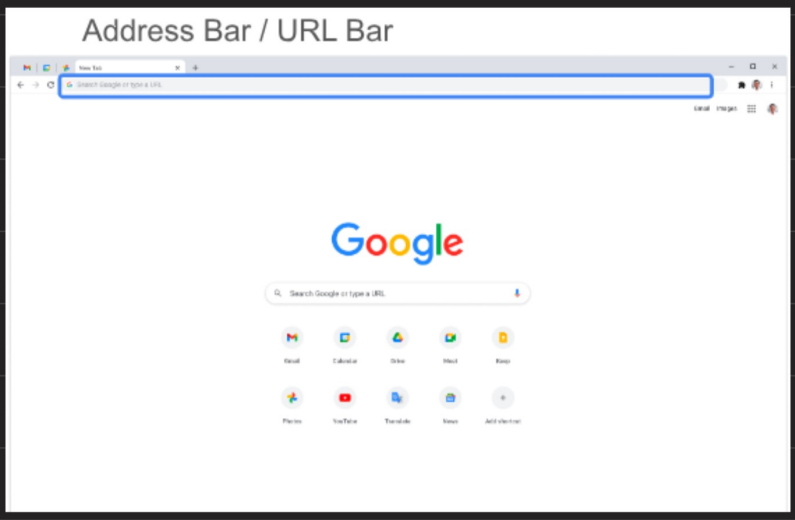
The Big picture - Part 1

What happens when you type "www.google.com" on browser ??

Browser

Address Bar / URL Bar



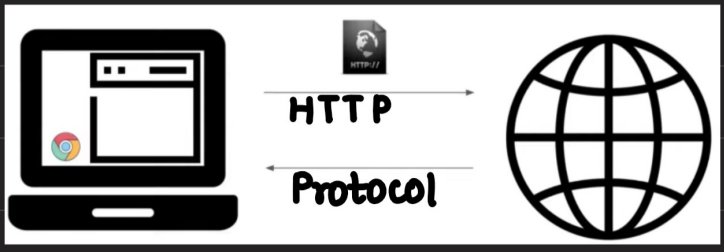
Client

A client is computer software, with underlying hardware, that access services hosted on a server. in the context of this article, the usage of the term "client" will refer to the web browser.

My computer is the client.

Protocol

A protocol is an established set of rules or procedures that dictate how data is transmitted between different devices on a network.

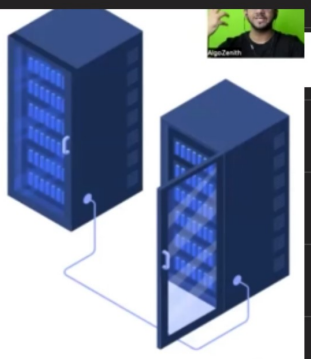


Data Packets

The actual request sent on the internet. Might be broken down into chucks.

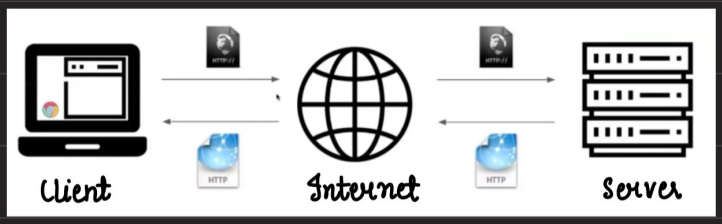
Server

A server is a computer software, with underlying hardware, that's dedicated to providing a service to other computers. In the context of this article, the usage of the term "server(s)" will refer to the computer system(s) hosting a website.

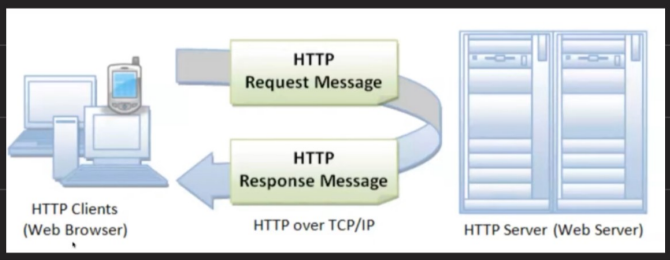


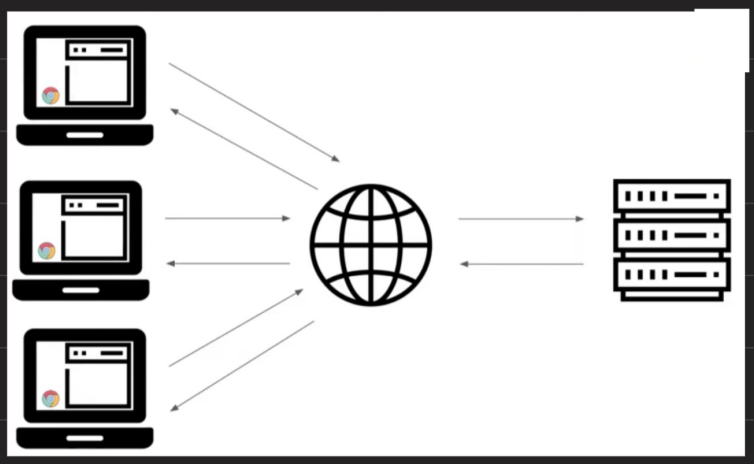
Web Server

A webserver is a type of server that uses HTTP, or it's variant, and other protocols to respond to web client requests made over the internet. It is its responsibility to display website content through storing, processing, and delivering web pages to the web browser.



Client - Server model





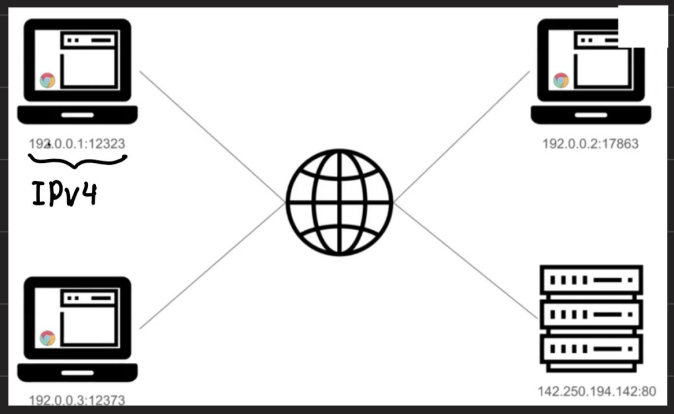
We need address

↓ ↗ IP address

Which computer ?

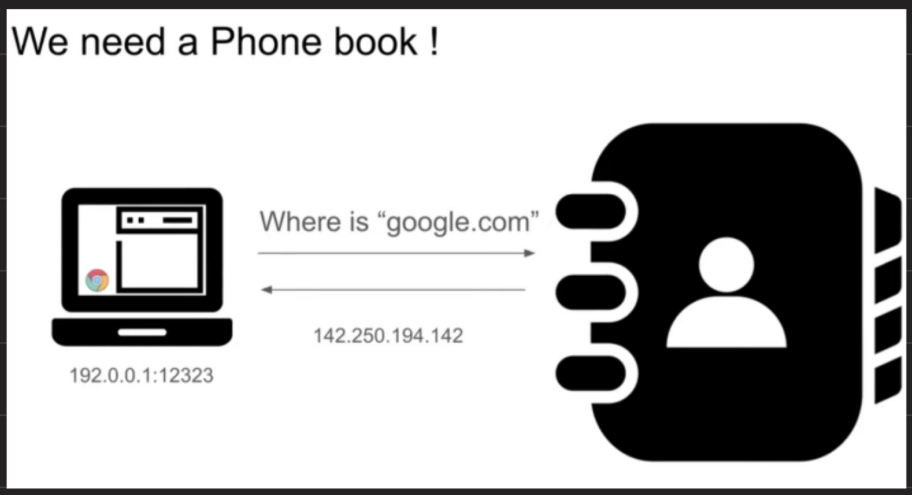
↓ ↗ Port

Which application ?



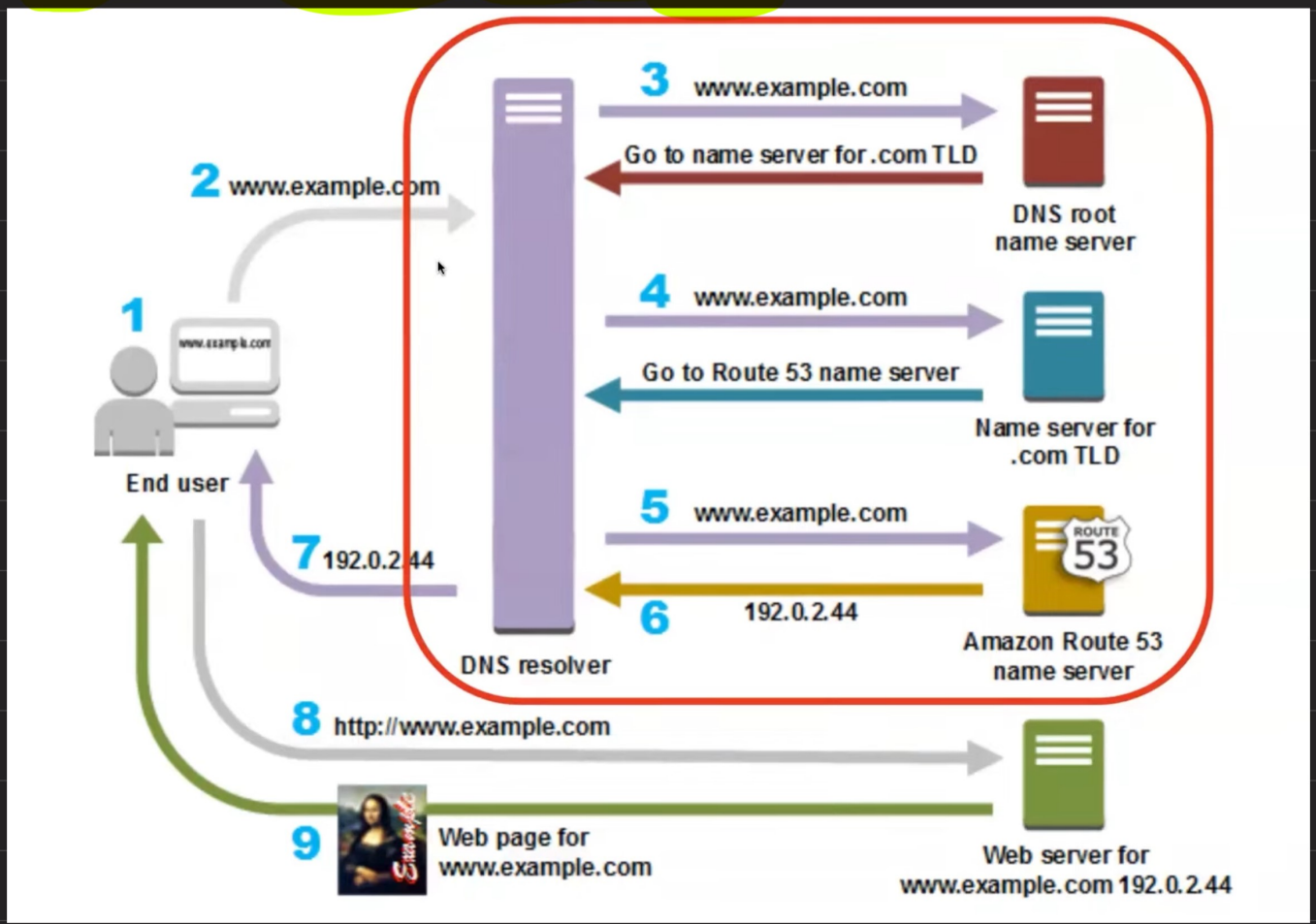
HTTP by default use port no. 80, so web servers open at 80

How do i know its - 142.250.194.142 - in the first place??



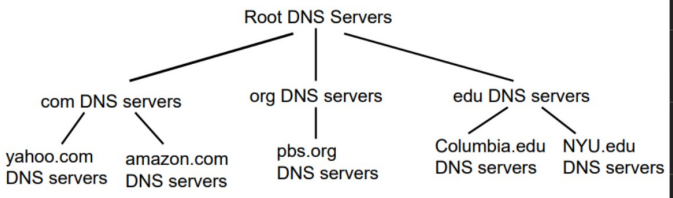
this phone book s/m is called DNS. (Domain name system)

Domain Name System



Caching is done to save IP address of recently searched websites.

Distributed, Hierarchical Database



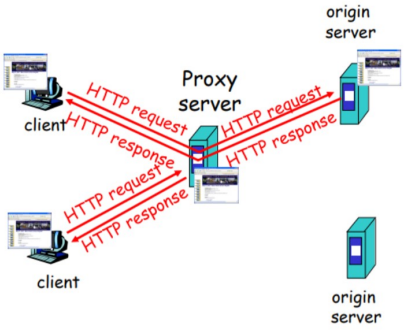
Client wants IP for www.amazon.com; 1st approx:

- client queries a root server to find com DNS server
- client queries com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

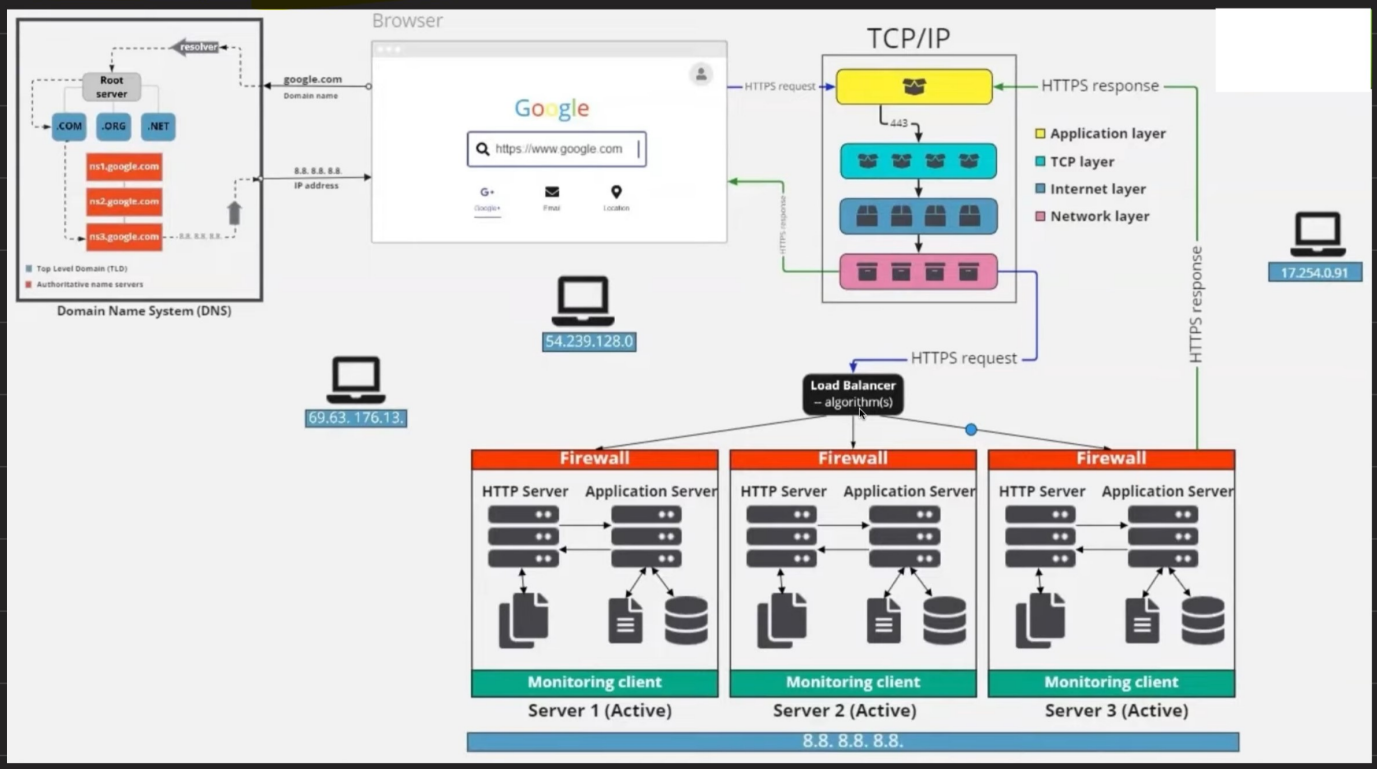
Web caches (proxy server)

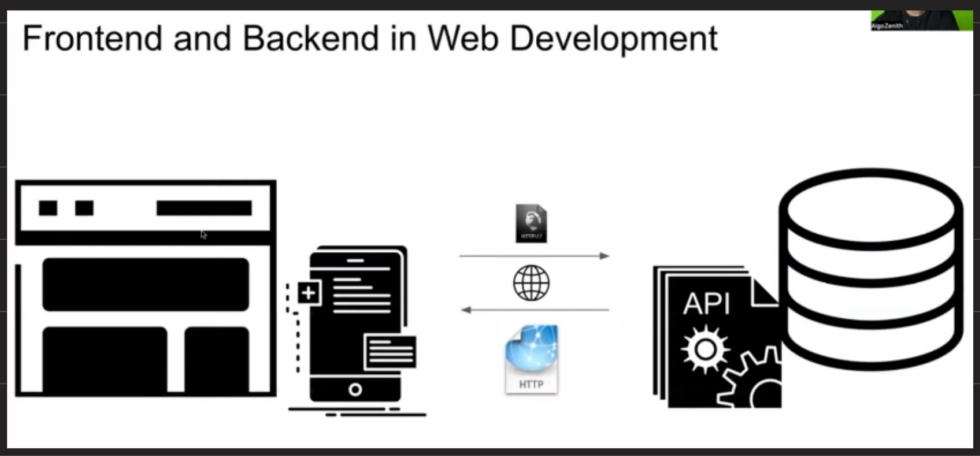
Goal: satisfy client request without involving origin server

- user sets browser: Web accesses via cache
- browser sends all HTTP requests to cache
  - object in cache: cache returns object
  - else cache requests object from origin server, then returns object to client

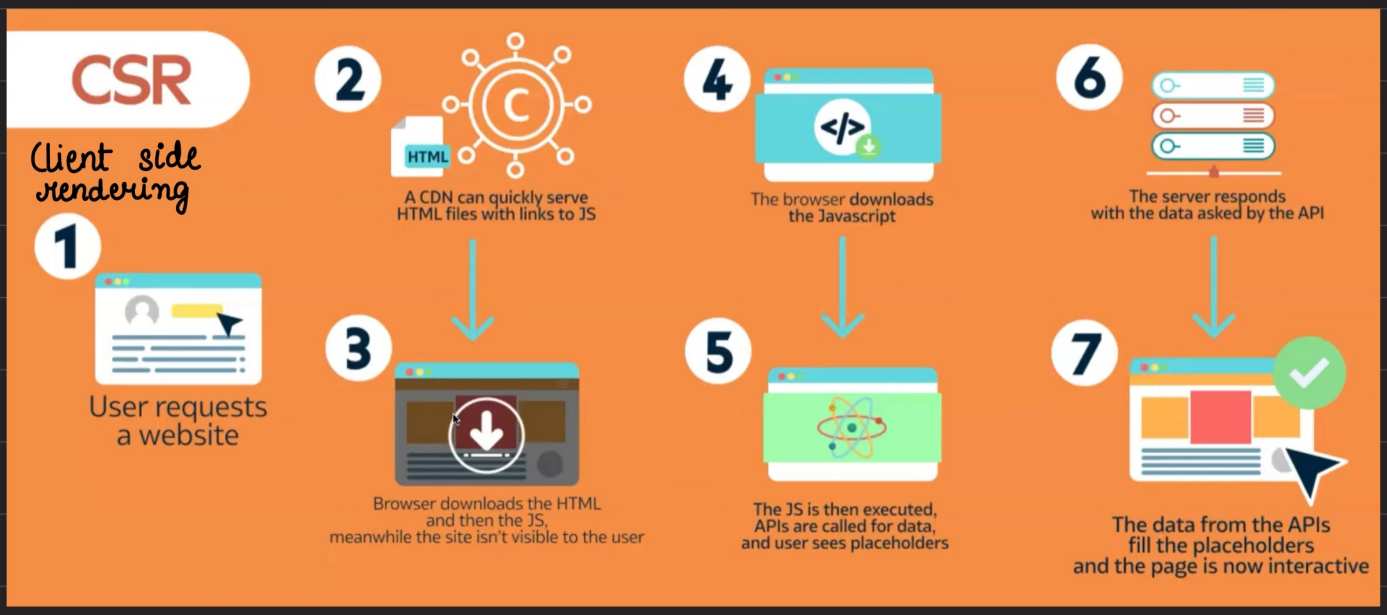


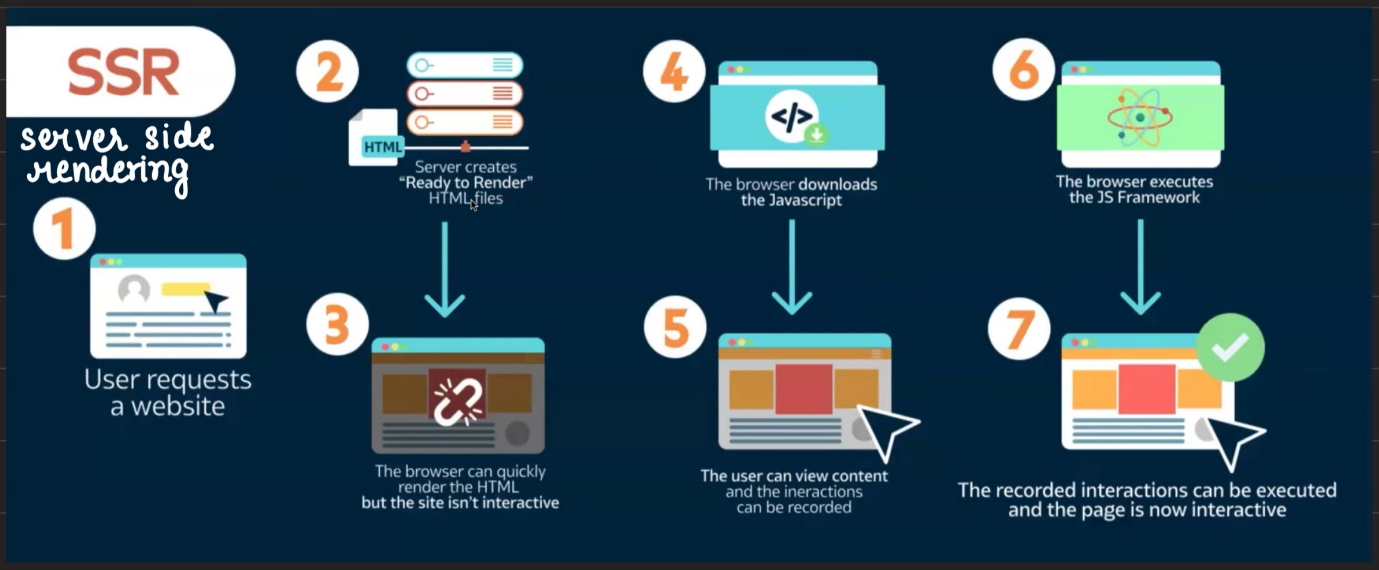
The Big picture



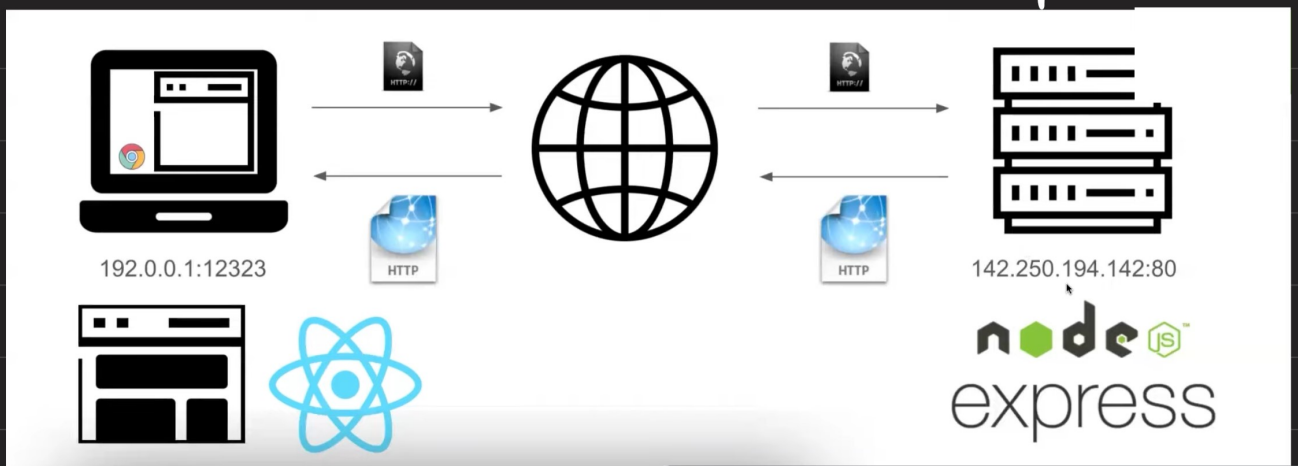


Frontend Backend





ReactJS provides CSR and NextJS provides SSR.

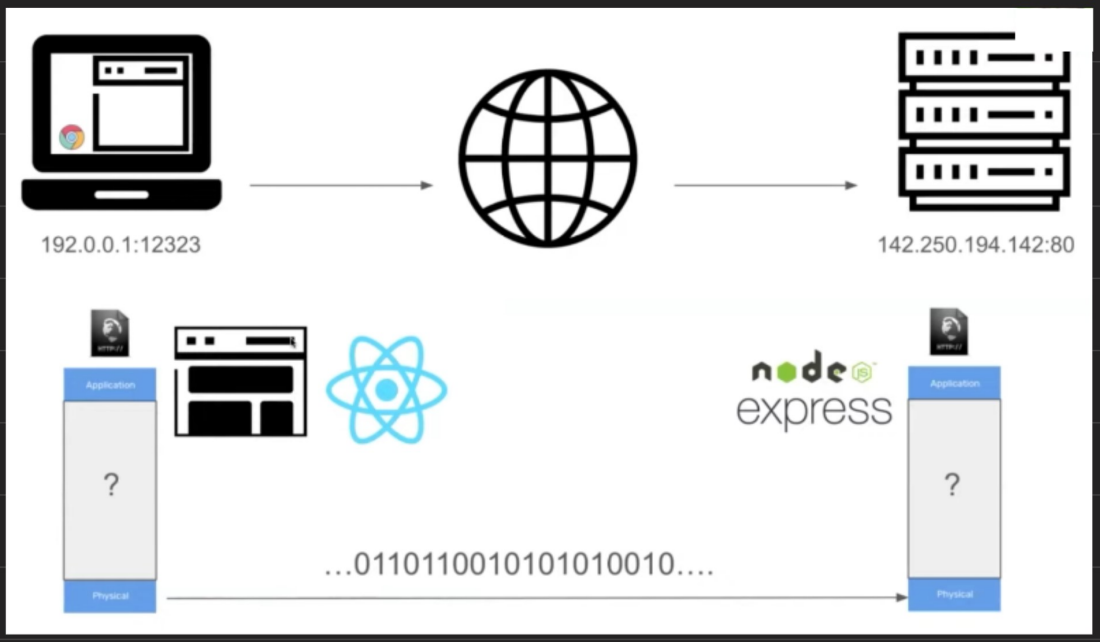


fetch data from API

```
import React, { useState, useEffect } from 'react';

function App() {
    const [data, setData] = useState(null);

    useEffect(() => {
        fetch('/api/data')
            .then(response => response.json())
            .then(data => setData(data))
            .catch(error => console.error('Error fetching data:',
```

return message "Hello from....."

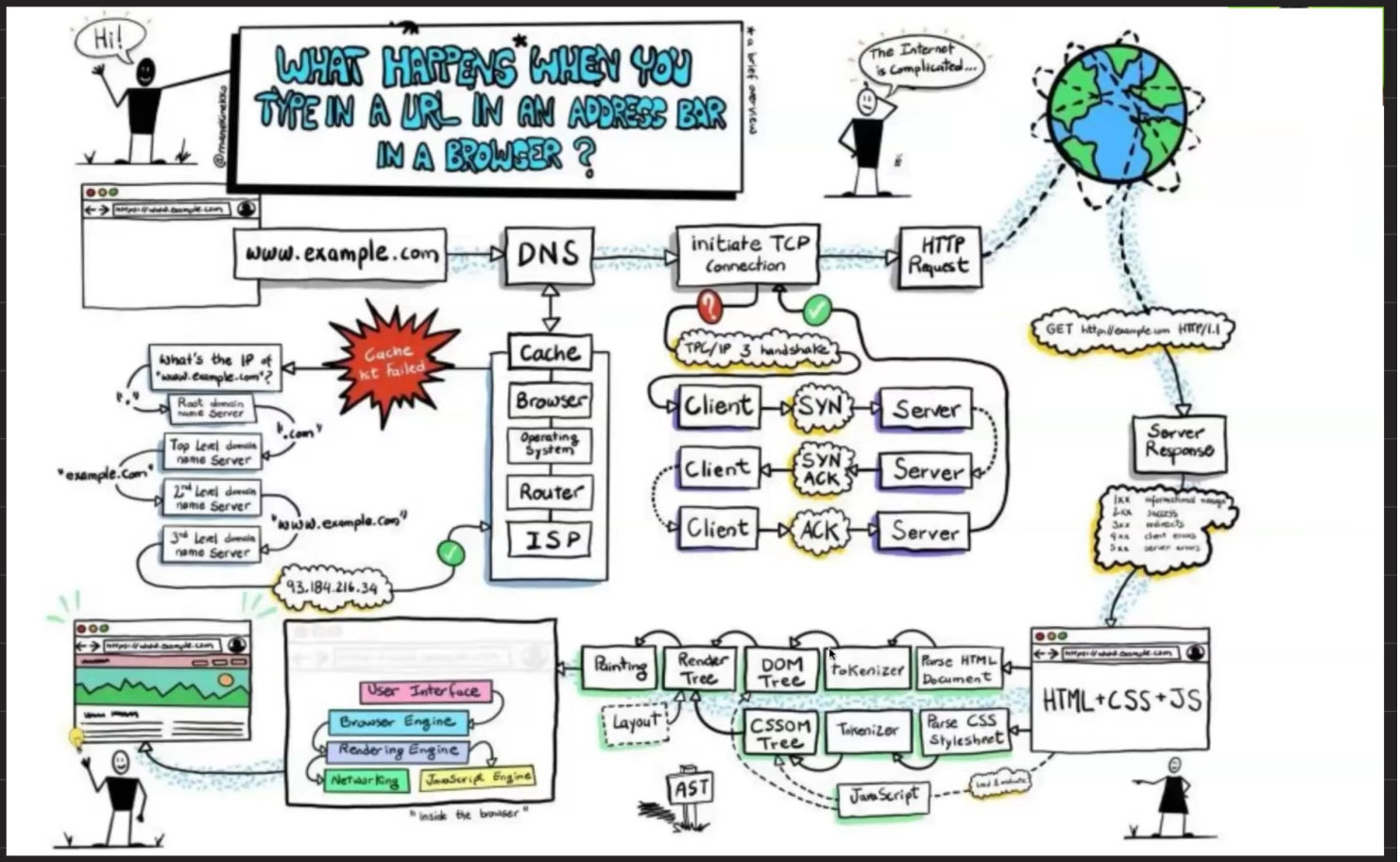
```
const express = require('express');
const app = express();
const port = 3001;

app.get('/api/data', (req, res) => {
    res.json({ message: 'Hello from Express!' });
});

app.listen(port, () => {
    console.log(`Server running on http://localhost:${port}`);
});
```

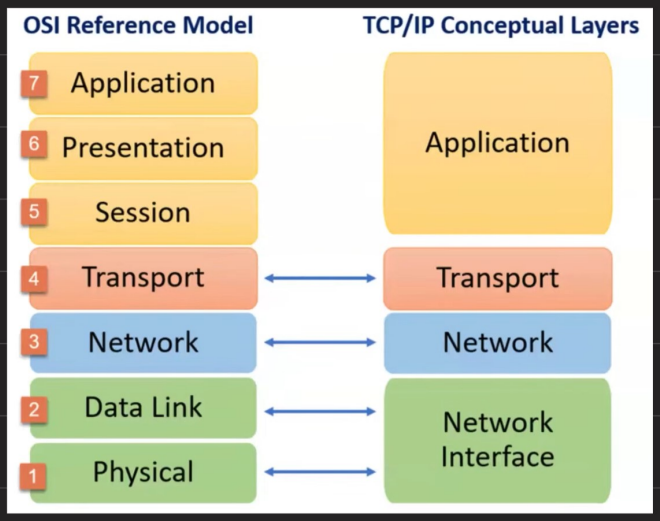


Application

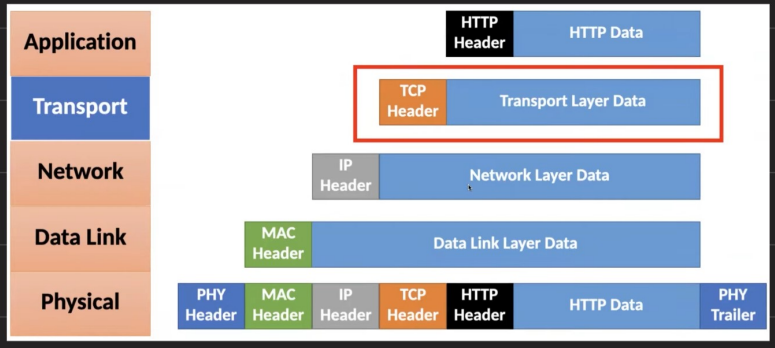
?

Physical





Data passing through different layer.



Functions of Different Layers

Application Layer

Network Services to End-Users: Provides network services to the applications of the end user, like email, file transfer, and web browsing.

Interface to Application Software: Acts as the interface for the network services to application software, abstracting the underlying networking details.

Presentation Layer

Data Translation and Encoding: Translates data from the application layer into an intermediary format and vice versa, also manages data encryption and decryption.

Data Compression: Reduces the size of data to be transmitted, optimizing the use of network resources.

Session Layer

Session Management: Establishes, manages, and terminates sessions between applications.

Synchronization: Adds checkpoints to the data stream; if a session is interrupted, it can be restarted from the last checkpoint.

Transport Layer

End-to-End Communication: Manages the end-to-end control and error-checking to ensure complete data transfer.

Segmentation, Reassembly, and Flow Control: Divides messages into segments and reassembles them at the destination, also controls the flow rate of data to ensure fast and reliable delivery.

Network Layer

Routing of Packets: Determines the best physical path for data to travel across a network from the source to the destination.

Logical Addressing: Introduces logical addresses (like IP addresses) for devices to ensure that each one has a unique identification.

Datalink Layer

Frame Traffic Control: It frames data for transmission, handles error detection (but not correction), and provides a way to regulate the flow of packets to and from the physical layer.

Physical Addressing: Adds physical addresses to the frames to identify devices on a local network.

Physical Layer

Transmission and Reception of Raw Bit Streams: Transmits raw bit streams over a physical medium like a cable. It involves the hardware transmission technologies.

Physical Connection Establishment and Termination: Establishes, maintains, and deactivates the physical connection.

```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

Request Line

Request Headers

Request Message Header

A blank line separates header & body

Request Message Body

- GET Method.
- HEAD Method. ...
- POST Method. ...
- PUT Method. ...
- DELETE Method. ...
- CONNECT Method. ...
- OPTIONS Method. ...
- TRACE Method.

```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

Status Line

Response Headers

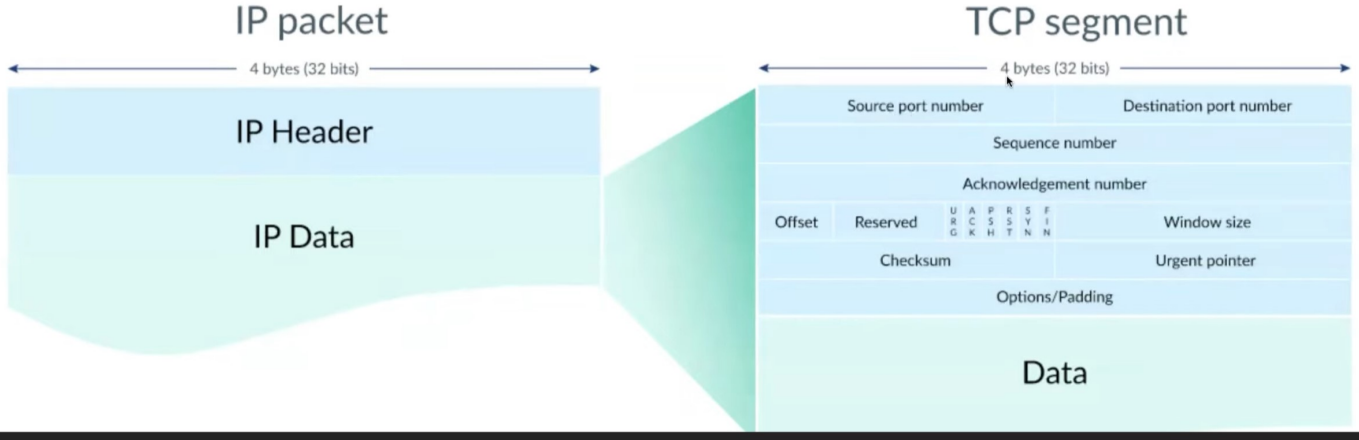
Response Message Header

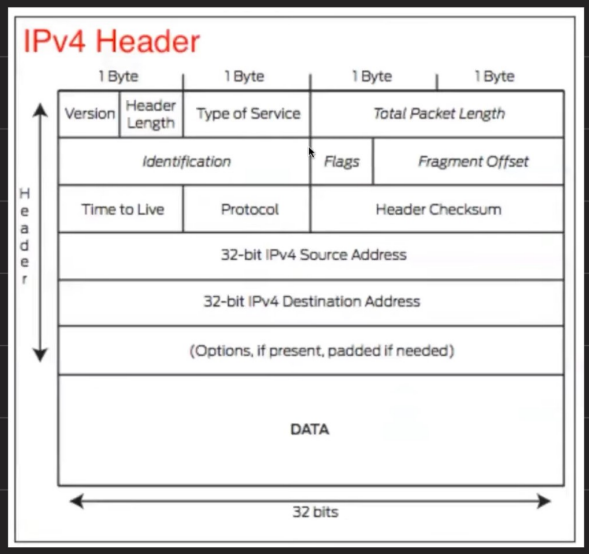
A blank line separates header & body

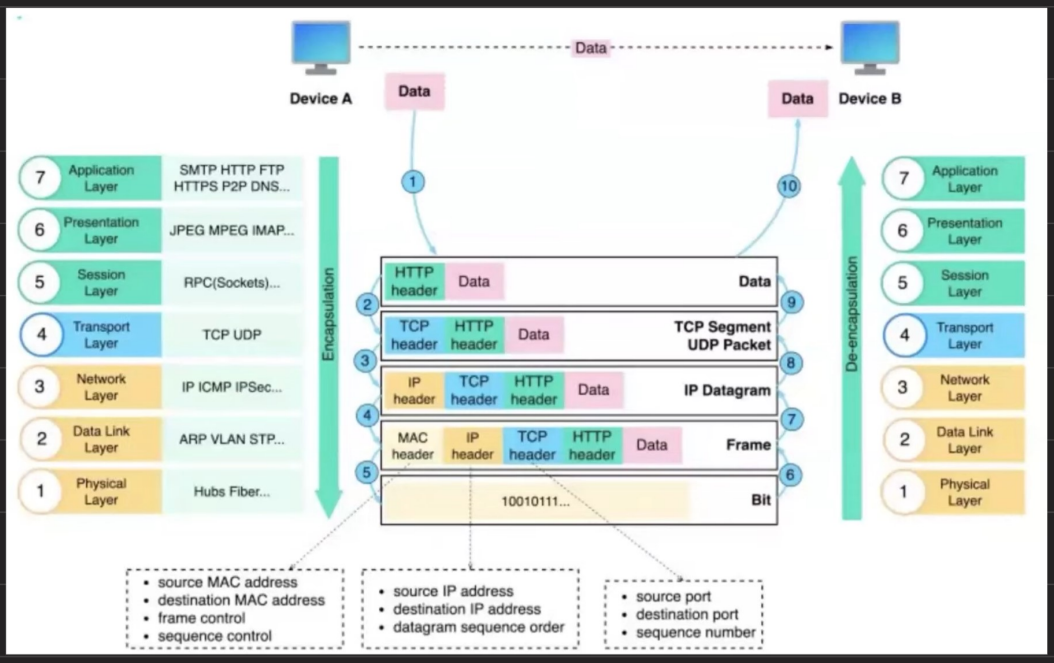
Response Message Body

1. Informational responses (100 – 199)
2. Successful responses (200 – 299)
3. Redirection messages (300 – 399)
4. Client error responses (400 – 499)
5. Server error responses (500 – 599)

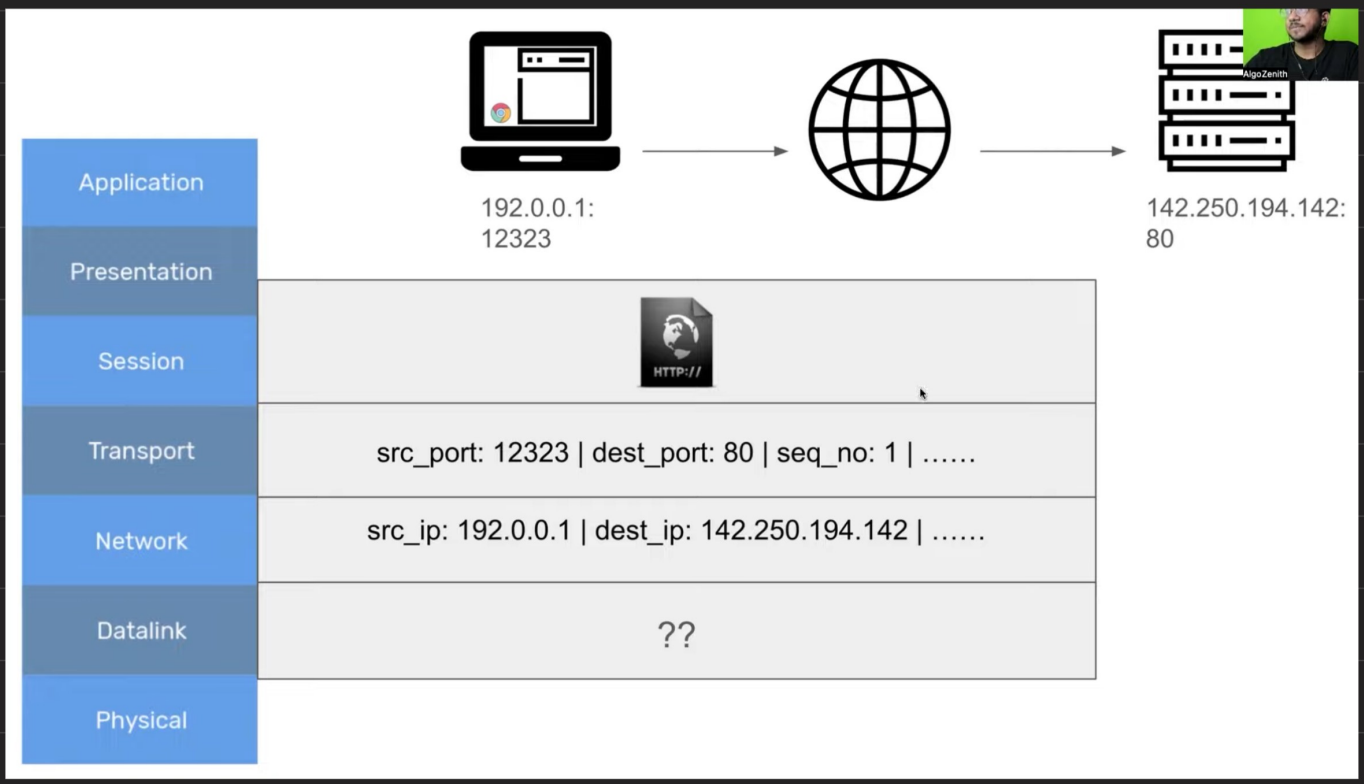
TCP - Transmission Control Protocol in Transport layer







Encapsulation & De-encapsulati.



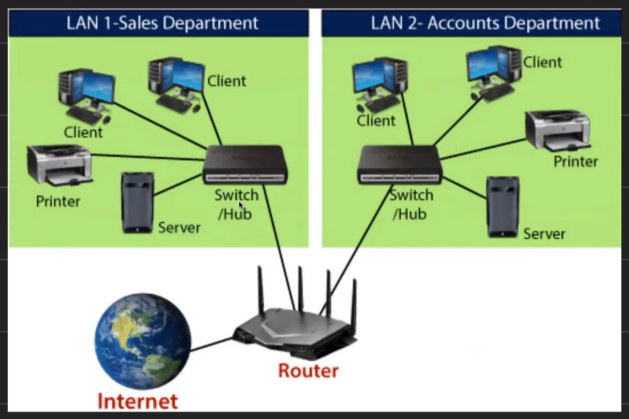
Sequence no. : Data is broken down into small units, each having a uni. sequence no.

Switches

A network switch is networking hardware that connects devices on a computer network by using packet switching to receive and forward data to the destination device. A network switch is a multiport network bridge that uses MAC addresses to forward data at the data link layer of the OSI model

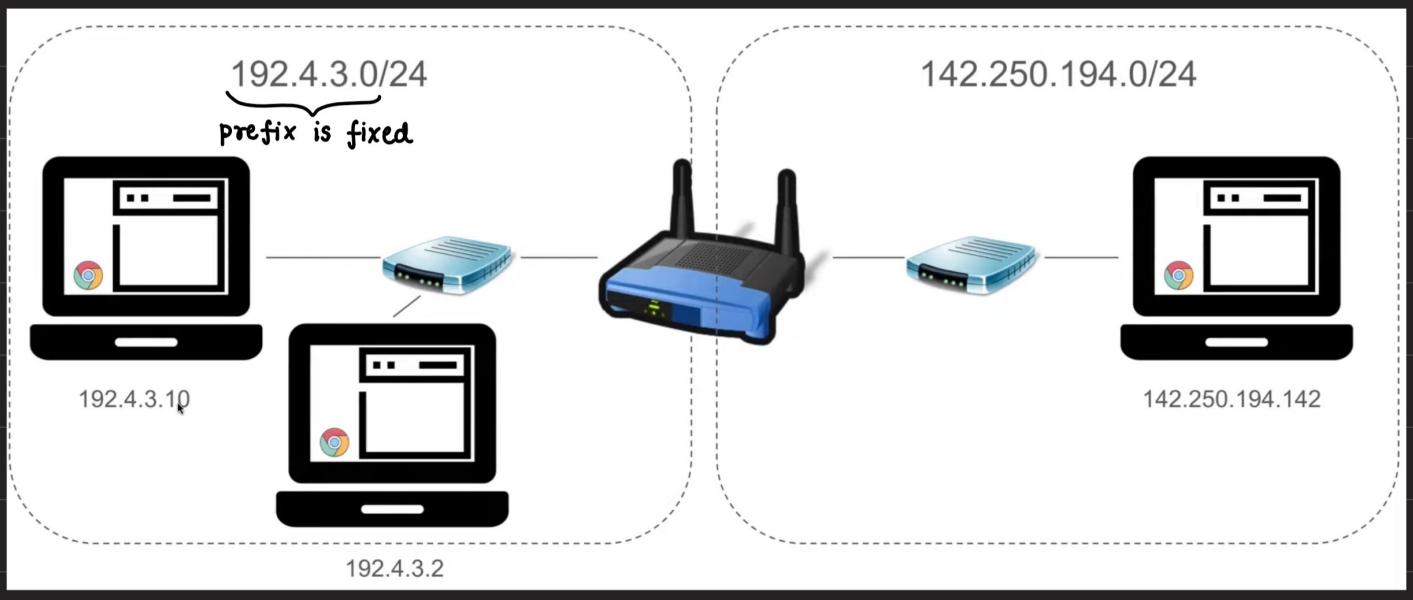
Router

A router is a device that connects two or more packet-switched networks or subnetworks. It serves two primary functions: managing traffic between these networks by forwarding data packets to their intended IP addresses, and allowing multiple devices to use the same Internet connection.



Everything **host** has an IP - 32 Bit number

```
10001000000010110000100010110010 0

1000 1000 . 0001 0110 . 0001 0001 . 0110 0010

 136   .   22   .   17   .   98
[0-255]    [0-255]    [0-255]    [0-255]
```

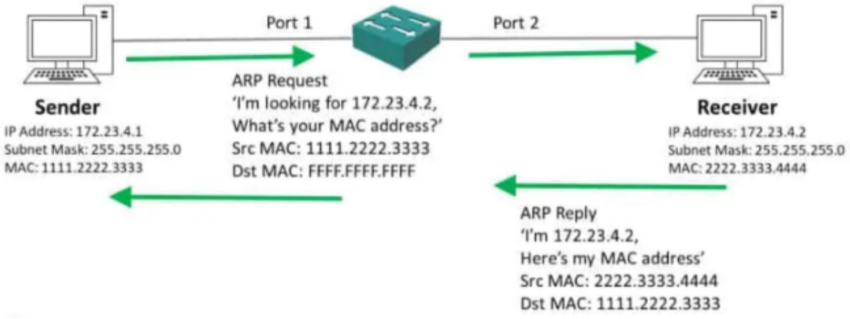


MAC Address (local address)

A MAC (Media Access Control) address, sometimes referred to as a hardware or physical address, is a unique, 12-character alphanumeric attribute that is used to identify individual electronic devices on a network. An example of a MAC address is: 00-B0-D0-63-C2-26.

Address Resolution Protocol (APR)

Address Resolution Protocol (ARP) is a protocol or procedure that connects an ever-changing Internet Protocol (IP) address to a fixed physical machine address, also known as a media access control (MAC) address, in a local-area network (LAN).

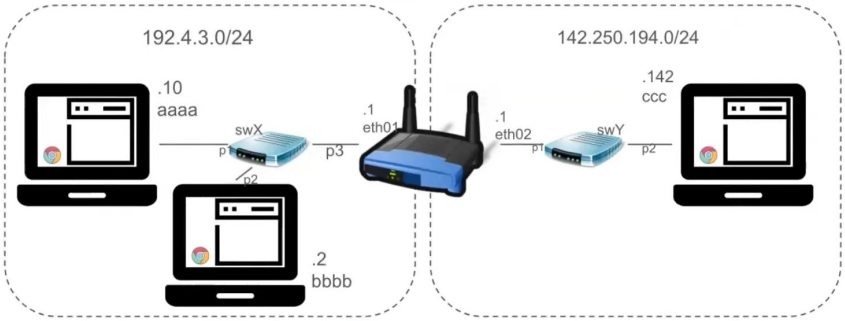


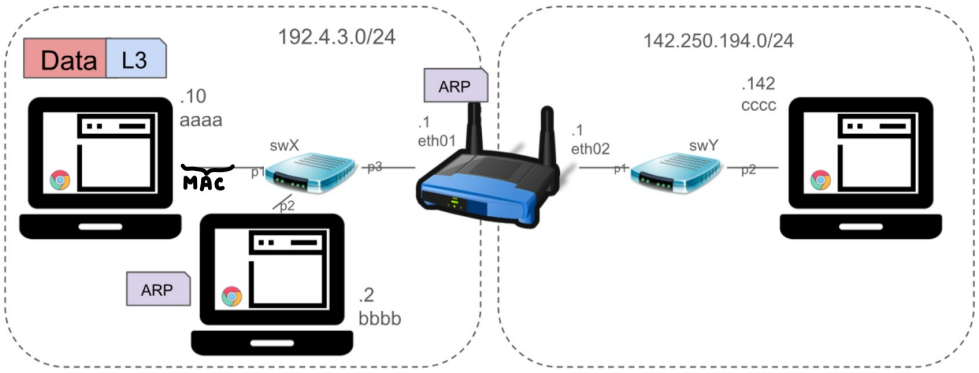
3 Things the Devices maintain

ARP Table - Mapping IP addresses to MAC Address

MAC Address Table - mapping of Switchport to MAC addresses (Switches)

Routing Table - Mapping of IP Networks to Interfaces





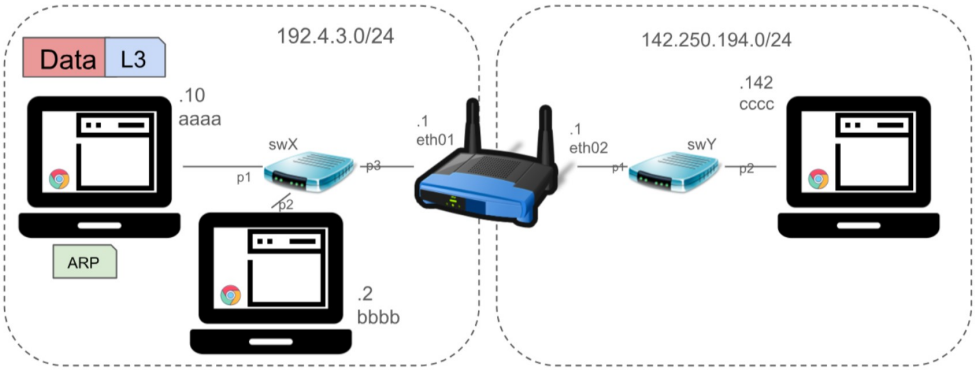
Sender ARP Table	Router ARP Table	Receiver ARP Table
	192.4.3.10 ..:aa:aa	

swX MAC Addr. Table	Router Routing Table	swY MAC Addr. Table
1 aa:aa:aa:aa:aa:aa	eth01 192.4.3.0/24 DC eth02 142.250.194.0/24 DC	

STEPs

1. When the networks starts, Routing table is populated from Directly connected interfaces
2. Sender adds L3 header with the target IP Address
3. Since not in the same IP mask… find Default Gateway's MAC address. But not in ARP !
4. Do an ARP request for Default gateway : 192.4.3.1
5. Switch X receives ARP request, saves mapping and forwards request to all (flood)
6. 192.4.3.2 drops the packet, router saves on ARP table

L3 header is header added till Network layer.



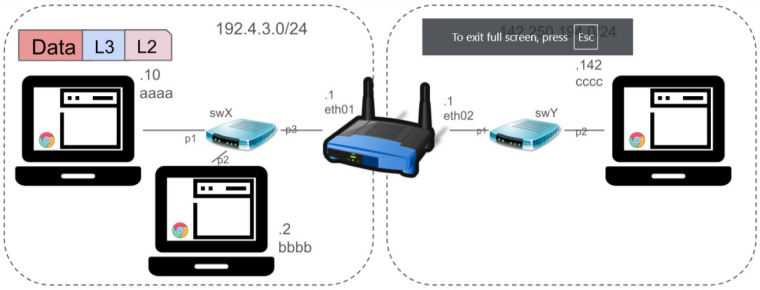
Sender ARP Table	Router ARP Table	Receiver ARP Table
192.4.3.1 ...:eth01	192.4.3.10 ..:aa:aa	

swX MAC Addr. Table	Router Routing Table	swY MAC Addr. Table
1 aa:aa:aa:aa:aa:aa 3 eth01:eth01:eth01	eth01 192.4.3.0/24 DC eth02 142.250.194.0/24 DC	

STEPs

6. 192.4.3.2 drops the packet, router saves on ARP table
7. The router responds to the ARP request back to the same switch for ..:aa:aa
8. The switch now saves the mac for router and sends to :aa saved on port 1.
9. The sender now saves the mac address of Router and knows where to send the data!

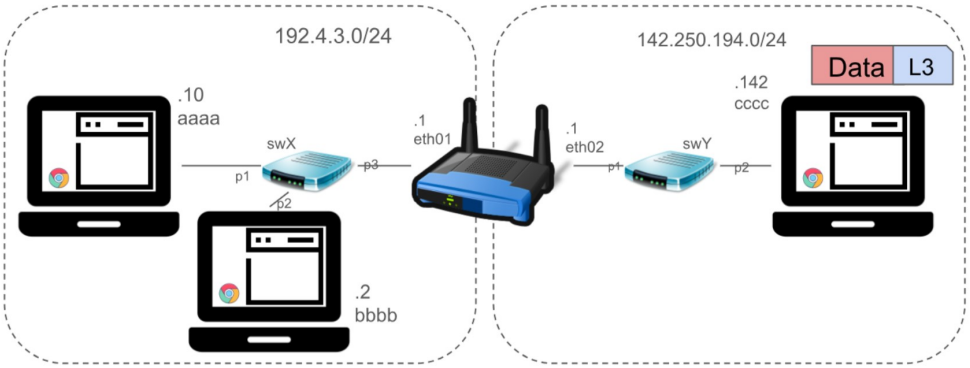
END OF PHASE 1
LETS RECAP !!



STEPs

END OF PHASE 1
LETS RECAP !!

10. Add the L2 header with mac address of router and send to the switch. The switch forwards.



Sender ARP Table

192.4.3.1	...:eth01

Router ARP Table

192.4.3.10	..:aa:aa
142.250.194.142	..:cc:cc

Receiver ARP Table

142.250.194.1	eth02

swX MAC Addr. Table

1	aa:aa:aa:aa:aa:aa
3	eth01:eth01:eth01

Router Routing Table

eth01	192.4.3.0/24	DC
eth02	142.250.194.0/24	DC

swY MAC Addr. Table

1	eth02.eth02.eth02
2	cc:cc:cc:cc:cc:cc

STEPs

END OF PHASE 1
LETS RECAP !!

10. Add the L2 header with mac address of router and send to the switch. The switch forwards.

11. Removes the L2 Header, Sees the L3 that it wants to go to 142.250.194.142.

12. Since it doesn't know the mac for 142.250.194.142, activate ARP!!

13. Add the L2 header with mac address of .142 and send to the switch. The switch forwards.

13. 142.250.194.142 receives, removes L2 and L3 headers … and processes data.

NOTE:

Ethernet (802.3) Frame Format

7 bytes	1 byte	6 bytes	6 bytes	2 bytes	42 to 1500 bytes	4 bytes	12 bytes
Preamble	Start of Frame Delimiter	**Destination MAC Address**	**Source MAC Address**	Type	Data (payload)	CRC	Inter-frame gap

For TCP/IP communications, the payload for a frame is a packet

WiFi (802.11) Frame Format

2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes	6 bytes	0 to 2312 bytes	4 bytes
Frame Control	Duration	MAC Address 1 (Destination)	MAC Address 2 (Source)	MAC Address 3 (Router)	Seq Control	MAC Address 4 (AP)	Data (payload)	CRC

Interview Questions

What is firewall??

Firewall – A firewall is a network security device that monitors incoming and outgoing network traffic and permits or blocks data packets based on a set of security rules. Its purpose is to establish a barrier between your internal network and incoming traffic from external sources (such as the internet) in order to block malicious traffic like viruses and hackers.

How do Firewalls Work :

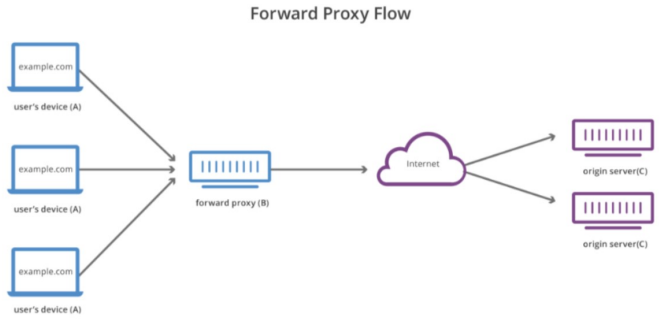
Firewalls carefully analyze incoming traffic based on pre-established rules and filter traffic coming from unsecured or suspicious sources to prevent attacks. Firewalls guard traffic at a computer's entry point called ports, which is where information is exchanged with external devices. For example, "Source address 172.18.1.1 is allowed to reach destination 172.18.2.1 over port 22."

Firewalls can either be software or hardware, though it's best to have both. A software firewall is a program installed on each computer and regulates traffic through port numbers and applications, while a physical firewall is a piece of equipment installed between your network and gateway.

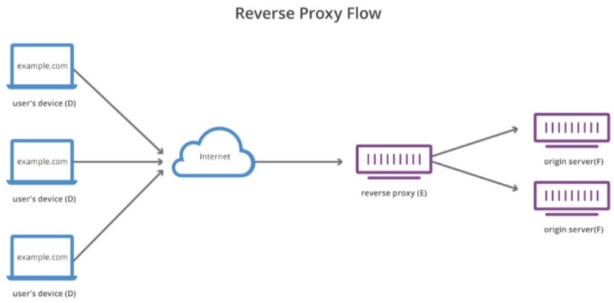
Packet-filtering firewalls, the most common type of firewall, examine packets and prohibit them from passing through if they don't match an established security rule set. This type of firewall checks the packet's source and destination IP addresses. If packets match those of an "allowed" rule on the firewall, then it is trusted to enter the network.

What's a proxy server?

A forward proxy, often called a proxy, proxy server, or web proxy, is a server that sits in front of a group of client machines. When those computers make requests to sites and services on the Internet, the proxy server intercepts those requests and then communicates with web servers on behalf of those clients, like a middleman.



- **To avoid state or institutional browsing restrictions**
- **To block access to certain content**
- **To protect their identity online**



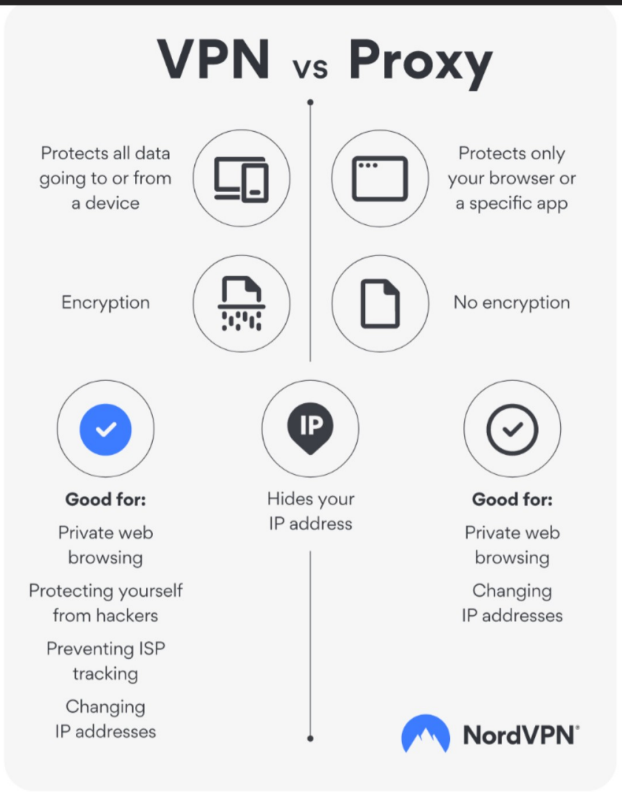
- **Load balancing** / **Global Server Load Balancing** **(GSLB)**
- **Protection from attacks** - such as a DDoS attack. Instead the attackers will only be able to target the reverse proxy, such as Cloudflare's CDN, which will have tighter security and more resources to fend off a cyber attack.
- **Caching** - A reverse proxy can also cache content geographically, resulting in faster performance.

What is bandwidth?
Ans: Transmission rate (bits/second)

What's a VPN?

Like a proxy, a VPN also reroutes your internet traffic through a remote server and hides your IP address so websites can't see your original IP or location. However, it works on the operating system level, meaning that it redirects all your traffic, whether it's coming from your browser or a background app.

A VPN client also encrypts your traffic between the internet and your device. That means the Internet Service Provider (ISP) monitoring your internet activity and collecting data about you can no longer see what you're doing online – just that you're connected to a VPN server. The encryption also protects you from government surveillance, website tracking, and any snoopers or hackers who might try to intercept your device.

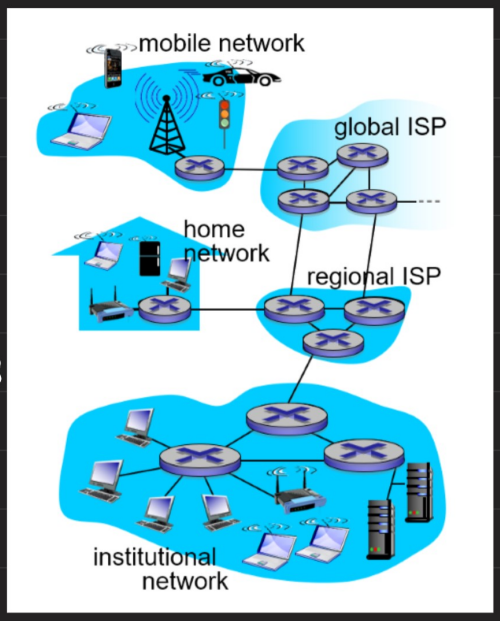


Gateway	Router
A device that acts as a gate and connects two networks having different sets of protocols.	A networking device that manages and forwards data packets to computer networks.
The main principle of a gateway is to convert one protocol to the other.	The main principle of the router is to route traffic from one network to another with the best route.
A gateway acts as a connection between two dissimilar networks.	A router transports data packets over similar networks.
A gateway does not support dynamic routing.	A router supports dynamic routing.
A gateway can operate until layer 5 of the OSI model.	A router can operate only on layer 3 and layer 4 of the OSI model.

3G vs 4G vs 5G

Feature	3G	4G	5G
Data Speed	Up to 2 Mbps	Up to 1 Gbps	Up to 10 Gbps and beyond
Latency	Around 100-150 ms	Around 30-50 ms	1 ms or less
Frequency Bands	1.8 - 2.5 GHz	2 - 8 GHz	24 GHz and above (millimeter waves)
Technology	WCDMA, HSPA	LTE, LTE-A	NR (New Radio)
Bandwidth	5-20 MHz	Up to 40 MHz	Up to 100 MHz and 1 GHz (mmWave)
Connection Density	Up to 100,000 devices per km^2	Up to 1,000,000 devices per km^2	Up to 1,000,000 devices per km^2
Key Use Cases	Voice calls, SMS, basic internet	HD video streaming, online gaming	IoT, autonomous vehicles, AR/VR
Peak Download Rate	21.6 Mbps	1 Gbps	20 Gbps

Q1. What is internet ?

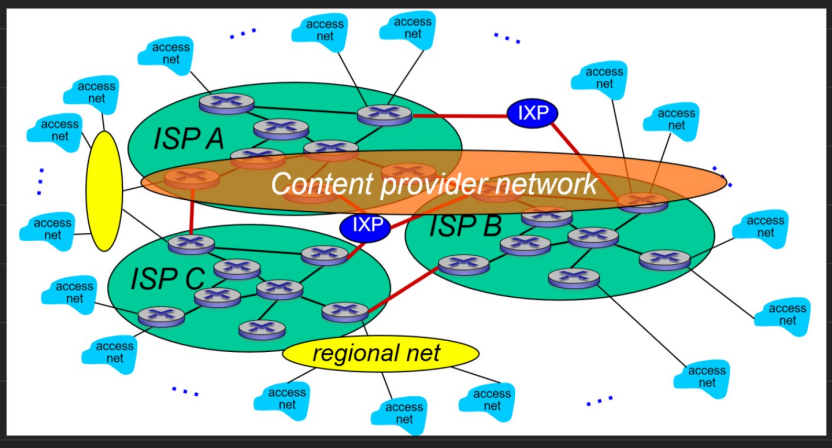


Ans: It is a network of networks.

- network edges : hosts (clients and servers)
- access networks : wireless/ wired communication links.
  eg: digital subscriber line, cable network, home network. Enterprise access network (Ethernet), wireless LANs, etc.
- network core : interconnected routers.

End devices are connected to edge routers through access net. This access net is provided to us through ISP (internet service provider).

Now, all ISPs are connected to each other through global ISPs. These global ISPs are also connected to each other through IXP (internet exchange points). Regional nets may arise to connect access net to ISP. Google, Microsoft, etc may run their own networks to bring services and content close to end users.



This whole network of networks is what we call as internet.

Some Basic terminologies

Networks are classified on the basis of area of distribution of the network.

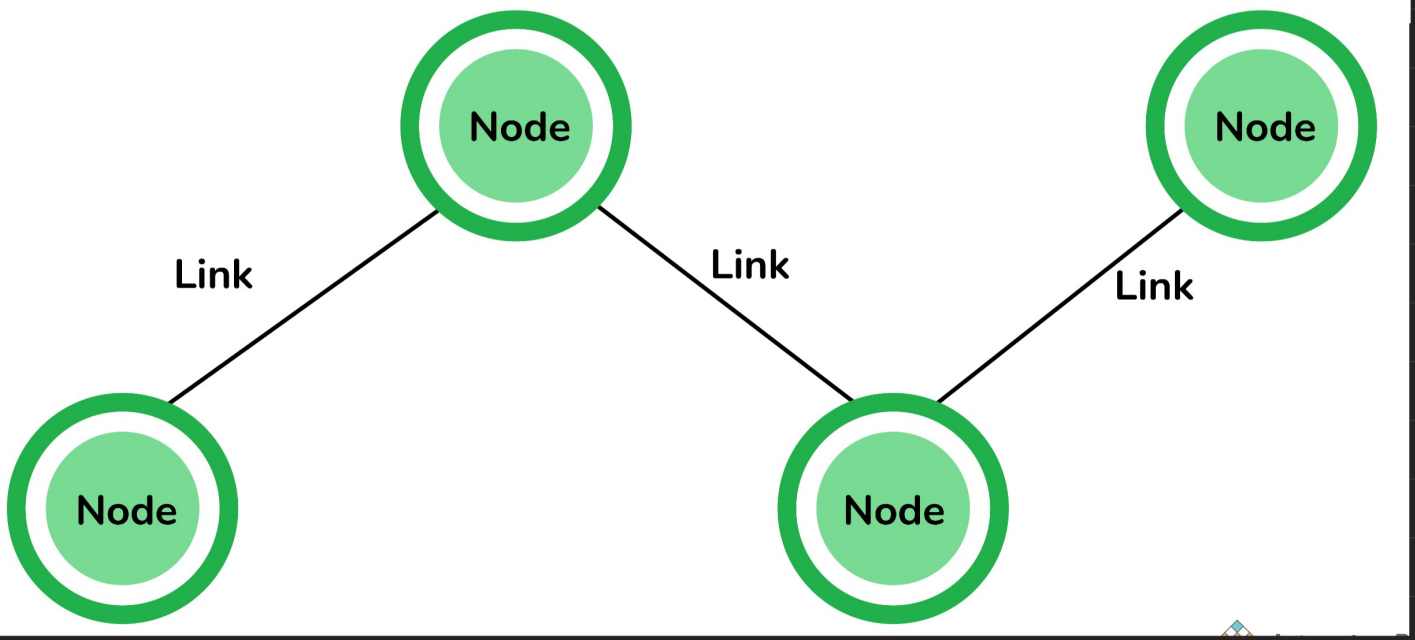
i) PAN (person area network)
eg: Bluetooth, wireless mouse, wireless headphone.

Distance	Region	
1m	Square meter	Personal area network
10m	Room	Local area network
100 m	Building	
1 km	Campus	
10 KM	City	Metropolitan area network
100 KM	Country	Wide area network
1000 KM	Continent	
10,000 km	Planet	The Internet (Global Area Network)

ii) LAN iii) MAN iv) WAN → The global area network.

Node: Any communicating device in a network is called a Node. Node is the point of intersection in a network. It can send/receive data and information within a network. Examples of the node can be computers, laptops, printers, servers, modems, etc.

Link: A link or edge refers to the connectivity between two nodes in the network. It includes the type of connectivity (wired or wireless) between the nodes and protocols used for one node to be able to communicate with the other.



Delays

Types of delay:

1. **Transmission Delay:** Transmission delay is referred to as the time taken to put the data packet onto the outgoing transmission link. Transmission delay depends on the length/size of the packet(L) and bandwidth of the network(B) and is calculated as :

Transmission Delay(T_t) = Data size/bandwidth = (L/B) second

2. **Propagation Delay:** The time it takes for the last bit of a data packet to pass across the medium and reach the other end is known as propagation delay. The propagation delay is determined by the distance (D) between the transmitter and receiver, as well as the wave signal propagation speed (S). It's calculated as follows:

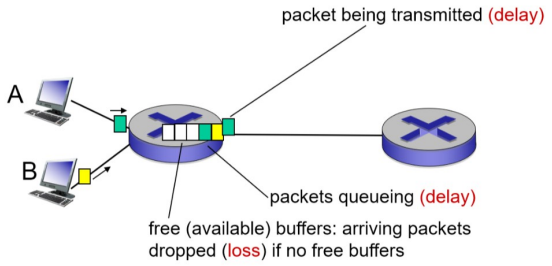
Propagation Delay(Tp)= Distance / Velocity=(D/S)second

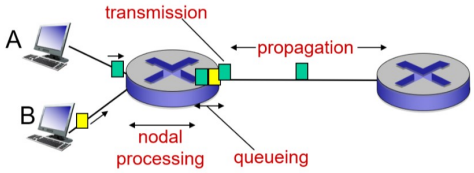
3. **Queuing Delay:** When a data packet arrives at its destination, it must queue before being processed. Queuing delay is the amount of time a data packet spends waiting in a queue before being processed. There is no set formula for calculating queuing delay; it is determined by the rate at which incoming packets arrive, the outgoing link's transmission capacity, and the nature of the network's traffic.

4. **Processing Delay:** The amount of time it takes processors to process the packet header is known as processing delay. The processing of packets helps in detecting errors and deciding where to route the packet. It doesn't have a formula because it is determined by the processor's speed.

packets *queue* in router buffers

- packet arrival rate to link (temporarily) exceeds output link capacity
- packets queue, wait for turn





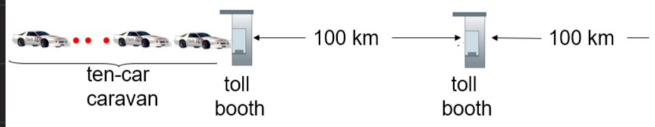
$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$

d_{trans}: transmission delay:
- L: packet length (bits)
- R: link *bandwidth (bps)*
- $d_{trans} = L/R$

d_{trans} and d_{prop} *very* different

d_{prop}: propagation delay:
- d: length of physical link
- s: propagation speed (~2×10^8 m/sec)
- $d_{prop} = d/s$

Caravan analogy



- cars "propagate" at 100 km/hr
- toll booth takes 12 sec to service car (bit transmission time)
- car ~ bit; caravan ~ packet
- Q: How long until caravan is lined up before 2nd toll booth?

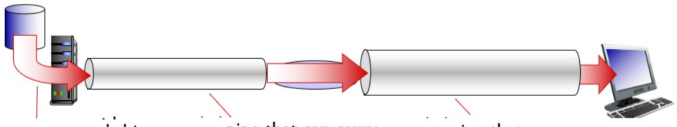
- time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- A: 62 minutes

transmission delay = 120 sec

propagation delay = 1 hr

Throughput

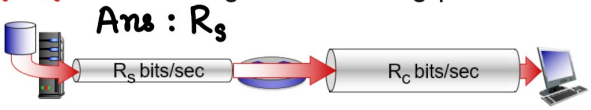
- *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time



Throughput (more)

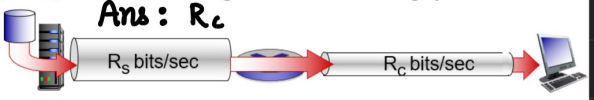
- $R_s < R_c$ What is average end-end throughput?

Ans : R_s



- $R_s > R_c$ What is average end-end throughput?

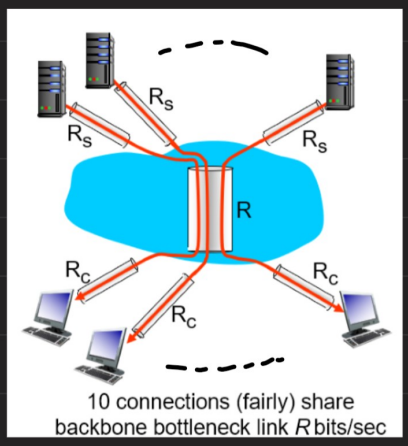
Ans : R_c



bottleneck link

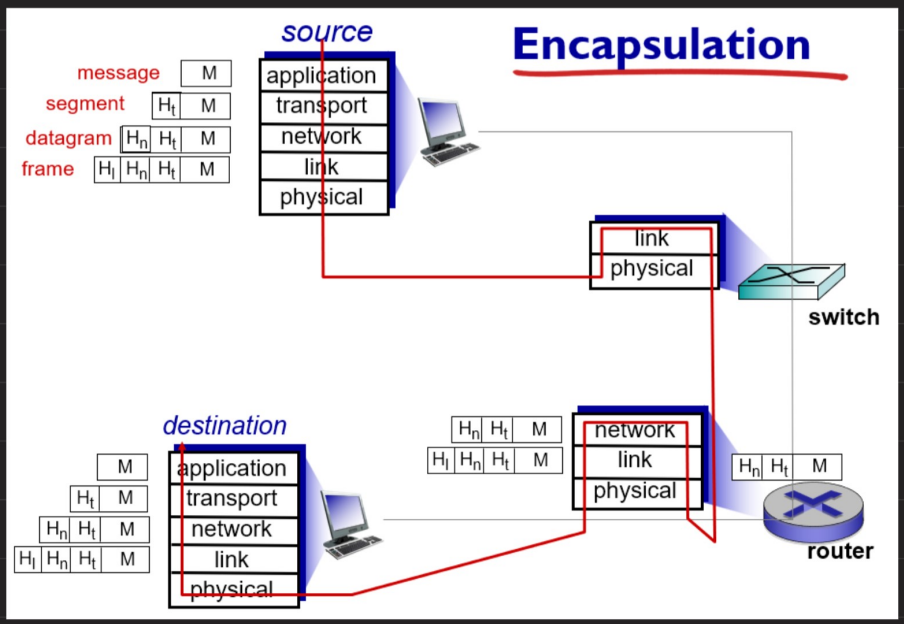
link on end-end path that constrains end-end throughput

Throughput is the minimum of bit transmission rate of all the connections b/w sender and receiver.



Here, throughput is min (R_s, R_c, $R/10$)

Encapsulation



Layer	Name
Application	message
Transport	segment
Network	datagram
Link	frame

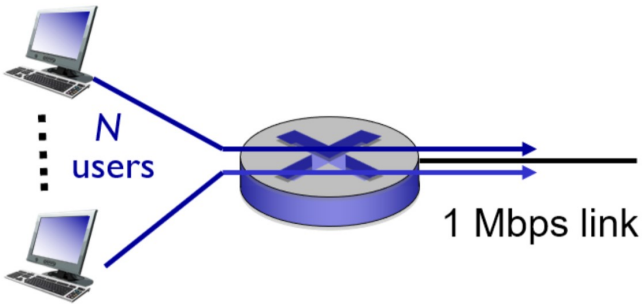
Packet Switching and Circuit Switching

Feature	Packet Switching	Circuit Switching
Data Transmission	Data is divided into packets, each taking its own path.	A dedicated communication path is established for the session.
Flexibility	Packets can take different routes based on network load.	The same path is used for the entire session, no flexibility.
Reliability	Robust, as packets can be re-routed if a route fails.	Stable connection with a reserved circuit for the session.
Efficiency	More efficient, allows dynamic sharing of network resources.	Less efficient, as the dedicated path is reserved even if idle.
Applications	Suited for data that can tolerate delays (e.g., web browsing, emails).	Ideal for real-time applications (e.g., voice calls, video conferencing).

packet switching allows more users to use network!

example:

- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time



- *circuit-switching:*
  - 10 users
- *packet switching:*
  - with 35 users, probability > 10 active at same time is less than .0004 *

Q: how did we get value 0.0004?

Q: what happens if > 35 users ?

Application Layer

Content

- Intro
- Protocols
- DNS
- Caching (proxy servers)

Application Layer

Network Services to End-Users: Provides network services to the applications of the end user, like email, file transfer, and web browsing.

Interface to Application Software: Acts as the interface for the network services to application software, abstracting the underlying networking details.

Protocols

- HTTP
- FTP
- SMTP
- TELNET
- DNS

Feature	HTTP	FTP	SMTP	Telnet
Full Form	Hyper Text Transfer Protocol	File Transfer Protocol	Simple Mail Transfer Protocol	Telecommunications Network
Purpose	Transmitting hypermedia documents (e.g., HTML)	Transferring files between devices	Transferring electronic mail	Managing files over the Internet
Client-Server	Yes	Yes	Yes	Yes
Port Number	80	20 (data), 21 (control)	25, 587	23
Stateless	Yes	No	No	No
Use Cases	Web browsing, web server communication	File sharing, remote file access	Email sending	Remote resource access
Other Purposes	Can be used for several other purposes	Reliable, efficient data transfer	Used by end users to send emails with ease	Accessing Telnet server resources

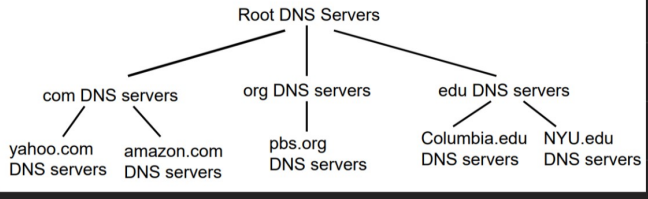
Different between HTTP and HTTPS:

Feature	HTTP	HTTPS
Full Form	Hyper Text Transfer Protocol	Hyper Text Transfer Protocol Secure
Purpose	Transmitting hypermedia documents (e.g., HTML)	Transmitting hypermedia documents securely
Port Number	80	443
Data Encryption	No	Yes (using SSL/TLS)
Certificate	No certificate required	Requires SSL/TLS certificate
Use Cases	General web browsing	Secure web browsing (e.g., online banking, shopping)
Performance	Slightly faster due to lack of encryption	Slightly slower due to encryption overhead
SEO Impact	Neutral	Preferred by search engines (can improve ranking)

Domain name System

DNS: DNS stands for Domain Name System. The DNS service translates the domain name (selected by user) into the corresponding IP address. For example- If you choose the domain name as www.abcd.com, then DNS must translate it as 192.36.20.8 (random IP address written just for understanding purposes). DNS protocol uses the port number 53.

Distributed, Hierarchical Database



Client wants IP for www.amazon.com; 1st approx:

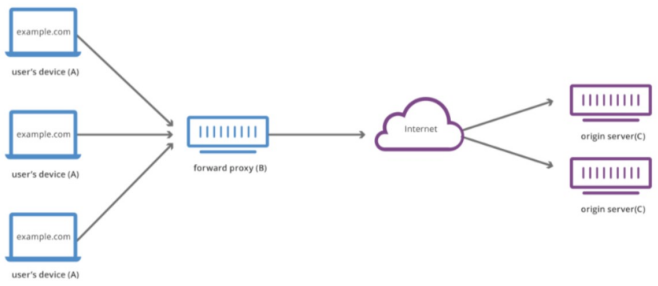
- client queries a root server to find com DNS server
- client queries com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

Caching (Proxy Servers)

What's a proxy server?

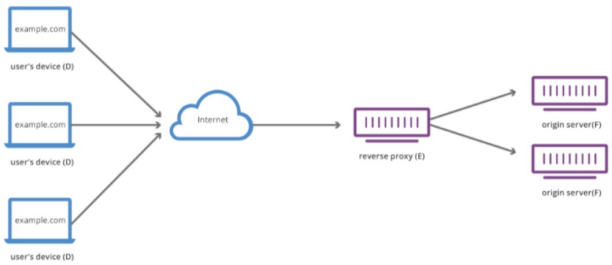
A forward proxy, often called a proxy, proxy server, or web proxy, is a server that sits in front of a group of client machines. When those computers make requests to sites and services on the Internet, the proxy server intercepts those requests and then communicates with web servers on behalf of those clients, like a middleman.

Forward Proxy Flow



- **To avoid state or institutional browsing restrictions**
- **To block access to certain content**
- **To protect their identity online**

Reverse Proxy Flow



- **Load balancing** / **Global Server Load Balancing** **(GSLB)**
- **Protection from attacks** - such as a DDoS attack. Instead the attackers will only be able to target the reverse proxy, such as Cloudflare's CDN, which will have tighter security and more resources to fend off a cyber attack.
- **Caching** - A reverse proxy can also cache content geographically, resulting in faster performance.

- Forward proxy - Request is first sent to forward proxy.
- Reverse proxy - The request is first sent to reverse proxy cache.

Caching Algorithms

i) FIFO ii) LRU iii) LFU

Presentation Layer

The presentation layer is the sixth layer in the OSI (Open Systems Interconnection) model, which is responsible for the syntax and semantics of the information transmitted between two systems. It acts as a translator and formatter, ensuring that data sent from the application layer of one system can be read by the application layer of another. Here's a detailed explanation of the presentation layer:

Key Functions of the Presentation Layer

1. **Translation**:
   - **Syntax Conversion**: Converts data from the application layer into a common format and vice versa. This allows systems with different data representation methods to communicate.
   - **Character Encoding**: Handles the conversion between different character encoding schemes such as ASCII, EBCDIC, and Unicode.
2. **Encryption and Decryption**:
   - **Data Security**: Encrypts data to protect it during transmission and decrypts it upon arrival. This ensures confidentiality and prevents unauthorized access.
   - **Encryption Algorithms**: Utilizes algorithms such as AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman).
3. **Compression and Decompression**:
   - **Data Compression**: Reduces the size of data to save bandwidth and transmission time. Common methods include ZIP, JPEG, and MP3 compression.
   - **Data Decompression**: Restores the compressed data to its original form for use by the application layer.
4. **Data Formatting**:
   - **Format Conversion**: Converts data to a format suitable for the receiving application, such as converting images from one format to another (e.g., JPEG to PNG).
   - **File Types**: Manages various data formats including multimedia files (audio, video), text files, and graphics.

Session Layer

The session layer is the fifth layer in the OSI (Open Systems Interconnection) model, responsible for managing sessions between applications on different devices. It ensures sessions are established, maintained, and terminated properly, facilitating reliable data exchange. Here's a concise explanation:

Key Functions

1. **Session Management**:
   - **Establishment**: Initiates sessions between devices, setting up necessary parameters.
   - **Maintenance**: Keeps sessions active and manages dialog between devices.
   - **Termination**: Closes sessions gracefully after communication is complete.
2. **Dialog Control**:
   - **Full-Duplex and Half-Duplex**: Manages bidirectional (full-duplex) or unidirectional (half-duplex) communication.
   - **Turn-Taking**: Ensures orderly communication by controlling which device sends or receives data.
3. **Synchronization**:
   - **Checkpoints and Recovery**: Uses checkpoints to resume communication from the last successful point in case of failure.
   - **Data Sequencing**: Ensures data packets are received and processed in the correct order.

Examples of Session Layer Protocols and Services

- **NetBIOS (Network Basic Input/Output System)**: Provides services related to the session layer for allowing applications on separate computers to communicate over a local area network.
- **PPTP (Point-to-Point Tunneling Protocol)**: Used for creating virtual private networks (VPNs), managing sessions for secure remote access.
- **RPC (Remote Procedure Call)**: Allows a program to execute a procedure (subroutine) on another address space (commonly on another physical machine).

Application: Online gaming, voice over IP, etc.

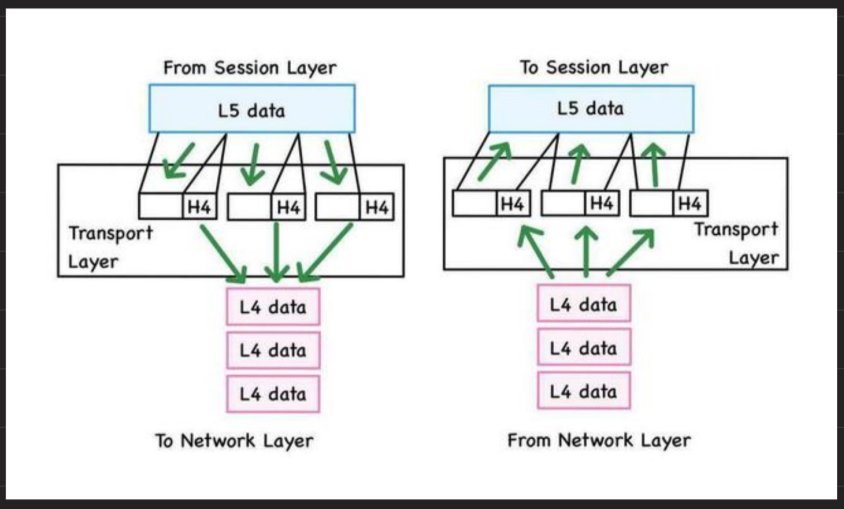
Transport Layer:

Content

- Functions
- Multiplexing / Demultiplexing
- TCP vs UDP
- Working of TCP (hand-shaking)
- Segment structures
- Flow and congestion control

provides logical communication between app processes running on different hosts.

NOTE: logical communication between hosts is provided by Network layer



Functions:

i) Splits message coming from Application layer into smaller units known as segments (Sender's side)
At receiver's side, transport layer put these segments back together into the original message.

ii) Uses TCP (transmission control protocol) for reliable data transfer.

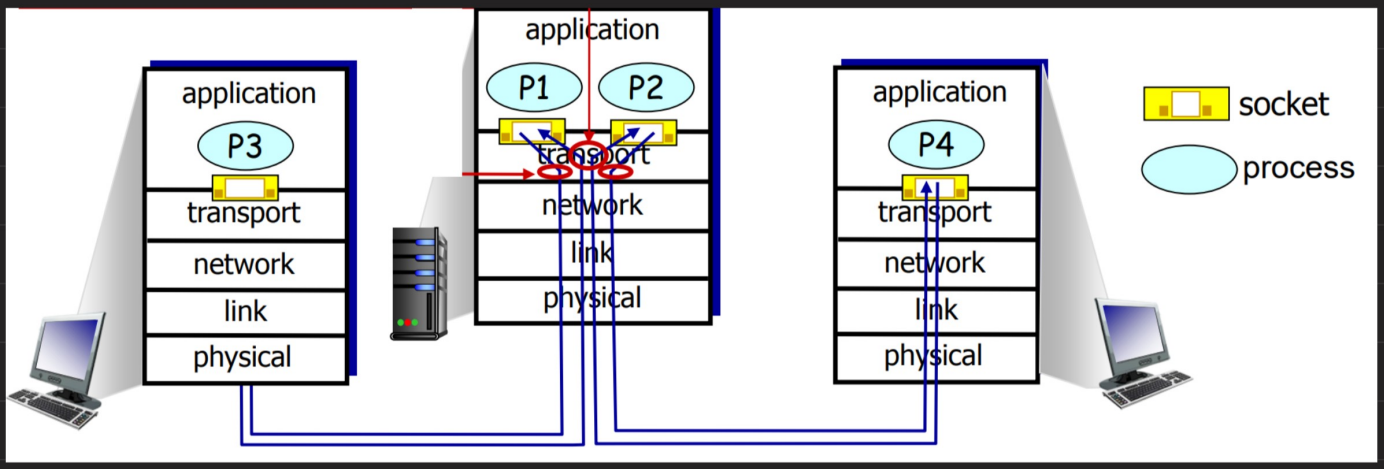
iii) Handles error detection and repair through Checksums.

Multiplexing and De-multiplexing

Socket : It is one endpoint of a two way communication link between two programs running on a network. (Present at session layer).

Multiplexing (at Sender): The process of handling data from multiple sockets, add transport header

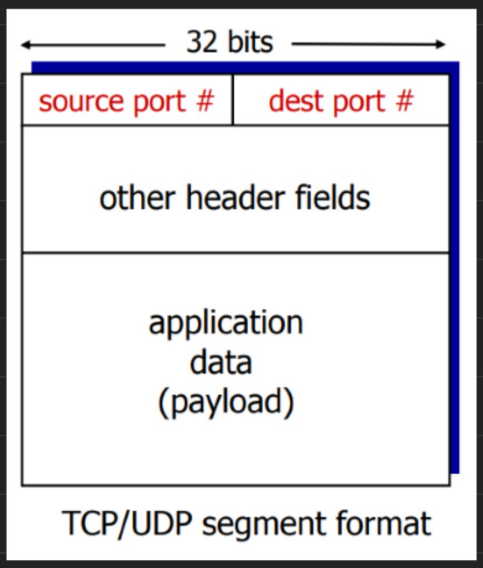
Demultiplexing (at receiver) : use header info to deliver received segments to correct socket.



How demultiplexing works ?

TCP socket is identified by 4-tuple:
- source IP address
- source port number
- destin. IP address
- dest port number

Each datagram has these 4-tuple.
Host uses IP addresses & port numbers to direct segment to appropriate socket.



TCP/UDP segment format

Connection-less transport UDP

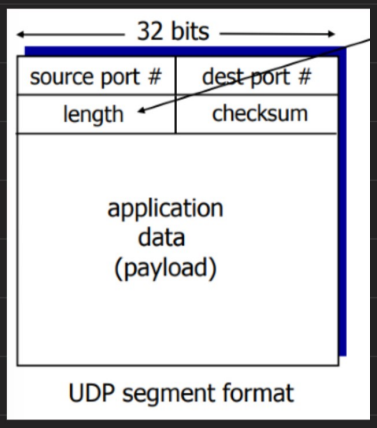
UDP - user datagram protocol. It provides best effort service.

- connection less : no handshaking b/w UDP sender and receiver each UDP segment is handled independently of other.

UDP segments may be lost and delivered out-of-order to app.

Why use UDP ?

i) Since no connection establishme. is required b/w senders and receiver (which adds delay), it is fast.



length in bytes of UDP segment including header

ii) Simple iii) Small header size (8 bytes)

iv) No congestion control: UDP can blast away as fast as desired.

v) Used when speed and size are more important than security and order. eg: Streaming multimedia apps, DNS (domain name system)

UDP- checksum : goal - detect errors in transmitted segment.

sender:
- treat segment contents, including header fields, as sequence of 16-bit integers
- checksum: addition (one's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

receiver:
- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?* More later ….

example: add two 16-bit integers

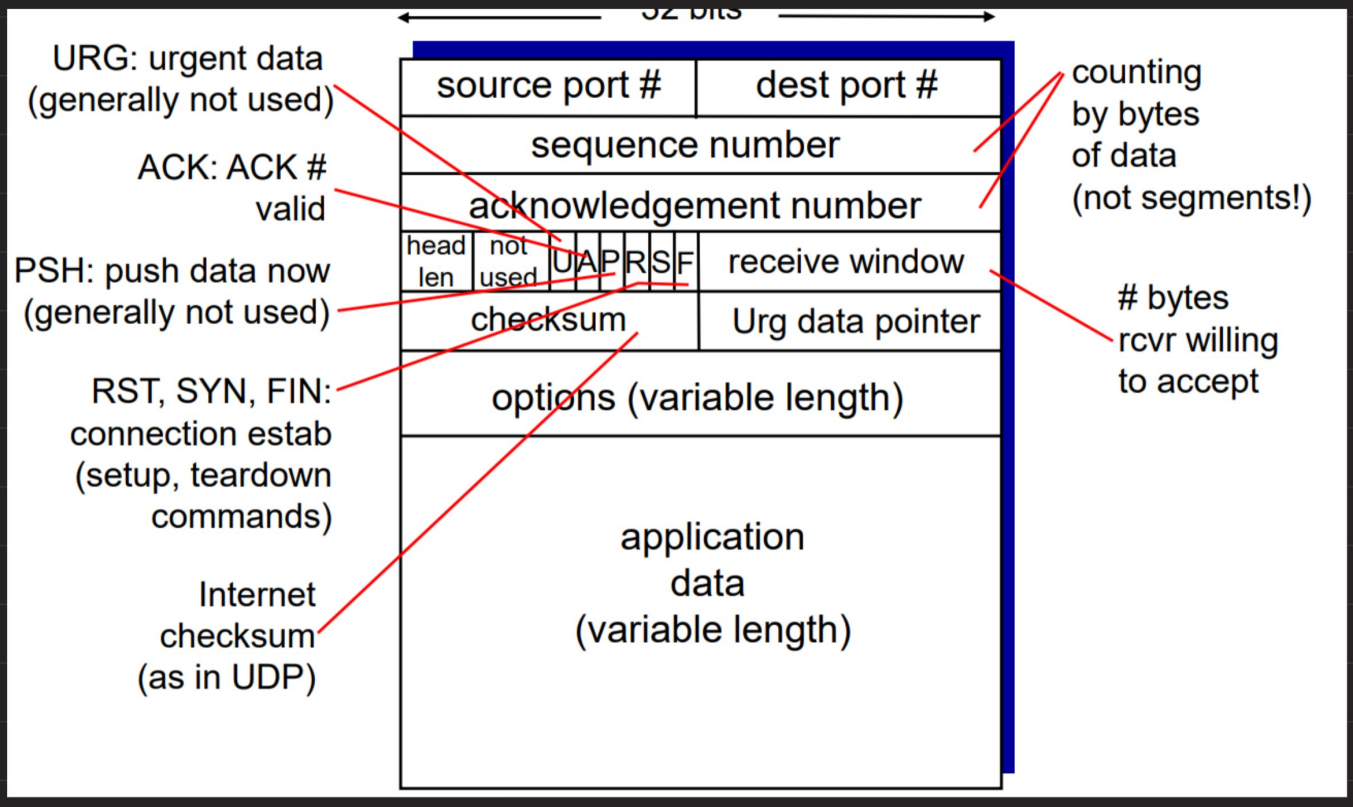
		1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
		1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
wraparound	(1)	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
sum		1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
checksum		0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

Note: when adding numbers, a carryout from the most significant bit needs to be added to the result

Still there can be errors : 2+5 (sender) = 3+4 (receiver)

TCP (transmission control protocol)

i) point-to-point : one sender / one receiver
ii) reliable - in order byte steam
iii) flow controlled - sender will not overwhelm receiver
iv) full duplex data transmission
v) connection oriented - handshaking blw sender / receiver

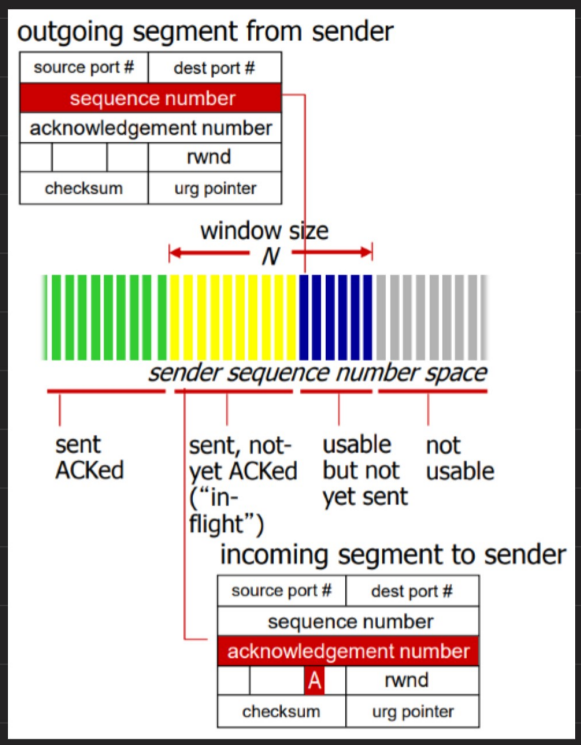


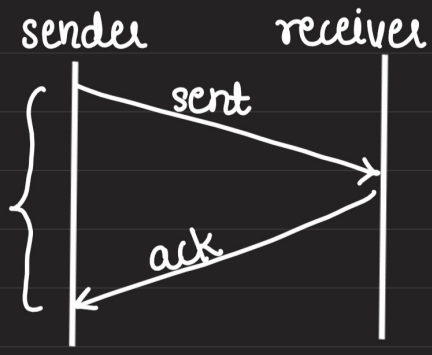
Working of TCP :

- Sender sends 'N' segments to the receiver and then waits for the ACKs (cumulative acknowledgements).
- Once ACKs for all these 'N' segments are received from the receiver, it sends the next 'N' segments.
- If ACKs is not received after waiting for a certain time period, then it retransmit segment that caused timeout. The same process continues.

Sequence numbers : byte stream 'number' of first byte in segment's data.

Acknowledgements : seq# of next byte expected from other side.

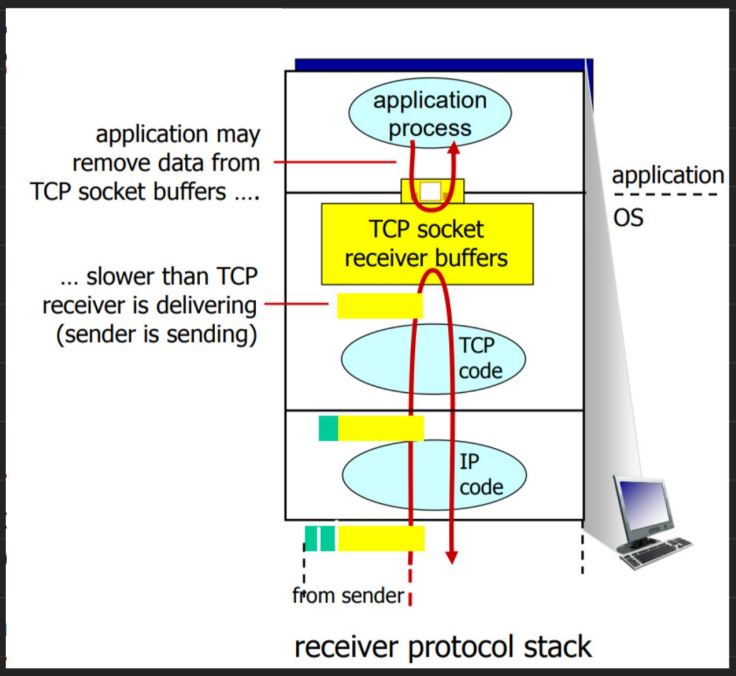




Round trip time (RTT)

Flow control in TCP :

Receiver controls sender, so sender wont overflow receiver's buffer by transmitting too much too fast

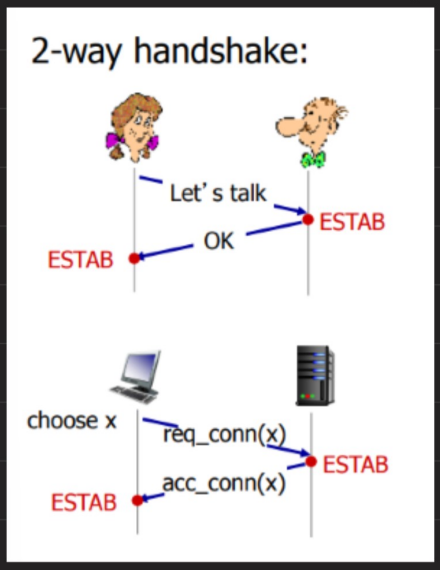


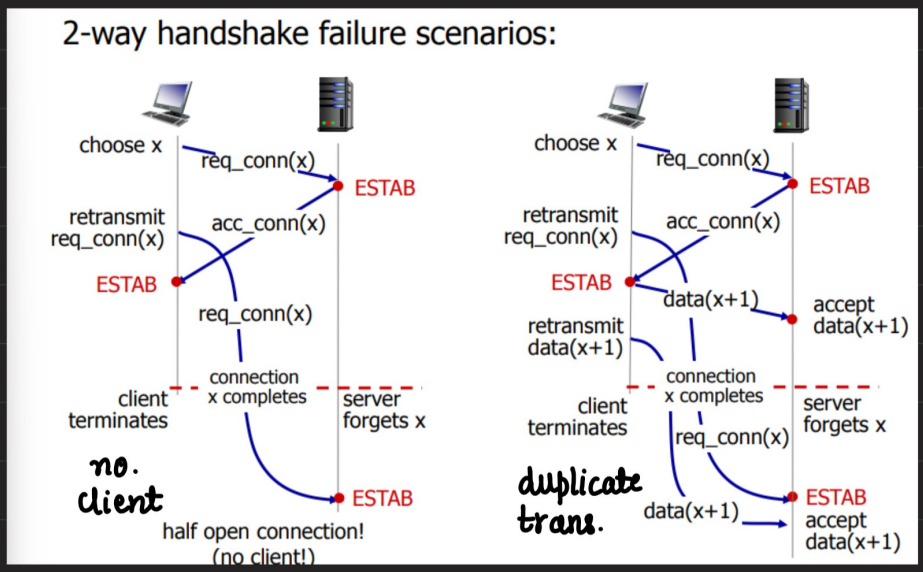
Note : congestion control is different from flow control.

Congestion - too many sources sending too much data too fast for network to handle.

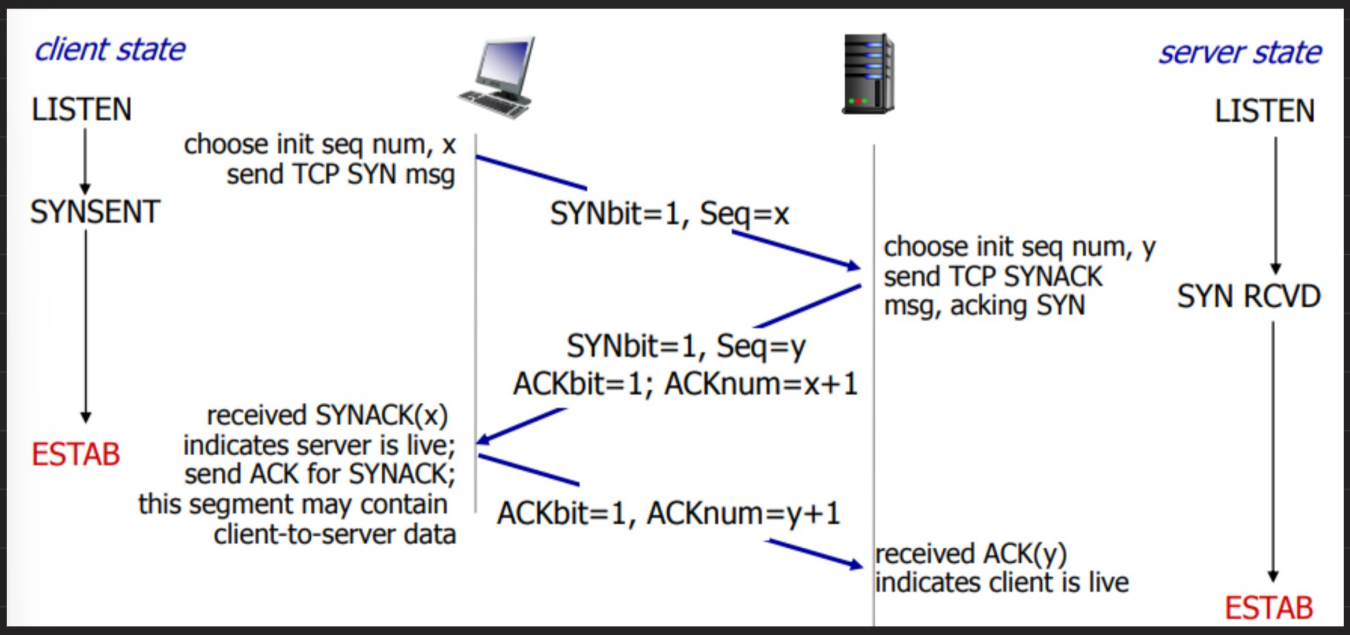
Connection Management

Before exchanging data, sender / receiver "handshake"
TCP does a 3-way handshake.





2 way handshake fails due to variable delays, retransmitted message. We can't see the other side.

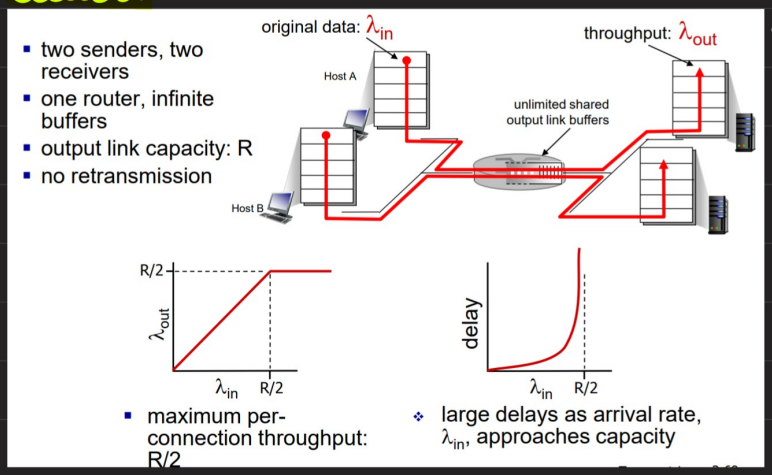


Closing a Network

- client, server each close their side of connection
  - send TCP segment with FIN bit = 1
- respond to received FIN with ACK
  - on receiving FIN, ACK can be combined with own FIN
- simultaneous FIN exchanges can be handled

Principle of congestion control:

Cause:



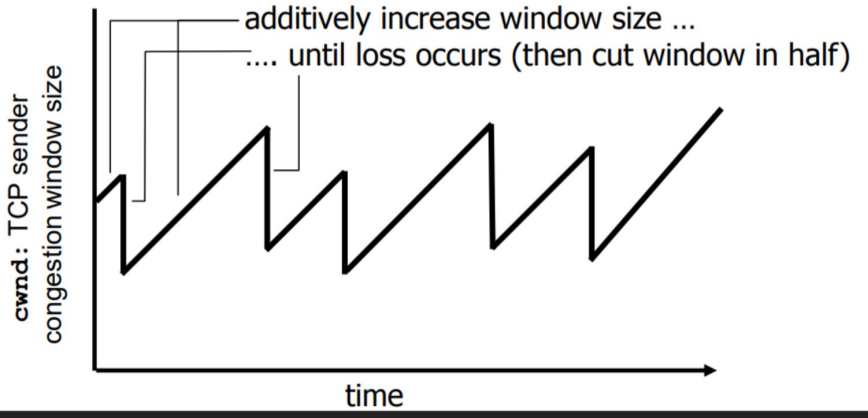
If there is finite buffer packets can be lost, dropped at router due to full buffers.

Senders only resends if packet is known to be lost.

TCP congestion control

- *approach:* sender increases transmission rate (window size), probing for usable bandwidth, until loss occurs
  - *additive increase:* increase **cwnd** by 1 MSS every RTT until loss detected
  - *multiplicative decrease:* cut **cwnd** in half after loss

AIMD saw tooth behavior: probing for bandwidth



Q) Difference b/w TCP and UDP.

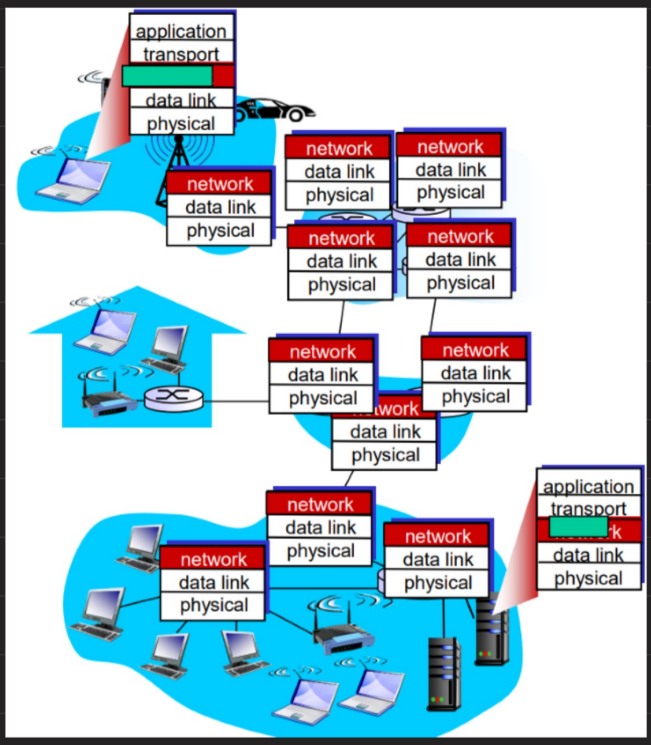
Ans:

TCP	UDP
TCP is a connection-oriented protocol	UDP is the connection-less protocol
TCP is reliable.	UDP is not reliable.
TCP supports error-checking mechanisms.	UDP has only the basic error-checking mechanism using checksums.
An acknowledgment segment is present.	No acknowledgment segment.
TCP is slower than UDP	UDP is faster, simpler, and more efficient than TCP.
Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in the User Datagram Protocol (UDP)
TCP has a (20-60) bytes variable length header.	The header length is fixed of 8 bytes.

TCP is reliable because
→ 3 way handshaking
→ ACKs.

Network Layer

- On sending side, encapsulates segments into datagrams.
- On receiving side, decapsulates datagrams into segments and passes it to transport layer.
- network layer protocols are present in every host, router



Two key network layer functions

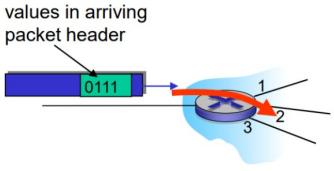
i) forwarding : move packets from router's input to appropriate router output.

ii) Routing : determine route taken by packets from source to destination.

Network Plane

Data plane

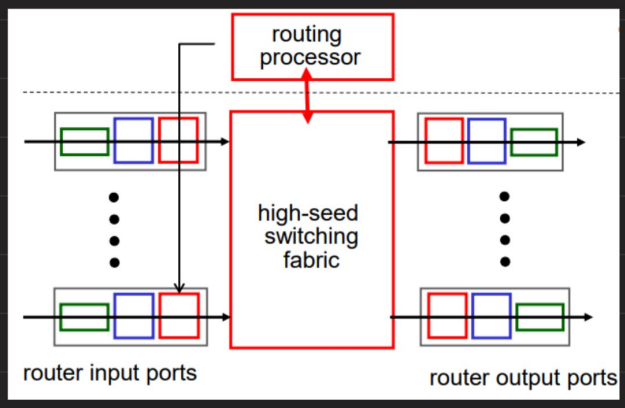
- local, per-router function
- determines how datagram arriving on router input port is forwarded to router output port
- forwarding function



Control Plane

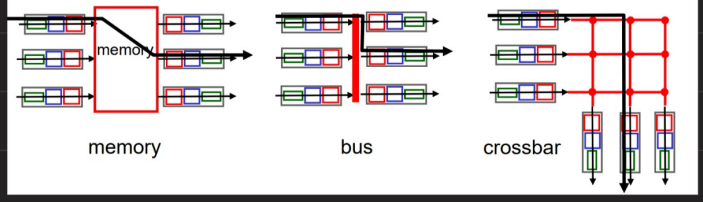
- network-wide logic
- determines how datagram is routed among routers along end-end path from source host to destination host
- two control-plane approaches:
  - *traditional routing algorithms:* implemented in routers
  - *software-defined networking (SDN)*: implemented in (remote) servers

Router

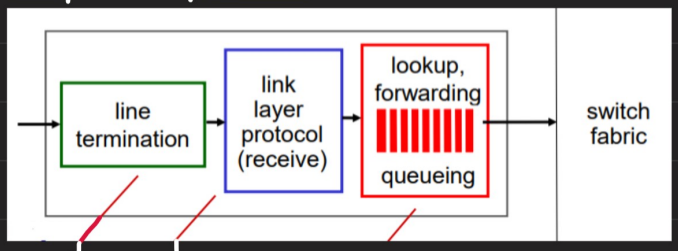


Switching fabrics : transfer packet from input buffer to output buffer.

- three types of switching fabrics



Input port

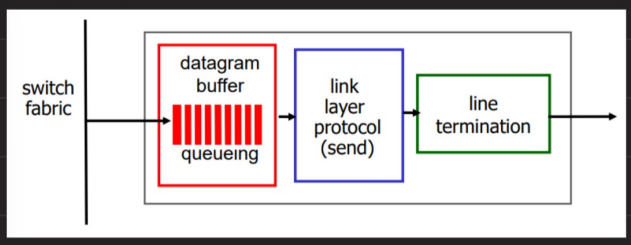


Phyical layer — Data link layer (Ethernet)

(bits reception)

- using header field, lookup output port using forwarding table.

Output Port



Buffering required when datagrams arrive from fabric faster than the transmission rate

Datagram (packets) can be lost due to congestion, lack of buffers

Forwarding from Input port to output port

Destination based forwarding

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

used when ranges divides

longest prefix matching

longest prefix matching

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

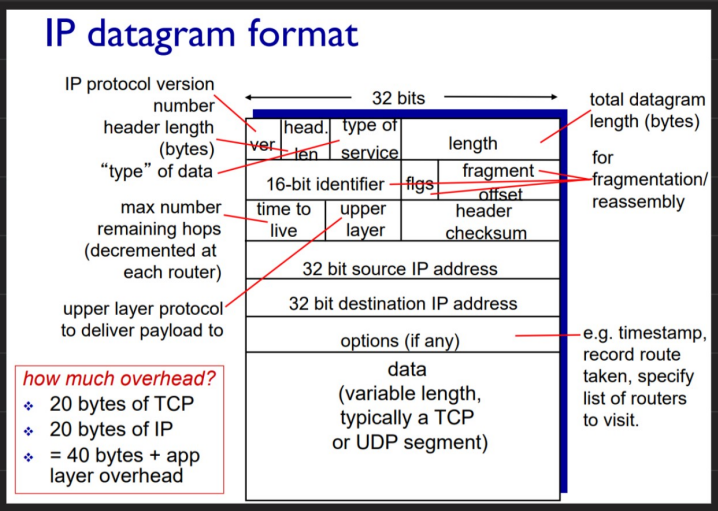
Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

examples:

DA: 11001000 00010111 00010110 10100001 — which interface? 0

DA: 11001000 00010111 00011000 10101010 — which interface? 1

Internet Protocol

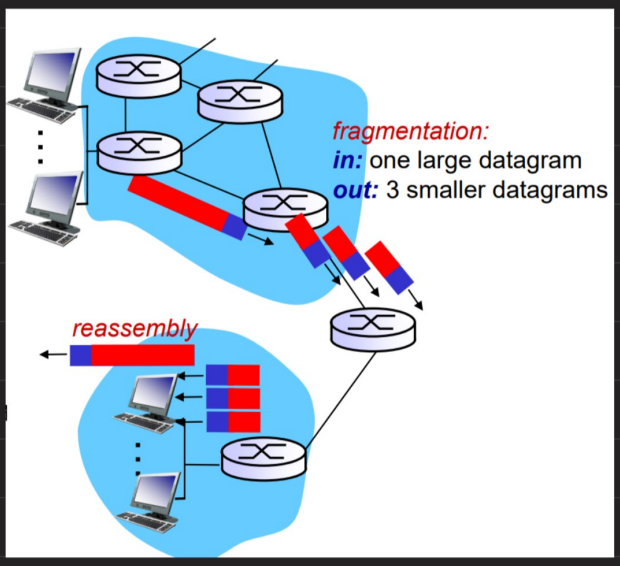


- source IP address
- dest. IP address
- TTL (time to live)
- checksum

made using 16 bit-identifiers

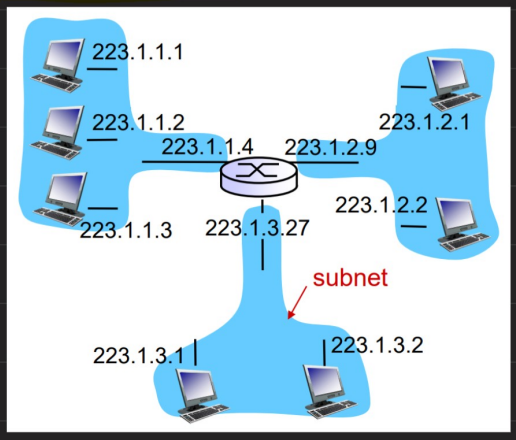
Fragmentation / Reassembly

- Network links might have maximum transfer size, so, large IP datagrams are divided or fragmented.
- Finally, at destination, they are reassembled. IP header bits are used to identify, order related fragments.



IPv4 addressing : 32-bit identifier for host, router interface (connection b/w host/router and physical link is called interface)

Subnets:

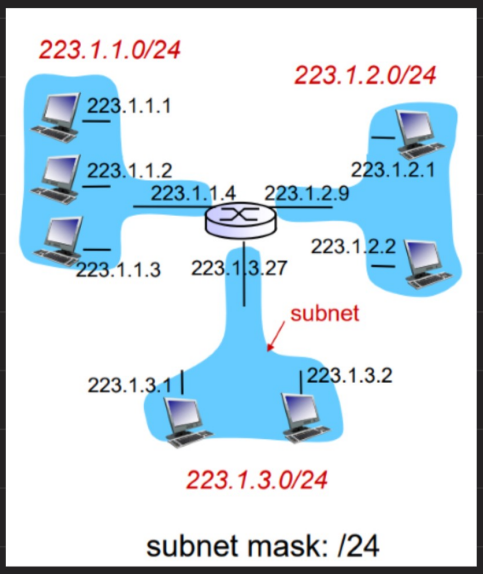


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Higher order bits (subnet part) | lower order (Host part)

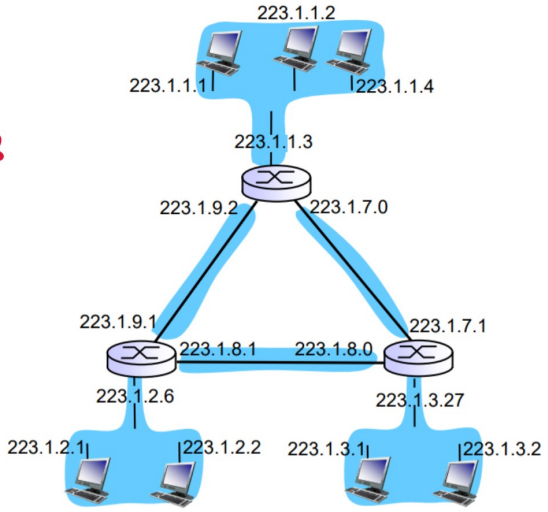
Subnet : Device interfaces with same subnet part of IP address can physically reach each other without intervening router.



Subnets

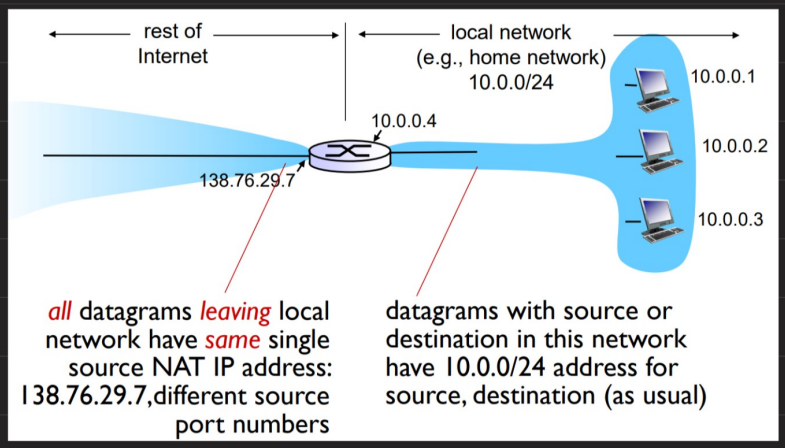
how many?

7 subnets



How to get IP address ?
Ans → DHCP (Dynamic Host configuration protocol) dynamically get address from a server

NAT (Network address translation)



- range of addresses no needed from ISP, just one IP add.
- enhances security
- can change address of devices in local network without notifyi: outside world.

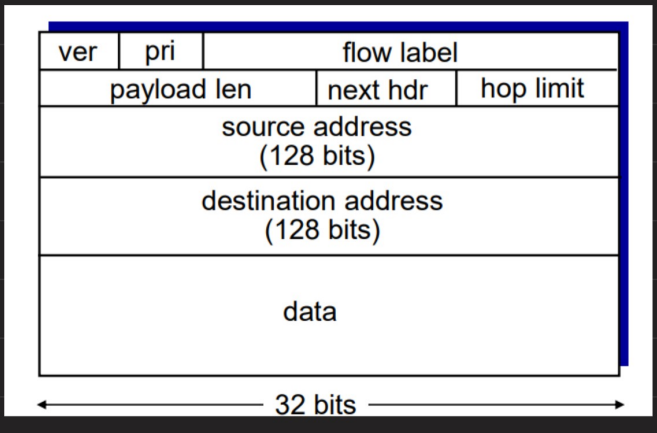
NAT is controversial, address shortage should be solved by IPv6.

IPv6

IPv6 datagram format:
- fixed-length 40 byte header
- no fragmentation allowed

- *checksum:* removed entirely to reduce processing time at each hop

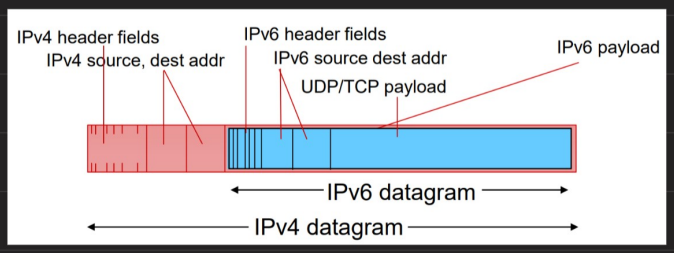
priority: identify priority among datagrams in flow

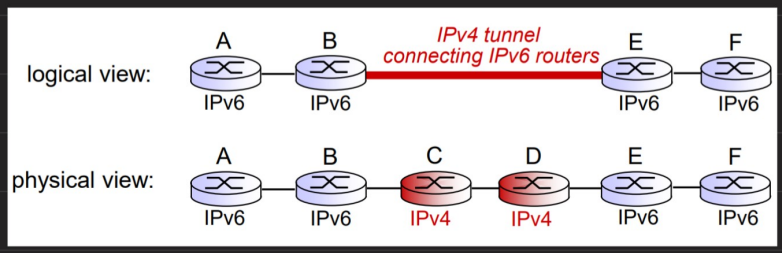


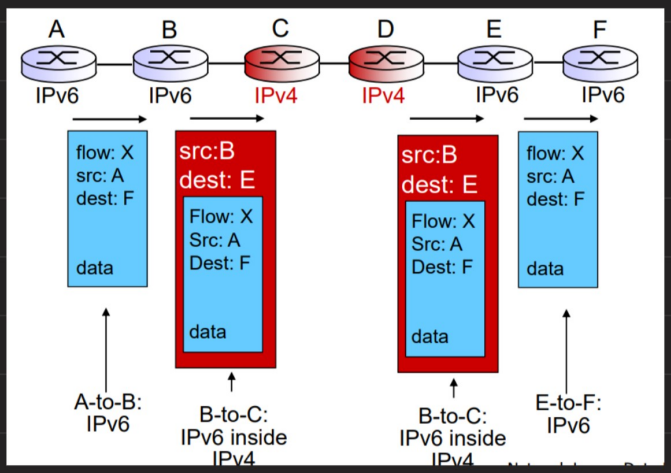
Transition from IPv4 to IPv6

- Not all routers can be upgraded simultaneously.

Tunneling : IPv6 datagram carried as payload in IPv4 datagram among IPv4 routers.



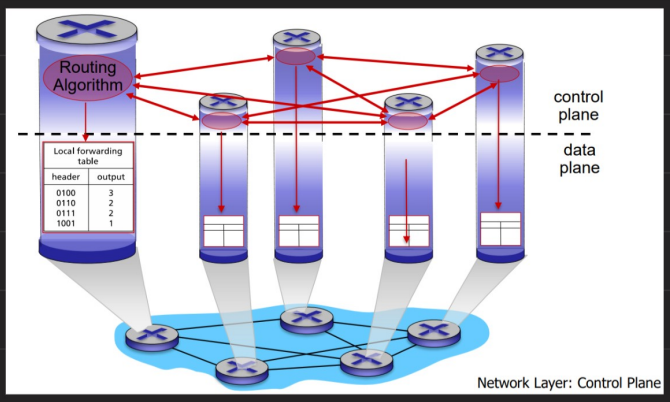




Control plane

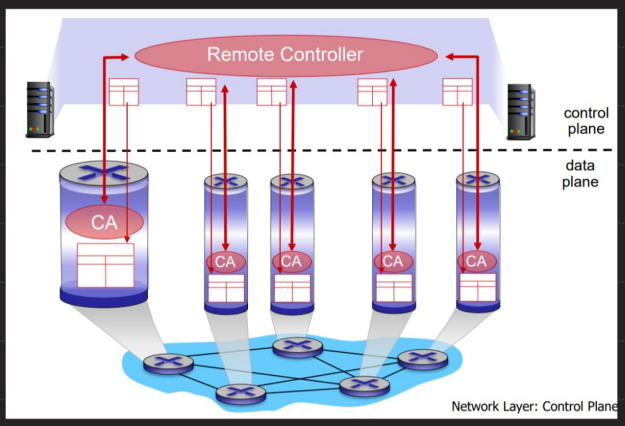
i) Per - Router control plane

Individual routing algorithm components in each router interacts with each other to compute forwarding tables.



ii) logically centralized control plane

A distinct controller interacts with local control agents (CAs) in routers to compute forwarding table



Routing Algorithms

i) Dijstra's Algorithm ii) Bellman - Ford algorithm, etc

SDN - software defined networking.

Data-link layer (DLL)

(Second layer from bottom) - responsible for the node to node delivery of data.

Two sublayers

a) logical link control (LLC) - deals with multiplexing, the flow of data among applications and other services.
LLC is responsible for providing error messages and acknowledgments as well.

b) Media Access control (MAC) - It manages the addressing of the frames.

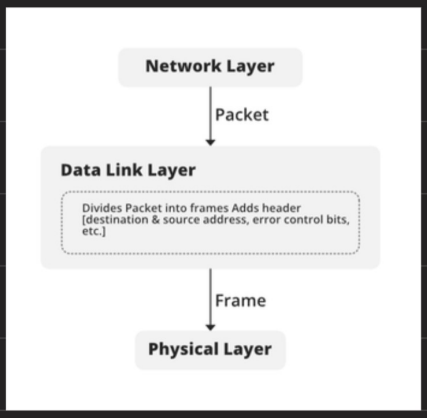
- **Framing (done by LLC)**

The packet received from the Network layer is known as a frame in the Data link layer.

At the sender's side, DLL receives packets from the Network layer and divides them into small frames, then, sends each frame bit-by-bit to the physical layer.

It also attaches some special bits (for error control and addressing) at the header and end of the frame.

At the receiver's end, DLL takes bits from the Physical layer organizes them into the frame, and sends them to the Network layer.



- **Addressing (done by MAC)**

The data link layer encapsulates the source and destination's MAC address/ physical address in the header of each frame to ensure node-to-node delivery. MAC address is the unique hardware address that is assigned to the device while manufacturing.

DLL has 3 more functions :

i) Error control - (LLC)

Data can get corrupted due to various reasons like noise, attenuation, etc. So, it is the responsibility of the data link layer, to detect the error in the transmitted data and correct it using error detection and correction techniques respectively.

DLL adds error detection bits into the frame's header, so that receiver can check received data is correct or not. It adds reliability to physical layer by adding mechanisms to detect and retransmit damaged or lost frames.

ii) Flow control -

If the receiver's receiving speed is lower than the sender's sending speed, then this can lead to an overflow in the receiver's buffer and some frames may get lost. So, it's the responsibility of DLL to synchronize the sender's and receiver's speeds and establish flow control between them.

iii) Access control -

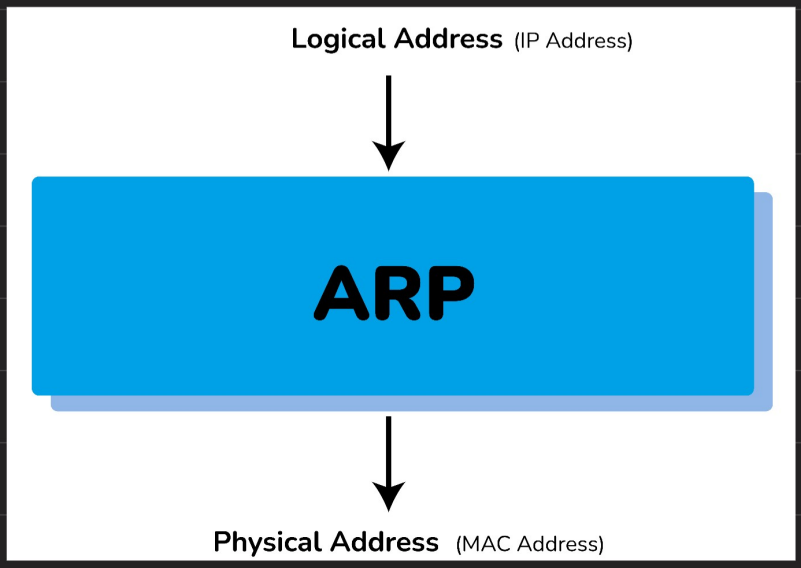
When multiple devices share the same communication channel there is a high probability of collision, so it's the responsibility of DLL to check which device has control over the channel and CSMA/CD and CSMA/CA can be used to avoid collisions and loss of frames in the channel.

ARP protocol

ARP is Address Resolution Protocol.

It is a network-level protocol used to convert the logical address i.e. IP address to the device's physical address i.e. MAC address.

It can also be used to get the MAC address of devices when they are trying to communicate over the local network.



What are Unicasting, Anycasting, Multicasting and Broadcasting?

- **Unicasting:** If the message is sent to a single node from the source then it is known as unicasting. This is commonly used in networks to establish a new connection.
- **Anycasting:** If the message is sent to any of the nodes from the source then it is known as anycasting. It is mainly used to get the content from any of the servers in the Content Delivery System.
- **Multicasting:** If the message is sent to a subset of nodes from the source then it is known as multicasting. Used to send the same data to multiple receivers.
- **Broadcasting:** If the message is sent to all the nodes in a network from a source then it is known as broadcasting. DHCP and ARP in the local network use broadcasting.

ARP uses Broadcasting.

CRC and generator Polynomial

CRC (Cyclic redundancy test) is a method of detecting accidental changes/errors in the communication channel.

→ CRC uses generator polynomial which is available on both sender and receiver side.

eg: $x^3 + x + 1$ (degree = 3) represents key 1011

$x^2 + 1$ (degree = 2) represents key 101

n: no. of bits in data to be sent from sender side

k: no of bits in the key obtained from generator polynomial

→ Sender Side (generation of encoded data)

i) Append K-1 zeros in the end of original data

ii) use modulo-2 division of data by key to get remainder

iii) Append remainder to the original data to get encoded data

→ Receiver side (check if error is introduced in transmission)

Perform modulo-2 division again and if remainder is 0, then there are no error.

→ Modulo - 2 division: Instead of substraction, we use XOR here

eg: Data - 100100 , key - 1101

appending , 4-1 = 3 zeros at the end of data : 100100000

```
        111101
    ____________
1101 ) 1001 00000
     ^ 1101
       ----
        1000
       ^ 1101
         ----
         1010
          1101
          ----
          1110
           1101
           ----
           0110
           0000
           ----
            1100
            1101
            ----
             001
```

Here

Quotient - 111101

Remainder - 001

Hence,

encoded data =

data + remainder

= 100100001

If receiver receives the correct encoded data, then divinding it by 1101 will give remainder as zero.

```
        111101
    ____________
1101 ) 100100001
       1101
       ----
        1000
        1101
        ----
        1010
        1101
        ----
        1110
        1101
        ----
        0110
        0000
        ----
        1101
        1101
        ----
         000   } zero remainder
```

If the remainder is not equal to zero there there is error in data transmission

CRC is by default implemented in the Data Link layer.

Can be used in other layers but we will need to manually implement it

Significance of Physical and Data - link layer

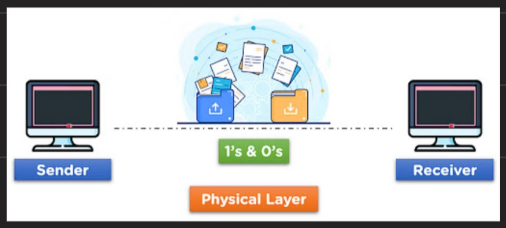
1. Physical layer

Bottom-most layer in OSI (open system Inter-connection) Model. It consists of various network components such as power plugs, connectors, receivers, cable types, Ethernet, Hubs, etc.

Functions

i) It provides an interface b/w devices (PCs) and transmission medium.

ii) It is responsible for converting the data frames received from the Data link layer into data bits of 1's and 0's for transmission over the network.



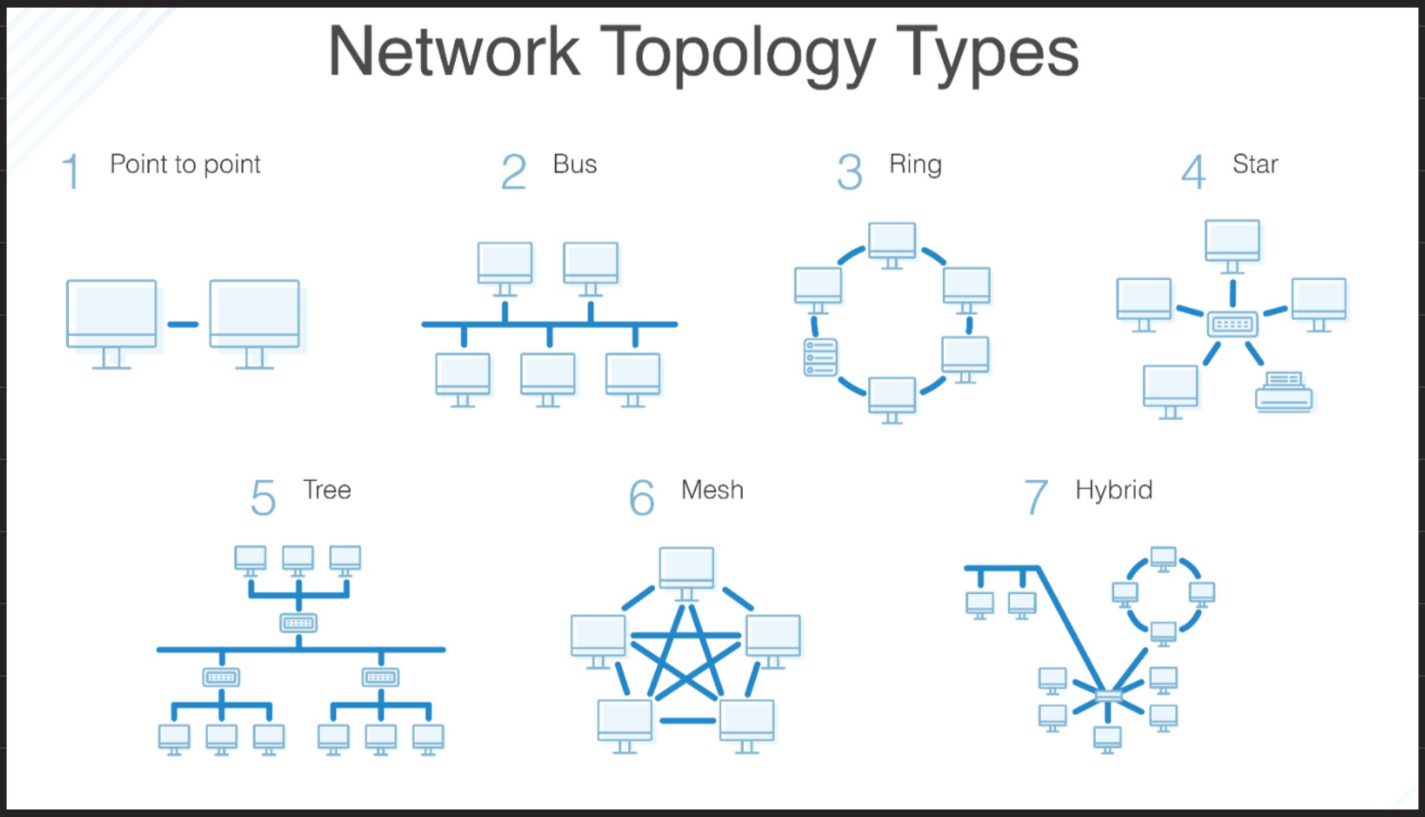
iii) Modulation & De-modulation are done by Physical layer.

iv) It controls the the data rate (how many bits a sender can send per-second)

Modulation is defined as the process of superimposing a low-frequency signal on a high-frequency carrier signal.

Demodulation is defined as extracting the original information-carrying signal from a modulated carrier wave.

Physical Topology



Mesh Topology: Each device is connected to every other device via dedicated channel.
For, N devices, no. of channels = $^{N}C_2 = N(N-1)/2$

Advantages of Mesh Topology

- Communication is very fast between the nodes.
- Mesh Topology is robust.
- The fault is diagnosed easily. Data is reliable because data is transferred among the devices through dedicated channels or links.
- Provides security and privacy.

Disadvantages of Mesh Topology

- Installation and configuration are difficult.
- The cost of cables is high as bulk wiring is required, hence suitable for less number of devices.
- The cost of maintenance is high.

used in milatary comm. s/m & aircraft navigation s/m.

Star topology

All devices are connected to a single hub through cable. Co-axial cables are used for connecting the devices to the central node (Hub)

eg: LAN in an office.

for N devices, no. of connections = N , so, it is easy to set up.

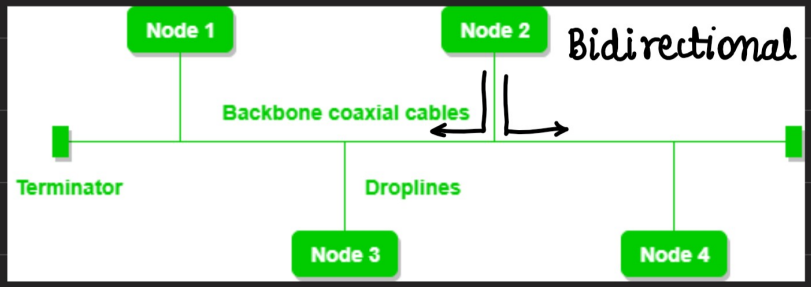
In Star Topology, many popular Ethernet LAN protocols are used as CD(Collision Detection), CSMA (Carrier Sense Multiple Access), etc.

Advantages • Cost effective ,• Easy fault identifi. Robust, if one link fails , other links are not affected

Disadvantage : If hub fails, whole s/m crash.

Bus Topology (eg: Ethernet LAN)

No. of cables = N + 1 (backbone coaxial cable)

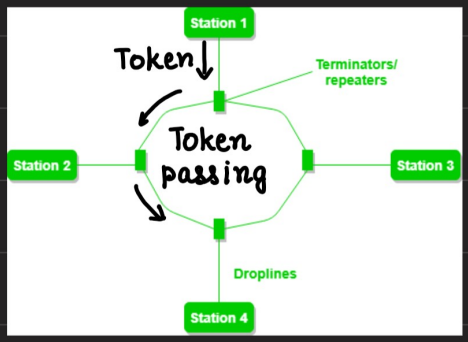


In Bus Topology, various MAC (Media Access Control) protocols are followed by LAN ethernet connections like TDMA, Pure Aloha, CDMA, Slotted Aloha, etc.

Advantage - cost effective
Disadvantage - If common cable fails, then the whole system will crash.

Ring Topology

In a Ring Topology, it forms a ring connecting devices with exactly two neighboring devices. A number of repeaters are used for Ring topology with a large number of nodes, because if someone wants to send some data to the last node in the ring topology with 100 nodes, then the data will have to pass through 99 nodes to reach the 100th node. Hence to prevent data loss repeaters are used in the network.



→ data flow is uni-directional
→ can be made bi-directional by having 2 connections (Dual - Ring Topology)

Advantages of Ring Topology

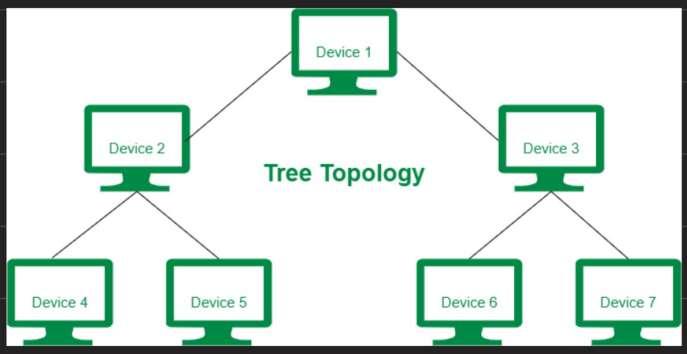
- The data transmission is high-speed.
- The possibility of collision is minimum in this type of topology.
- Cheap to install and expand.
- It is less costly than a star topology.

Disadvantages of Ring Topology

- The failure of a single node in the network can cause the entire network to fail.
- Troubleshooting is difficult in this topology.
- The addition of stations in between or the removal of stations can disturb the whole topology.
- Less secure.

Tree topology

Hierarchical flow of data. eg: Hierarchy in a large organization. At the top of the tree is the CEO, who is connected to the different departments.



In Tree Topology, protocols like DHCP and SAC (Standard Automatic Configuration) are used.

Advantages of Tree Topology

- It allows more devices to be attached to a single central hub thus it decreases the distance that is traveled by the signal to come to the devices.
- It allows the network to get isolated and also prioritize from different computers.
- We can add **new devices to the existing network.**
- **Error detection** and **error correction** are very easy in a tree topology.

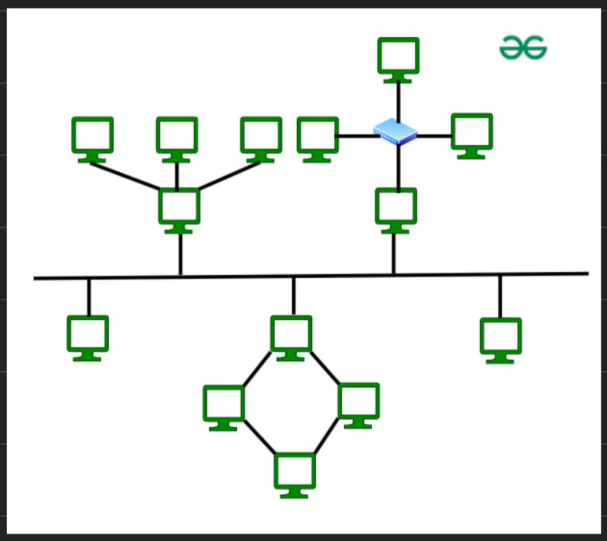
Disadvantages of Tree Topology

- If the central hub gets fails the entire system fails.
- The cost is high because of the cabling.
- If new devices are added, it becomes difficult to reconfigure.

Hybrid Topology

This topological technology is the combination of all the various types of topologies we have studied above. Hybrid Topology is used when the nodes are free to take any form.

Each individual topology uses the protocol that has been discussed earlier.



Hybrid topology is used in university campus network.

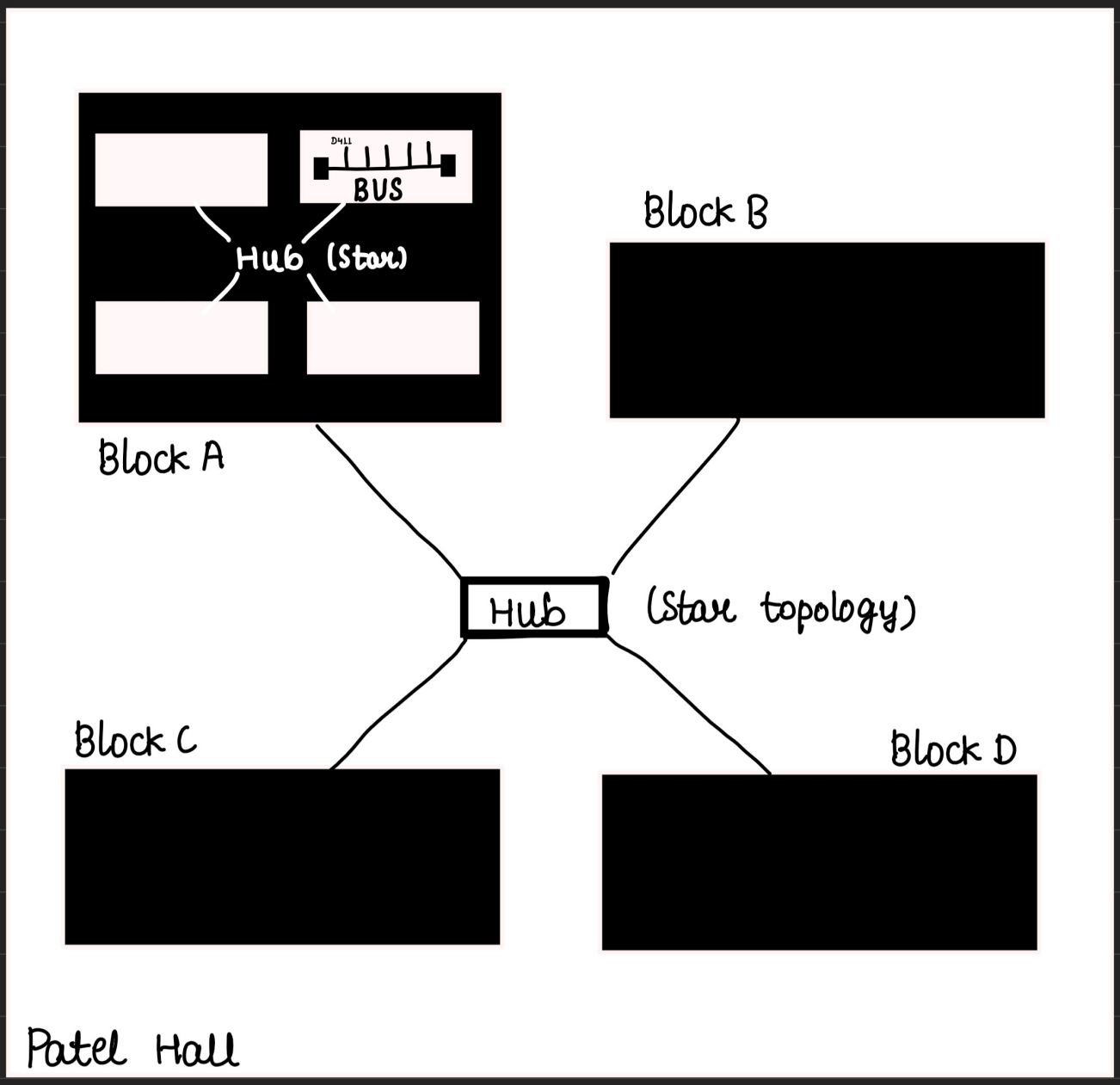
Advantages of Hybrid Topology

- This topology is **very flexible**.
- The size of the network can be easily expanded by **adding new devices.**

Disadvantages of Hybrid Topology

- It is challenging **to design the architecture** of the Hybrid Network.
- **Hubs** used in this topology are **very expensive.**
- The infrastructure cost is very high as a hybrid network **requires a lot of cabling and network devices**.

eg of Hybrid :



one node of the tree topology with the CIC node at the top.

Line Configuration

- **Point-to-Point configuration:** In Point-to-Point configuration, there is a line (link) that is fully dedicated to carrying the data between two devices.
- **Multi-Point configuration:** In a Multi-Point configuration, there is a line (link) through which multiple devices are connected.

Modes of Transmission Medium

1. **Simplex mode:** In this mode, out of two devices, only one device can transmit the data, and the other device can only receive the data. Example- Input from keyboards, monitors, TV broadcasting, Radio broadcasting, etc.
2. **Half Duplex mode:** In this mode, out of two devices, both devices can send and receive the data but only one at a time not simultaneously. Examples- Walkie-Talkie, Railway Track, etc.
3. **Full-Duplex mode:** In this mode, both devices can send and receive the data simultaneously. Examples- Telephone Systems, Chatting applications, etc.

Interview Question

Which topology is best for large networks?

For large networks, mesh and tree topologies are often preferred. Mesh topology offers high reliability and redundancy, while tree topology supports scalability and efficient data organization.

Compare the hub vs switch

Hub	Switch
Operates at Physical Layer	Operates at Data Link Layer
Half-Duplex transmission mode	Full-Duplex transmission mode
Ethernet devices can be connectedsend	LAN devices can be connected
Less complex, less intelligent, and cheaper	Intelligent and effective
No software support for the administration	Administration software support is present
Less speed up to 100 MBPS	Supports high speed in GBPS
Less efficient as there is no way to avoid collisions when more than one nodes sends the packets at the same time	More efficient as the collisions can be avoided or reduced as compared to Hub