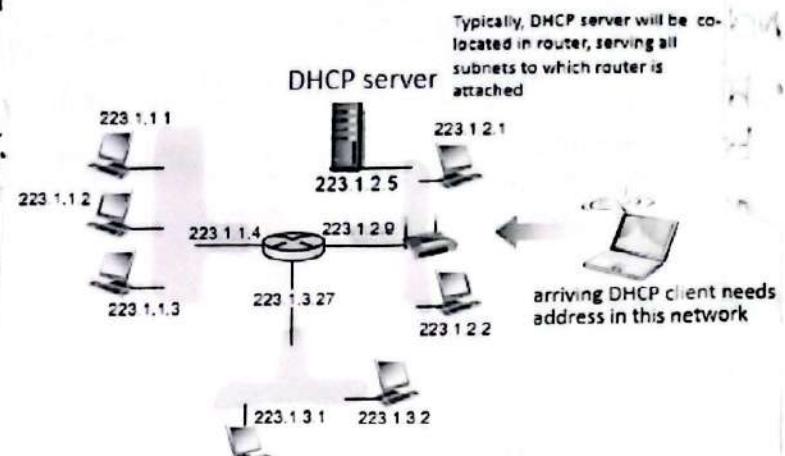


Fig 1: New device connecting using DHCP

DHCP client-server scenario

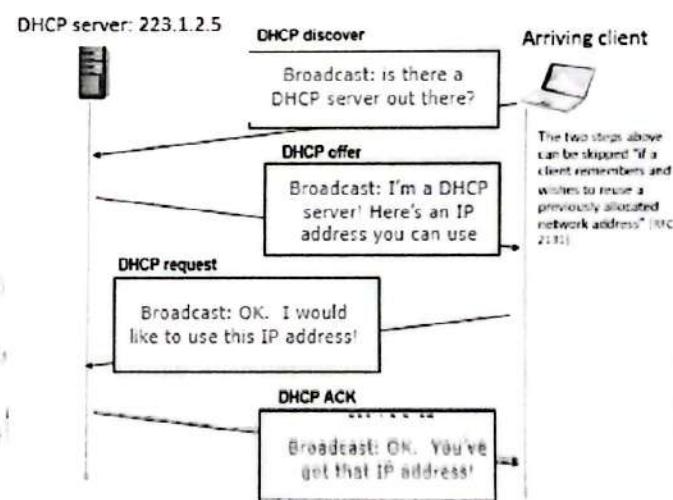
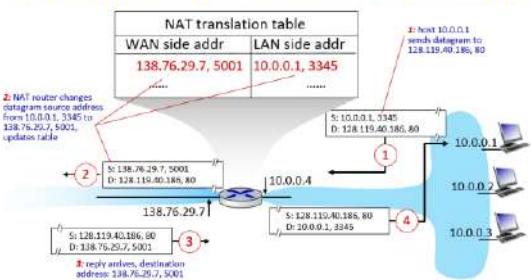


Fig 2: DHCP Registration process.

NAT:

Network Address Translation is a technique used to map private IP addresses inside a local network to a single public IP address (or a few) for communication with external networks, like the internet.

NAT: network address translation



Purpose of using NAT Table:

- IP address conservation: Allows multiple devices to share one public IP.
- Security; Hides internal IPs from the outside world.
- Routing: Keeps tracks of active connections with source/destination IP and port mappings.
- Session management: Ensures return traffic reaches the correct internal device.

NAT table maintains translation entries (private \rightarrow public) for each connection session.

Working:

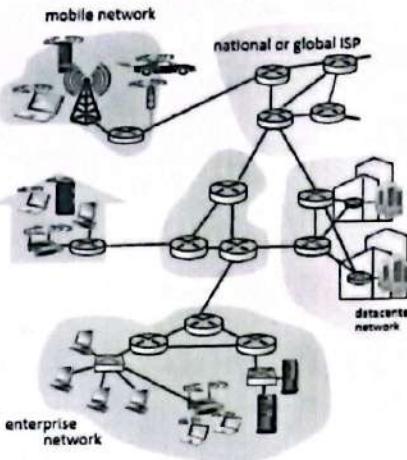
- A device in a private network sends data to the internet.
- NAT-enabled router replaces the private IP with its public IP and assign a unique port number.
- It records this mapping in the NAT Table to translate the public IP and port back to the correct private IP and port.
- The response is delivered to the correct internal devices.

The Link Layer:

The layer 2 of the OSI model, responsible for node to node communication over a physical link, converts raw bits from the physical layer into frames.

Key Services:

- Framing: groups bits into frames.
- Addressing: Uses MAC address to identify source and destination.
- Error-detection and correction: Detects and optionally corrects transmission errors.
- Flow control: Prevents fast sender from overwhelming slow receiver.
- Access control: Determines which device can use the link especially in shared media.



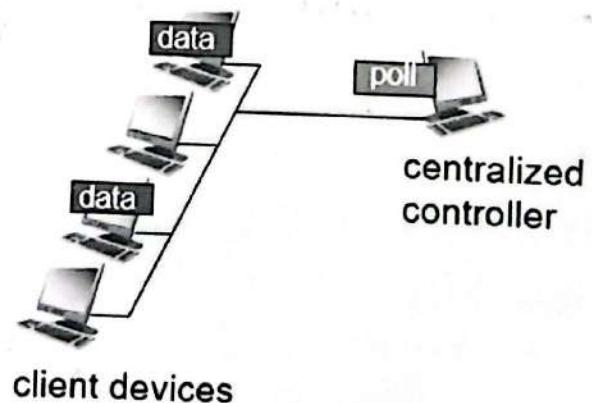
Common protocols: are Ethernet, PPP (Point-to-point protocol), HDLC, WiFi (802.11).

MAC Protocol:

A set of rules in the data link layer that controls how devices access and share the communication media.

Media Access Control:

- Prevents collision.
- Ensure efficient and fair access to the channel.
- Co-ordinates who sent data and when.



Devices use protocols to sense or reserve medium before transmitting.

Ethernet Frame Structure:

Ethernet frame structure				
type	preamble	dest address	source address	data (payload)
				CRC

An ethernet frame is the basic unit of data transmission in Ethernet networks. It includes the following fields:

1. Preamble: (7 bytes) - synchronizes sender and receiver clocks.
2. Destination Address: (6 bytes) MAC address of receiver.
3. Source Address: (6 bytes) MAC address of sender.
4. Data (Payload): () - Actual data being carried.
5. CRC: (4 bytes) Error checking (cyclic redundancy check).

Slotted ALOHA:

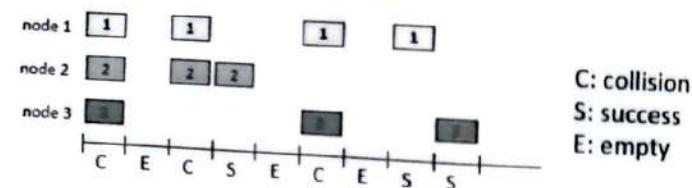
Slotted ALOHA is a MAC protocol used to control access to a shared communication channel.

Working:

- Time is divided into equal slots.
- A device can only transmit at the start of a time slot.
- If two or more nodes transmit in the same slot \rightarrow collision occurs.
- After a collision nodes wait a random number of slots before retrying (backoff).
- Reduces collision chances compared to pure ALOHA.

Maximum efficiency is 37%.

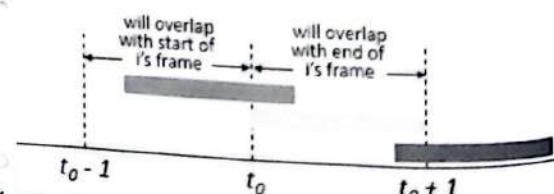
Slotted ALOHA



PURE ALOHA:

Working:

- Devices transmit any time without checking the channel.
- If collision occurs (overlapping frames), both packets are destroyed.
- Sender waits a random time (backoff) and re-transmits.
- No time slots - transmission can begin at any time.
- Higher chance of collision compared to slotted ALOHA.



Mention about the efficiency of slotted ALOHA:

Let N be the number of nodes each transmitting with per slot.

Probability that a node transmits successfully in a slot.

$$P_{\text{success}} = P(1-P)^{N-1}$$

Probability that any one of nodes transmits successfully:

$$\eta = N P(1-P)^{N-1}$$

To find maximum efficiency, optimizing η max when $P = \frac{1}{N}$
taking limit as $N \rightarrow \infty$ then $N-1 \approx N$

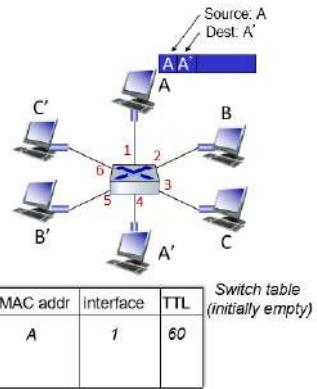
$$\eta_{\text{max}} = \lim_{N \rightarrow \infty} N \cdot \frac{1}{N} \cdot \left(1 - \frac{1}{N}\right)^N = \frac{1}{e} \approx 0.37\% \quad \begin{bmatrix} \text{using,} \\ \lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n = \frac{1}{e} \end{bmatrix}$$

that yields only 37% of slots carrying successful transmission in the best case.

✓ ARP:

Address Resolution protocol maps an IP address to MAC address in a Local Network (LAN)

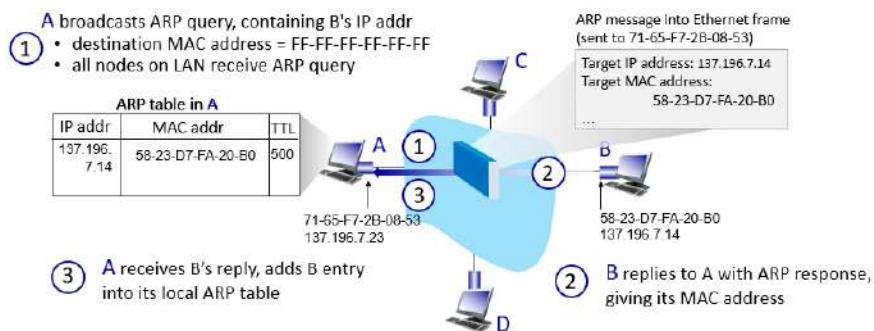
- When a device wants to communicate but knows only the IP address, it sends an ARP request (broadcast).
- All devices receive it, but only the device with the matching IP replies with an 'ARP Reply' (unicast) providing its MAC address.
- The sender stores this IP-MAC pair in the ARP Cache for future use.



ARP protocol in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

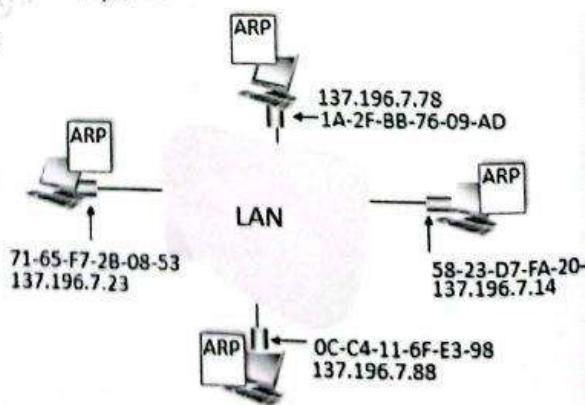


✓ ARP addressing for Routing to another LAN:

When a host sends data to a device outside its LAN, it needs the MAC address of the default gateway (router), not the destination host.

- Host uses ARP to find the MAC address of the router's interface (next-hop) on the same LAN.
- ARP request: Who has IP of default gateway.
- Router replies with its MAC address.

Host sends the packet to the router's MAC, and the router forwards it toward the destination LAN based on its routing table.



REPEATER:



Figure 1: A Repeater

A repeater is a network device that regenerates and amplifies signals to extend the transmission distance in a network. It operates at the physical layer of the OSI model and is used to overcome attenuation or signal degradation. Repeaters do not process data; they simply boost the existing signal without altering the content. They are ideal for connecting two segments of a LAN or extending the reach of wireless networks. A common Ethernet rule is that no more than four repeaters may be used to join segments together. This is a physical limitation designed to keep collision detection working properly. Repeaters operate at layer 1 (Physical layer) of the OSI model.

BRIDGES:



Figure 2a: A Typical Bridge

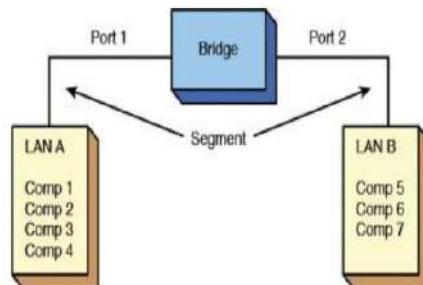


Figure 2b: Working of a Bridge

A bridge is a network device that connects two or more LAN segments and filters traffic based on MAC addresses. It operates at the data link layer and reduces network congestion by segmenting traffic. Bridges forward or block packets depending on the destination address, ensuring efficient communication. They are ideal for expanding networks without broadcasting unnecessary traffic.

When network grows in size, it is often necessary to partition it into smaller group of nodes to help isolate traffic and improve performance. One way to do this is to use bridge, the operation of it is to keep one segment traffic to that side and other side will cross the bridge. The bridge learns which packets should cross it as it is used.

ROUTERS:



Figure 3: A Typical Router

A router is a network device that connects different networks and directs data packets based on IP addresses. Operating at the network layer, routers determine the best path for data using routing protocols like RIP, OSPF, or BGP. They enable communication between networks with different architectures, such as LANs and WANs, and support features like NAT and DHCP.

The router examines each packet of information to determine whether the packet must be translated from one network to another network performing a function similar to a bridge. Unlike a Bridge, a router can connect networks that use different technologies, addressing methods, media type, frame format, and speeds. A router is a special-purpose device designed to interconnect networks. Such that three networks can be connected using two routers. Routers maintain routing tables in their memories to store information about the physical connection on the networks; the router examines each packet of data, checks the routing table, and then forwards the packet if necessary. Every other router in the path (between any state destinations) performs a similar procedure. Note that a router does not maintain any state information about the packets; it simply moves them along the network. Routers are operated at layer 3(network) of OSI model.

GATEWAYS:

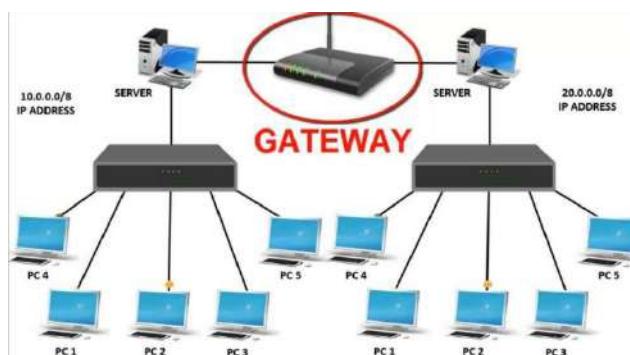


Figure 4: Gateway in a Network

A gateway is a network device that connects two networks with different protocols, enabling communication between them. It operates at multiple layers, from transport to application,

depending on the implementation. Gateways perform protocol translation, data conversion, and application compatibility. They are essential in heterogeneous network environments like connecting enterprise networks to the internet.

It is also known as a computer system located on earth that switches Gateway data signals and voice signals between satellites and terrestrial networks. A gateway can operate at any layer of the OSI or TCP/IP reference model. The hub of a gateway, also called a protocol converter, is much more complex than that of a router or switch. Typically, a gateway must convert from one protocol stack to another. E.g. a gateway may connect a TCP/IP network to an IPX./SPX network. A Circuit Level Gateway function provided by Application level gateway products enables trusted users on private network to access Internet services with all security of a proxy server. An Application Level Gateway provide much stricter form of security than packet filters, but they are designed to regulate access only for a particular application.

HUBS:



Figure 5a: A Typical HUB

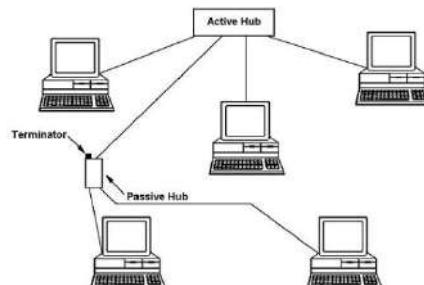


Figure 5b:Types of HUB

A hub is a simple networking device that connects multiple devices in a LAN. Operating at the physical layer, it uses a broadcast mechanism, sending data packets to all connected devices regardless of the destination. Hubs are less efficient compared to switches as they cannot differentiate between devices, leading to network collisions. They are inexpensive but unsuitable for modern, complex networks due to their lack of intelligence.

There are three types of HUB passive hub, active hub and intelligent hub. The Passive hubs do not process data signals with only purpose to combine the signal from several networks cables segments. All devices attached to the passive hub receive another packets that pass through the hub. Hub does not clear up or amplify the signals, on the contrary absorbs a small part of the signals that is why the distance between a hub and a computer should not be more than half of the permissible distance between two computers. Passive hubs have limited functionality so are inexpensive and easy to configure. It has four ports with four BNC (British Naval Connectors) female connectors to configure networks station or terminated with a 93Ω BNC Terminator.

The active hubs incorporate electronic components that amplify and cleanup the signals, that flaw between devices on the network. The process of cleaning up signal is called “signal regeneration”. The benefits of signals regeneration are:

- A network is more robust i.e. less sensitive errors.
- Distance between devices can be increased.

Active hubs cost is considerably more than passive hub (active hub function imparts as multiport repeaters).

Intelligent hubs are enhanced active hubs the following functions add intelligence to a hub. Intelligent Hubs are units have form of integrated management capability. Hub Management A hub supports networks network management protocols that enable the hub to send packets to central network console. These protocols enable network console to manage or control hub. Switching hubs Switching hubs include circuitry that quickly routes signals between ports on the hub. Insured of repeating a packet to all ports on the hub, it repeats a packet only to the port that connects to the destination computer for the packet.

SWITCHES:



Figure 6:A Typical Switch.

A switch is a more advanced network device than a hub, operating at the data link layer of the OSI model. It uses MAC addresses to forward data packets to the correct destination, significantly reducing network collisions. Switches support multiple simultaneous connections and can operate in full-duplex mode. They are critical in modern networks for efficient communication and enhanced performance.

It is similar to a bridge, with some important enhancement. First, as which may have multiple ports, thus directing packets to several different segments further partitioning and isolating network traffic in a way similar to router. For example, if 8-port n way switch is there it can route packets from any input to any output. Some or all of incoming packet is called store and forward, which stores the received packet before examining it for error before retransmitting. Bad packets are not forwarded. A switch typically has auto-sensing 10/100 mbps ports and will set the speed of each port accordingly; furthermore, a managed switch supports SNMP for further control over network traffic. Switches operated at layer 2 (Data Link) of OSI model.

🕒 Why is IPv6 the Candidate for Next-Generation Networking?

- **Larger Address Space:**
Supports 2^{128} addresses compared to IPv4's 2^{32} — solves address exhaustion.
- **Simplified Header Format:**
More efficient routing due to fixed-length headers.
- **Built-in Security:**
IPSec is mandatory in IPv6, enhancing data confidentiality and integrity.
- **Better Multicasting & Mobility Support:**
Native support for multicast and mobile devices.
- **No Need for NAT:**
Global addressing eliminates the complexity of Network Address Translation.
- **Auto-Configuration:**
Supports stateless address autoconfiguration (SLAAC), simplifying network setup.

Teacher Suggested Question

DHCP (Dynamic Host Configuration Protocol)	VS	DNS (Domain Name System)
Assigns IP addresses to hosts automatically		Translates domain names to IP addresses
Works during network joining or boot-up		Works during web access or name resolution
Provides other config info (e.g., gateway, DNS server)		Only resolves names to IPs and vice versa
Uses UDP ports 67 (server) and 68 (client)		Uses UDP/TCP port 53
IP addresses can be dynamic (temporary)		DNS mappings are usually static or cached
Managed by network admin or DHCP server		Managed by DNS server or registrar

🕒 Q: What are the applications of Mobile IP?

To the point:

- **Seamless roaming** across different networks without changing IP address.
- **Maintains ongoing sessions** during movement (e.g., VoIP, video calls).
- **Used in mobile workforce networks**, like delivery fleets or remote workers.
- **Supports IoT devices** moving across networks.
- **Enables mobile access in public transport**, smart cities, etc.

Less Important, but encountered in the previous year.

[To Learn About Mobile IP, Click me.](#)

✓ **Q: What is the Mobile IP registration process when a mobile node moves to a new network?**

To the point:

1. Mobile Node (MN) detects new foreign network.
2. Foreign Agent (FA) sends advertisement to MN.
3. MN sends **Registration Request** to FA.
4. FA forwards request to **Home Agent (HA)**.
5. HA updates location info and replies with **Registration Reply**.
6. FA forwards the reply to MN — registration is complete.

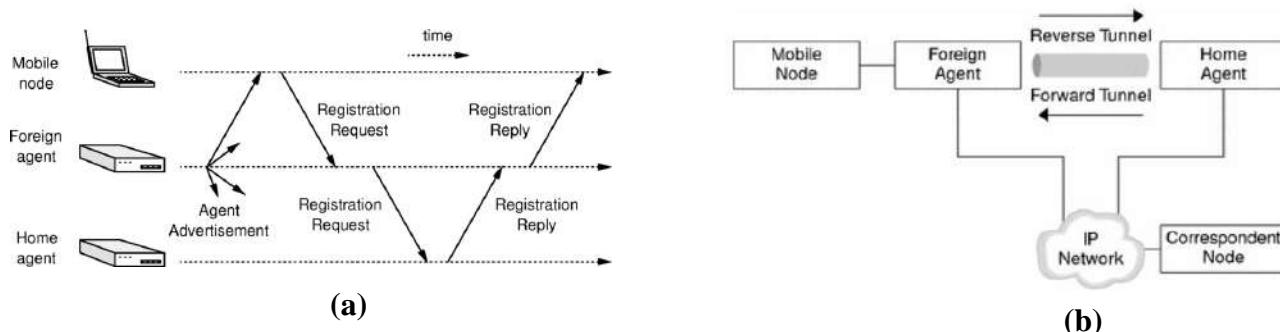


Figure 1: (a) Mobile IP registration Process (b) Reverse Tunneling.

✓ **Reverse Tunneling in Mobile IP Registration (to the point, under 100 words):**

- Reverse tunneling sends packets from the Mobile Node (MN) to the Home Agent (HA) via the Foreign Agent (FA).
- Tunnel is created **from FA to HA** (reverse path).
- Ensures packets appear to originate from the **home network**.
- Helps with **firewalls, ingress filtering**, and routing policies that block foreign-sourced packets.
- Used when direct sending from foreign network would be blocked or misrouted.
- Reverse tunnel is set up during **registration request phase**.

Q: What do you mean by Mobile SP (Service Provider) and Mobile Computing?

Mobile SP (Service Provider):

- A **Mobile Service Provider** offers wireless communication services (e.g., voice, SMS, internet) to mobile users.
- Manages network infrastructure like **cell towers, base stations, and data centers**.
- Examples: AT&T, Vodafone, T-Mobile.

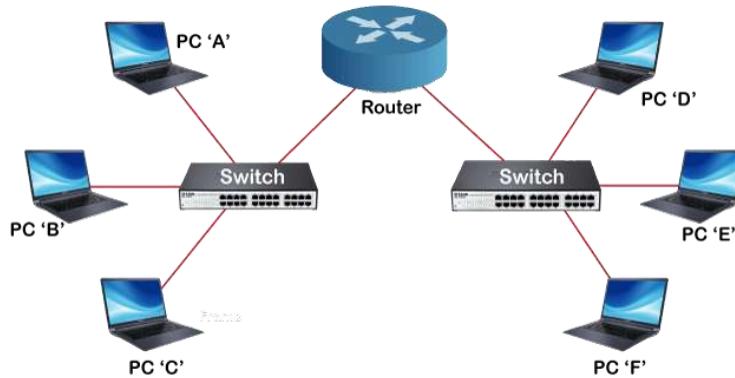
Mobile Computing:

- The ability to **use computing devices** (e.g., smartphones, laptops, tablets) while **being mobile**.
- Supports wireless access to data, applications, and services anytime, anywhere.
- Enables activities like emailing, browsing, cloud access, and remote work.



FTP (File Transfer Protocol) – To the Point:

- **Purpose:** Transfers files between client and server over a network.
- **Uses:** Uploading/downloading files, website management, backups.
- **Runs on TCP:** Reliable data transfer.
- **Ports:**
 - **Command control:** Port 21
 - **Data transfer:** Port 20 (active mode) or random port (passive mode)
- **Modes:**
 - **Active:** Server initiates data connection.
 - **Passive:** Client initiates both connections (better with firewalls).
- **Authentication:** Supports login with username/password or anonymous access.
- **Not secure by default** (use FTPS/SFTP for encryption).



Switch

Connects devices within the same network (LAN).

Operates at Data Link Layer (Layer 2).

Uses MAC addresses to forward data.

No IP address is assigned to ports.

Mainly used for internal communication.

Faster in data transmission within LAN.

Doesn't perform NAT.

VS

Router

Connects multiple networks (LAN to WAN).

Operates at Network Layer (Layer 3).

Uses IP addresses to route data.

Each interface has an IP address.

Used to connect to the internet or other networks.

Slower due to routing overhead.

Performs NAT to share a public IP.

ARP (Address Resolution Protocol)

Maps IP addresses to MAC addresses

Works within a local network (LAN)

Operates at Layer 2 & 3 (Link & Network)

Used for host-to-host communication

No path selection or route calculation

Examples: Standard ARP

VS

Routing Protocols

Determine best path for data between networks

Work across multiple networks (WAN/LAN)

Operate at Layer 3 (Network Layer)

Used for router-to-router communication

Calculate and update routing tables

Examples: RIP, OSPF, BGP, EIGRP



Q: ARP is a Plug and Play Protocol. Justify your answer (to the point, under 100 words):

ARP is called a Plug and Play protocol because it works automatically without manual configuration. When a device joins a network, it can dynamically discover the MAC address of another device using only its IP address. This eliminates the need for pre-configured address mappings. ARP handles the resolution process in real time, enabling seamless communication within a local network. It simplifies network setup, making it ideal for dynamic and user-friendly environments.

Want an example showing ARP in action right after connecting a new device?

Q: What are the design requirements of PPP (Point-to-Point Protocol)?

To the point:

- **Framing mechanism** for encapsulating network layer packets.
- **Link establishment and termination** procedures.
- **Support for multiple network layer protocols** (e.g., IP, IPX).
- **Authentication** (optional) using PAP or CHAP.
- **Error detection** using CRC.
- **Address and control field compression** (optional).
- **Link Quality Monitoring** to check link reliability.
- **Negotiation mechanisms** using LCP (Link Control Protocol) and NCP (Network Control Protocol).



Q: What is Byte Stuffing? Explain Briefly.

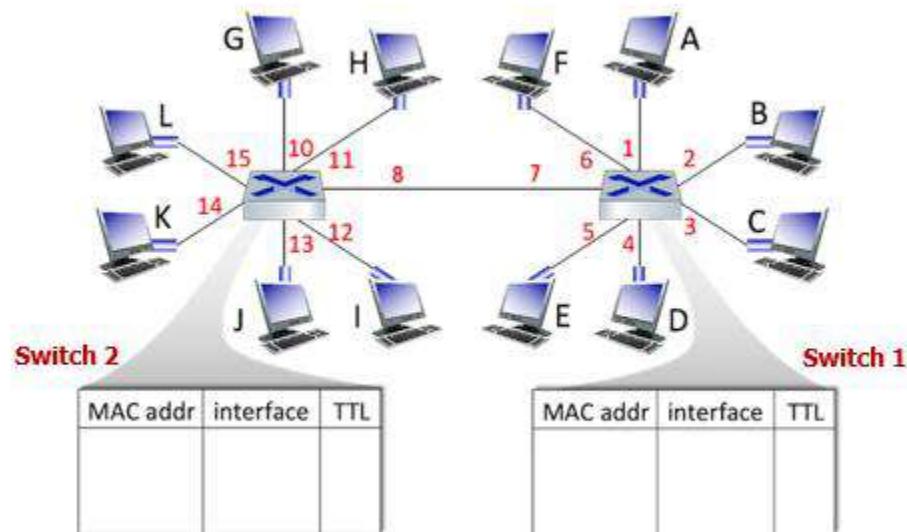
To the point:

- **Byte stuffing** is a technique used in data communication to **distinguish data from control information** (like frame delimiters).
- Special **flag bytes** (e.g., `01111110` in HDLC) mark the **start and end** of a frame.
- If the **flag byte appears in data**, it is "stuffed" with an **escape byte** (e.g., `ESC`) to avoid confusion.
- The receiver **removes** the escape characters during **destuffing** to restore original data.
- Ensures **data transparency** while maintaining frame boundaries.

Example:

If flag = `F`, ESC = `E`,

Data = `A F B` → Stuffed = `A E F B`



Q: Describe the self-learning process of a switch briefly.

To the point:

- When a switch receives a frame, it reads the **source MAC address** and **source port**.
- It **stores this info** in its **MAC address table** (MAC → Port).
- This process is called **self-learning**.
- The switch then uses this table to **forward future frames** directly to the correct port, instead of broadcasting.
- If the destination MAC is **unknown**, it **floods** the frame to all ports except the source.
- This improves **network efficiency** over time.

Want an example showing how a switch updates its MAC table?



What is a Virtual Circuit (VC)?

A Virtual Circuit is a logical path established between sender and receiver in a packet-switched network, where all packets follow the same path and are delivered in order—like a "virtual" dedicated connection.

VC Implementation (To the Point):

- **Setup Phase:** Before data transfer, a path is established (VC number, path info stored in routers).
- **Data Transfer Phase:** Packets carry VC identifier (not full source/dest IP); routers forward based on this.
- **Teardown Phase:** After data transmission, VC is removed from routers.

Example:

Router A assigns VC #12 from Host X to Host Y → Router B forwards packets with VC #34 → Router C uses VC #56 to deliver to Host Y. All routers maintain VC tables mapping incoming to outgoing VC numbers.

Network Security:

Network security is the practice of protecting data, devices and systems in a network from unauthorized access, misuse or attacks. It involves tools like firewalls, encryption, intrusion detection and prevention systems and access control.

Ensures Confidentiality, Integrity and Availability of information. prevents data breaches, malware infections and unauthorized access.

The Network security is essential for business continuity, user trust and legal compliance. Protects against threats like hackers, phishing and Dos attacks.

Without network security sensitive data and network services are vulnerable to theft, disruption or damage.

Desirable Security Properties:

1. Confidentiality: Ensures only authorized users can access data.

Example: Leaked email during a hack - Confidentiality lost.

2. Integrity: Guarantees data is not altered during transmission.

Example: Modified bank transaction amounts - integrity lost.

3. Availability: Ensures system are available and accessible when needed.

Example: Web-site down due to d-dos Attack Availability lost.

4. Access Control: Limits who can view or use resources

Example: Unauthorized login to private files - access control failed.

Cryptography:

Cryptography is the science of securing information by converting it into a coded format, so only authorized parties can understand it.

It protects data confidentiality, integrity and authenticity. In the digital era where online communication, banking and data sharing are widespread, cryptography is essential to defend against cyber threats, identity theft and data breaches. (For equation see the fig).

Symmetric Key Cryptography uses the same key for both encryption and decryption. It is fast and suitable for large data but requires secure key sharing.
No. of key \leq no. of users.

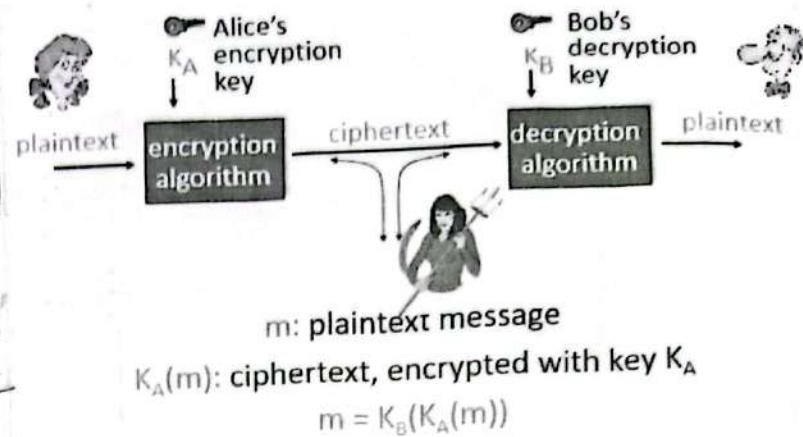
Asymmetric Key Cryptography

uses receiver's public key for encryption and receiver use his/her private key for decryption. It ensures secure communication without prior key sharing but is slower. No of key $2^n >$ no. of user

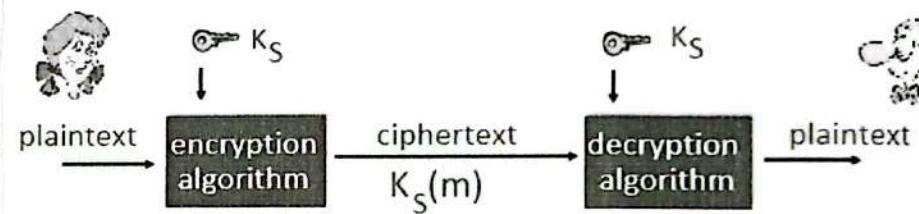
Symmetric

- same key for both ends
- fast and efficient
- key must be shared securely
- used in bulk data encryption.
- Example : AES, DES.

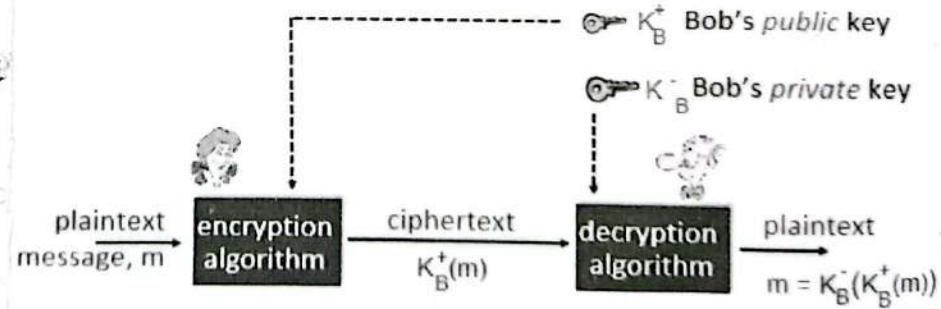
The language of cryptography



Symmetric key cryptography



Public Key Cryptography



Asymmetric

- Public key to encrypt, private key to decrypt.
- Slower due to complex algorithm.
- No need to share private key.
- Used in digital signature, key exchange.
- Example : RSA, ECC

✓ DES:

Data Encryption Standard is a symmetric key block cipher used for encrypting electronic data.

- Encrypts data in 64-bit blocks using a 56-bit key.
- Performs 16 rounds of encryption using substitution and permutation.
- Involves initial permutation (IP) and final permutation (FP).
- Each round uses a 48-bit round key generated from the main key.
- Uses Feistel structure: data split into left and right halves processed with round functions.
- Now considered insecure for sensitive data due to short key length and brute-force vulnerability.

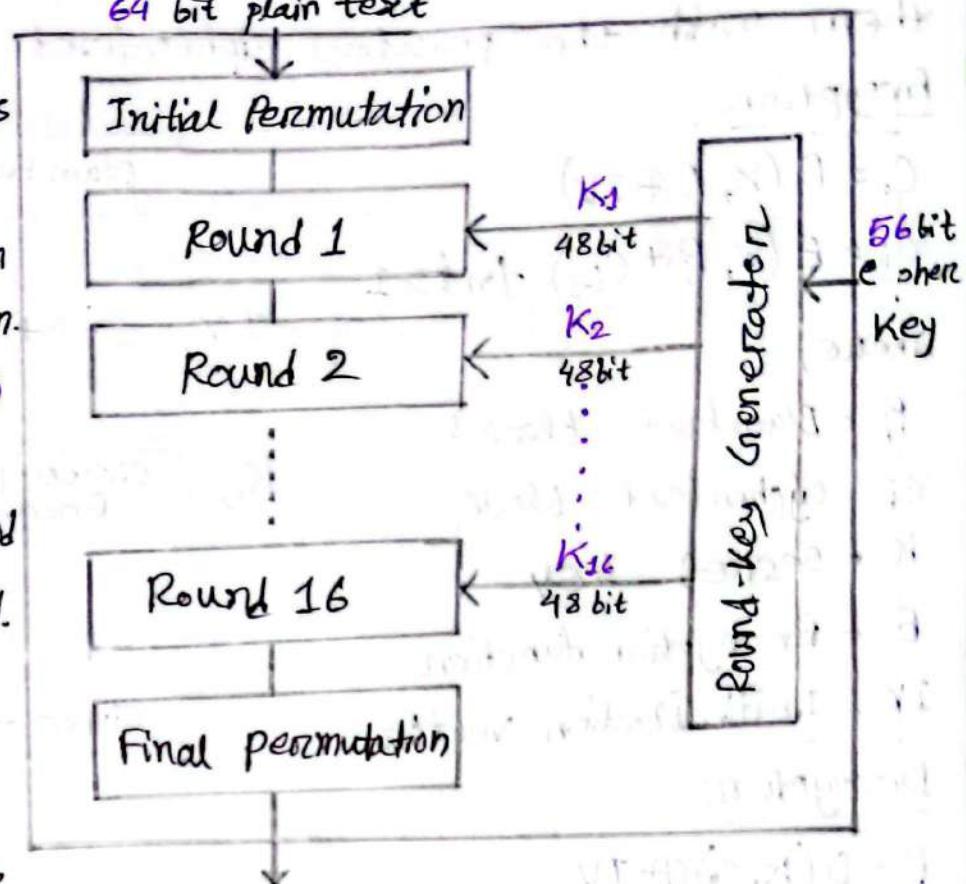


fig: Block diagram of DES cipher.

✓ Single round DES:

1. Input 64 bit block - divided into two halves L_j & R_j each 32 bit.
2. Function f
 - Takes 32 bit R_j and 48 bit subkey K_{j+1}
 - Performs expansion, key mixing (\oplus OR), substitution and permutation.

3. Left output, $L_{j+1} = R_j$ The new left half is the previous right half.

4. Right output, $R_{j+1} = L_j \oplus f(R_j, K_{j+1})$

the new right half is obtained by \oplus ORing the previous left half with the output of function f .

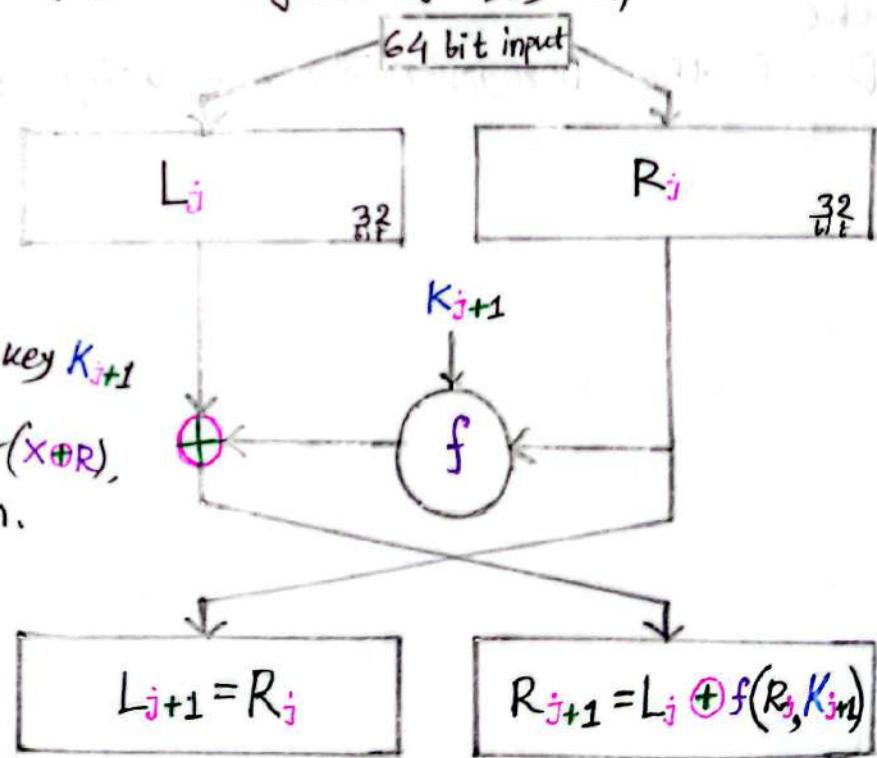


Fig 2: Single Round DES block diagram.

CBC:

Cipher Block Chaining mode encrypts plaintext blocks by chaining them with the previous ciphertext block to ensure randomness.

Encryption:

$$C_i = E(K, P_i \oplus IV)$$

$$C_i = E(K, P_i \oplus C_{i-1}) \text{ for } i \geq 1$$

where,

P_i - plaintext block i

c_i - ciphertext block

K - secret key

E - encryption function

IV - Initialization vector

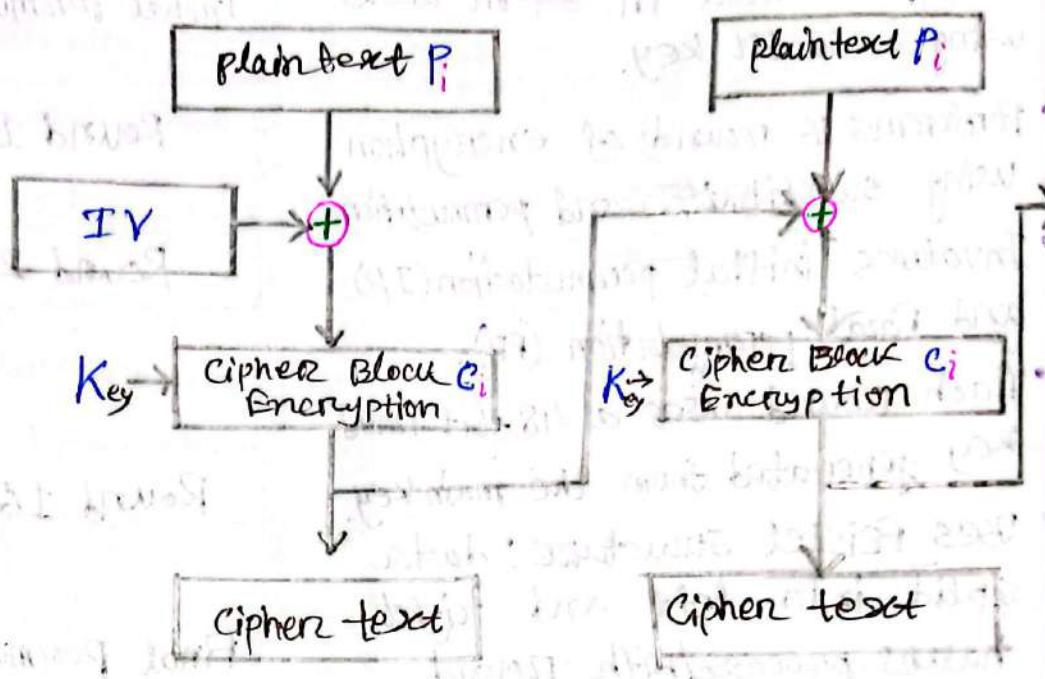
Decryption:

$$P_i = D(K, C_i) \oplus IV$$

$$P_i = D(K, C_i) \oplus C_{i-1} \text{ for } i \geq 1$$

where,

D - is the decryption function using the same key K .



RSA:

Rivest Shamir Adleman is an asymmetric encryption algorithm using two keys (aka public for encryption and private for decryption), based on the difficulty of factoring large prime numbers.

Applications:

- Secure email (e.g. PGP)
- Digital Signatures.
- Secure web browsing (HTTPS)
- Banking and e-commerce encryption.

Advantages:

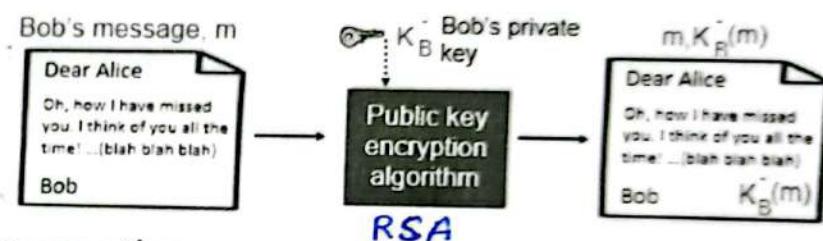
- High security for key exchange.
- Enables digital signatures.
- No need to share secret keys before hand.
- Ensures confidentiality and authentication.

Steps to solve RSA

1. Let p and q two distinct prime numbers. These are kept secret.
2. Compute $n = pq$, where n is the modulus used for both the public and private keys.
3. Compute Euler's Totient function $\varphi = (p-1) \times (q-1)$, used to key generation to find public and private exponents.
4. Choose public key exponent e must satisfy $1 < e < \varphi$ and $\text{GCD}(e, \varphi) = 1$ (i.e. e and φ are relatively prime).
5. Compute private key exponent d is the modular inverse of $(e \text{ mod } \varphi)$

$$d \times (e \text{ mod } \varphi) = 1 \quad \text{or} \quad d = \frac{1 + x\varphi}{e}$$

where, $x = 0, 1, \dots$ until d is found to be integer.
6. Public key (e, n) , private key (d, n)
7. To encrypt message m .
ciphertext, $c = m^e \text{ mod } n$
8. To decrypt message m .
plaintext/original message $m = c^d \text{ mod } n$



Q) Encrypt message 13 using RSA algorithm, where one prime number is 11.

Now,

$$n = p \times q = 7 \times 11 = 77$$

and,

$$\varphi = (p-1)(q-1) = 6 \times 10 = 60$$

Encryption, Public key $(e, n) = (7, 77)$.

$$\begin{aligned} C &= m^e \bmod n = 13^7 \bmod 77 \\ &= 62748517 \bmod 77 \\ &= 62 \end{aligned}$$

Decryption, finding d such that,

$$d \times (e \bmod \varphi) = 1$$

$$\text{or, } d = \frac{1 + x\varphi}{e}$$

where, $x = 0, 1, 2, \dots$ until d is found to be integer value.

$$d = \frac{1 + 5 \times 60}{7} = 43 \quad \text{For } x=5, d \text{ is found integer}$$

$$\text{Now, } m = C^d \bmod n = 62^{43} \bmod 77 \\ = 13$$

Analysis:

What if we take $q=3$ and $e=3$

$$n = 11 \times 3 = 33 \text{ and } \varphi = 2 \times 10 = 20$$

public key $(e, n) = (3, 33)$

$$C = m^e \bmod n = 13^3 \bmod 33 = 19$$

$$d = \frac{1 + x\varphi}{e} = \frac{21}{3} = 7$$

$$m = C^d \bmod n = 19^7 \bmod 33 \approx 13$$

Given,

$$p = 11$$

and,

$q = 7$ (taken another prime no.)

taken

$e = 7$ such that $\text{GCD}(7, 60) = 1$

given,

$$m = 13$$

mod calculation

$$62748517 \div 77 = 814915.8052$$

$$0.8052 \times 77 \approx 64$$

Lesson, take less value and make encryption-decryption easy.

Encrypt message 19 using RSA where one prime numbers is 13.

$$n = p q = 13 \times 3 = 39; z = 12 \times 2 = 24$$

public key $(e, n) = (3, 39)$,

cipher text, $C = m^e \bmod n = 19^3 \bmod 39 = 34$.

decrypted text,

$$m = C^d \bmod n = 34^{24} \bmod 39 = 19$$

$$p = 13$$

$$q = 3 = e$$

$$m = 19$$

$$d = \frac{1+z}{e}$$

$$= \frac{72}{3} = 24$$

Consider RSA with $p=5, q=7$ (i) What is n and z (ii) Let e be 5 Why is this acceptable choice for e (iii) Find d and (iv) Encrypt and decrypt the message is 12.

(i) $n = 5 \times 7 = 35$, and $z = 4 \times 6 = 24$

(ii) e is chosen to be 5, this is acceptable because e and z are relatively prime to each other. $\text{GCD}(24, 5) = 1$.

(iii) For, $d = \frac{1+z}{e} = \frac{1+24}{5} = 5$

(iv) Encryption,

$$C = m^e \bmod n = 12^5 \bmod 35 = 17$$

Decryption,

$$m = C^d \bmod n = 17^5 \bmod 35 \approx 12$$

Consider RSA with $p=5, q=11$ (i) what are n and z (ii) Let e be 3 and explain why it is acceptable (iii) find d (iv) encrypt the message 8 using the key (n, e) . let C denote the cipher text. (v) Decrypt the message.

(i) $n = 5 \times 11 = 55$; $z = 4 \times 10 = 40$ (ii) $e = 3$ because $\text{GCD}(40, 3) \approx 1$

(iii) $d = \frac{1+z}{e} = \frac{81}{3} = 27$ [$z=2$]; (iv) $C = m^e \bmod n = 8^3 \bmod 55 = 17$

(v) $m = C^d \bmod n = 17^{27} \bmod 55 = 8$

Hash Functions:

A hash function is a cryptographic algorithm that takes an input m and produces a fixed-size string of characters, called a hash or digest. It is widely used for data integrity and digital signature.

Properties:

- Deterministic: Same input always produces same output.
- fast computation: quick to compute # for any input.
- Pre-image resistance: Hard to reverse the hash to get original input.
- collision resistance: Hard to find two different inputs with same Avalanche effect: small input change \rightarrow drastically different output.

Types:

1. MD5: Message Digest 5, output 128 bit hash, vulnerable to collision.
2. SHA-1: Secured Hash Algorithm-1, more secure than MD5 but broken.
3. SHA-2: output 256/512 bit hashed, currently secure, widely used.
4. SHA-3: Designed to be resistant to SHA-2 vulnerabilities, highly secure.

Digital Certificates:

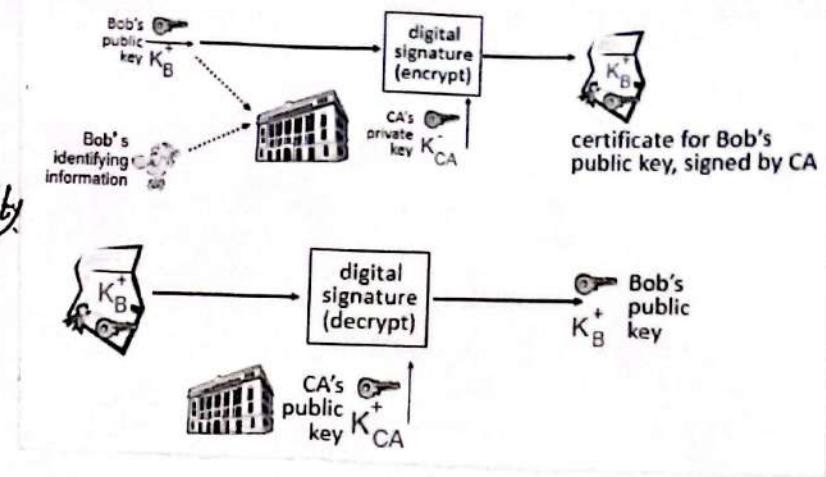
A certificate like SSL/TLS is a digital document issued by a Certificate Authority - CA to verify a website's ^(person's) identity and enable secure communication using encryption.

Why digital certificates?

- Ensures trust between users and websites.
- Enables HTTPS secure browsing.
- Protects data confidentiality and prevents man in the middle (MITM) attacks.

How to get certificates:

- Generate a certificate signing request (CSR) on your server.
- Submit to a certification authority (e.g. Let's encrypt, digicert)
- Complete Verification (domain).
- CA issues certificate.
- Install it on your server.

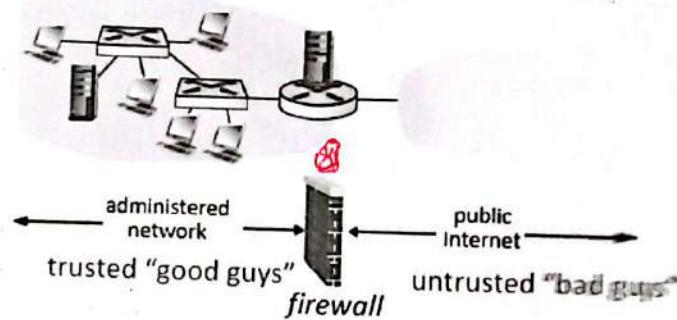


firewalls:

A firewall is a network security device or software that monitors and controls incoming and outgoing traffic based on security rules.

Used at:

- Network boundaries between internal network and internet, i.e. routers, servers, personal devices and corporate system.



Advantages:

- Blocks unauthorized access
- Filters malicious traffic.
- Enhances Network Monitoring and control

Disadvantages:

- Cannot detect internal threats.
- May slow down traffic.
- Needs regular updating and rule management.

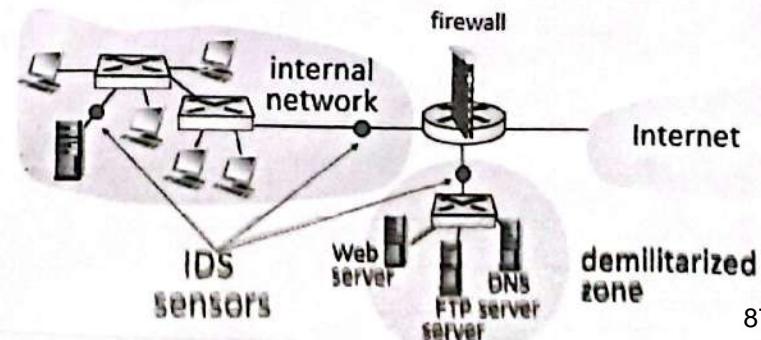
IDS - Intrusion Detection System

- Monitors network or system for malicious activity or policy violations.
- Alerts administrator but doesn't block traffic.

IPS - Intrusion prevention system

- Detects threats and actively blocks or prevents them in real time.
- Positioned in line with Network traffic.

A network system incorporated IDS & IPS enhances network security and helps in early threat detection and automatic response, reduces risk of data breaches.



Advantages of Public Key Cryptography over Symmetric Key Cryptography:

- No need to share secret keys securely
- Supports digital signatures for authentication
- Enables secure communication between strangers
- Better for open network environments
- Simplifies key management in large systems
- Enhances scalability for secure communications
- Enables non-repudiation (proof of origin)

Ensuring Message Authentication and Integrity (to the point):

- Use **Message Authentication Code (MAC)**: Sender generates MAC using a secret key and attaches it to the message; receiver verifies it using the same key.
- Use **Digital Signatures**: Sender signs the message hash with their private key; receiver verifies it using the sender's public key.
- Use **Hash Functions** (e.g., **SHA-256**): Ensures message integrity by checking if the hash matches after transmission.
- Combine **Encryption + MAC/Digital Signature** for confidentiality and authentication.

 Ensures message is from the legitimate sender and not altered.

What is Message Integrity?

Message integrity ensures that the contents of a message have **not been altered** during transmission, whether accidentally or maliciously. It verifies that the received data is exactly what the sender sent.

-  Achieved using cryptographic hash functions (e.g., SHA-256), **Message Authentication Codes (MACs)**, or **digital signatures**.
-  If even a single bit is changed, the integrity check fails.
-  Real-life example: Bank transaction details must remain unchanged from sender to bank server.

Q: Define Digital Signatures and What is the Purpose of a Certification Authority?

Digital Signature:

A digital signature is a cryptographic technique used to verify the authenticity and integrity of a message or document.

It uses the sender's private key to sign and the public key to verify.

Purpose of Certification Authority (CA):

- A Certification Authority issues **digital certificates** that bind a **public key** to a verified identity.
- Ensures **trust** in public key communication.
- Prevents **impersonation** and **man-in-the-middle attacks**.
- Enables secure services like **HTTPS**, **email encryption**, and **digital signatures**.



VLAN (Virtual Local Area Network)

A VLAN is a logical group of devices on the same network, segmented to improve security, reduce congestion, and simplify management.

◆ To-the-point Explanation:

- Separates broadcast domains without needing multiple switches.
- Devices in the same VLAN can communicate as if on the same physical network.
- Reduces network traffic and increases performance.

◆ Types of VLANs:

- **Default VLAN:** All ports are in VLAN 1 by default.
- **Data VLAN:** Carries user-generated data traffic.
- **Voice VLAN:** Dedicated for voice traffic (VoIP).
- **Management VLAN:** Used for switch management.
- **Native VLAN:** Carries untagged traffic on trunk ports.

Blow Your Mind:

Table: Impact on Cybersecurity

Technology	Effect on Cybersecurity	Challenges	Countermeasures/Benefits
Quantum Computing	Breaks RSA/ECC; threatens encryption & signatures	Weakens public-key systems, secure key exchange	Post-quantum cryptography, QKD, NIST PQC standards
Artificial Intelligence (AI)/ML	Enhances threat detection & response automation	Adversarial attacks, data poisoning, privacy risks	AI-driven SOCs, explainable AI, robust models
Starlink Antenna/Networking	Global low-latency secure internet access via LEO satellites	Satellite spoofing, jamming, tracking attacks	Encrypted links, quantum key distribution potential, resilience
Machine Learning	Predictive security analytics, anomaly detection	Model theft, false positives, bias	Secure model training, validation, and continuous learning