Chapter 1

INTERNET:

* Connects billions of computing devices (it includes computers, smart phones, smart appliances or any devices that has the need to connect to the internet)
* Basically internet has three components,

1. Client
2. Browser
3. Server

* Client is the end user or service provider : for example, client can be a person who is requesting for a google web service and google on the other hand provides services and products that utilize the internet, it acts as a client to the internet infrastructure.
* Browser is responsible for converting the bundle of code to a presentable page.
* Server stores the bundles of code and based of IP address the requested files are sent to the host systems.

Route aggregation/ address aggregation: the ability to use a single prefix to advertise multiple networks.

Section 1.1

## **Introduction**

**In order to understand computer networks, we first need to know what the internet is. The internet is a giant global network that connects billions of devices. It is not a single entity but rather a collection of ISPs (Internet Service Providers).**

**Computer networks explain how devices (computers, phones, smart appliances, etc.) are connected and how information and resources are shared.**

## **Hosts and End Systems**

**On the internet, devices are known as hosts and end systems. Both send and receive data, but when do they become different?**

* **Host refers to any device with an IP address that can communicate on a network, including end systems or devices that forward data (such as routers or switches, which are not end systems).**
* **End system refers to the devices that are the final source or the destination of the data.**

### **Example:**

* **Your laptop sending an email to your friend → Source (End System)**
* **Your friend's laptop receiving the email → Destination (End System)**

**End systems initiate and end a conversation but do not forward messages in between.**

**Note: All end systems are hosts, but not all hosts are end systems.**

## **Communication Links and Packet Switching**

**End systems are connected by a network of:**

1. **Communication Links – These can be physical (wired) or wireless connections.**
2. **Packet Switches – Devices responsible for moving data but unaware of the contents inside the packet.**

**Each communication link has a different transmission rate. When an end system has data to send or receive, it first breaks the data into parts, resulting in packets. These packets are forwarded within the network using packet-switching devices such as:**

* **Routers - l3**
* **Link-layer switches (wired or wireless) - l2**
* **A Link Layer switch operates at Layer 2 (Data Link Layer), while a router works at Layer 3 (Network Layer) of the OSI model.**
* **A switch forwards frames based on MAC addresses, whereas a router routes packets based on IP addresses.**
* **A switch uses a MAC address table to learn and forward data within a network, while a router relies on a routing table to determine the next hop for packets.**
* **A switch connects devices within the same local network (LAN), whereas a router connects different networks, such as a LAN to WAN or LAN to LAN.**
* **A switch performs switching, forwarding data based on MAC addresses, while a router performs routing, forwarding packets based on IP addresses.**

## **Route/Path in Networking**

**A route/path is the sequence of devices through which data packets travel from the source to the destination.**

## **Protocols in Computer Networks**

**Protocols specify how data is:**

* **Formatted**
* **Transmitted**
* **Error-handled**

**The internet runs on protocols to control the sending and receiving of information. The most important protocols include:**

* **TCP (Transmission Control Protocol)**
* **IP (Internet Protocol)**

## **End System Socket Interface**

**The end system socket interface defines a set of rules that the sending program must follow so that the internet delivers the data to the exact destination program.**

Section 1.3

the network core:

the network core is the central part that handles the main job of moving data around. it's made of devices that connect different parts of the network and make sure data gets from source to destination by managing all the traffic in between (refers to all the data packets traveling through the network from source to destination, traveling through devices like routers and switches) as we know, data is broken in smaller chunks called packets and this method of sending data over a network is called packet switching. it basically breaks data into packets, send them independently, reassemble at the destination. packets travel through communication links and packet switches like routers and link layer switches.

\*handshake in networking: a process where end systems establish a connection before exchanging data. it ensures that both systems are ready to communicate. packets include port numbers to help the receiver identify the correct application or service. store and forward transmission: what is it? a packet switch stores an entire packet before forwarding it to the next link(wired or wireless), it means, the switch needs to receive the full packet before it can start sending it to the next device.

forwarding tables and routing protocols (works at layer 3)

Forwarding table maps destination address, it has destination ip address, next hop(the device where the packet should be sent), outgoing interface(router), metric(value that determines the best route) and protocol.

A routing protocol automatically sets the forwarding table figuring out shortest path a packet can take to reach the destination.

Queuing delays and packet loss

packets are transmitted through outbound links, each link has an output buffer and queue where packets are stored temporarily if the outbound link is busy. When too many packets are sent to single attached link, they are stored in a queue called queuing delays. Packets might suffer packet loss as buffer has an infinite space where either the arriving packet or the already queued packet is dropped.

Circuit switching:

It establishes a dedicated line for the communication between two devices, the resources are specifically allotted to that line.

It is mostly seen in telephonic conversations, where a dedicated line is established.

Multiplexing:

The circuit is implemented using either frequency division multiplexing or time division multiplexing.

FDM: it establishes connection on a different spectrum of frequency. The link dedicates a frequency band to each for the rest of the duration.

Bandwidth : the width of the band or the amount of data that can be transmitted in a given period of time.

The higher the bandwidth, the higher the information can be transmitted and downloaded.

Time division multiplexing: each circuit gets the dedicated bandwidth periodically after brief intervals of time

Packet switching

1. More efficient as resources are not wasted
2. Better sharing of transmission capacity
3. But it has end to end delays.

Network of networks:

End systems connect to the internet via isp(internet service provider)

There are different levels of ISP’s

1. Tier-1: Global ISP -owns the fiber optic cables or satellites that connects internet.
2. Tier-2: Regional ISP - buys bandwidth from the global ISP, mainly focuses on providing the internet access to a certain geographic location.
3. Tier-3: Local ISP - it buys bandwidth from the tier 1 or 2 and mainly focuses on providing internet to the local customers tailoring a customed plan.

Internet exchange point: where multiple ISP’s meet each other to exchange internet traffic.

Section 1.4

DELAY LOSS AND THROUGHPUT IN PACKET SWITCHED NETWORKS.

When packets are transmitted, along the path packets suffer delay.

The most prominent types of delays:

Processing delay: the time taken to examine the packet’s header decide where to send the packet.

Queuing delay: time the packet needs to wait if other packets are ahead of it before being transmitted.

Transmission delay: time taken to transmit all the bits of packet onto the outbound link.

Propagation delay: time taken for the packet to travel physically to the next router.

Queueing delay and packet loss: the packets relatively at the last of the queue suffer packet loss compared to the packet ahead.

Packet loss increases as the intensity of the traffic increases.

End to end delay: No. of nodal delays(processing delay + queuing delay + transmission delay + propagation delay

TRACEROUTE: the no of hops taken to reach the destination ip address.

Example: Tracing route to google.com [216.58.196.174]

over a maximum of 30 hops:

1 9 ms 5 ms 5 ms 172.21.148.1

2 17 ms 6 ms 6 ms 192.168.41.105

3 5 ms 6 ms 5 ms 192.168.8.17

4 7 ms 6 ms 5 ms noc-cr-in.comp.iith.ac.in [103.232.241.2]

5 8 ms 6 ms 6 ms noc-cn-in.comp.iith.ac.in [10.119.254.121]

6 27 ms 27 ms 27 ms 10.160.24.5

7 27 ms 27 ms 29 ms 10.255.232.226

8 38 ms 26 ms 26 ms 10.119.73.122

9 33 ms 34 ms 33 ms 72.14.195.128

10 27 ms 28 ms 30 ms 216.239.43.131

11 23 ms 22 ms 22 ms 216.239.43.235

12 25 ms 23 ms 22 ms maa03s31-in-f14.1e100.net [216.58.196.174]

Throughput : the amount of data that is successfully transmitted.

Bandwidth vs Throughput : the amount of that can be transmitted vs the amount of data that successfully got transmitted

Ex: the bandwidth of your wifi could be 30mbps but your device could only get 15mbps.

Factors affecting throughput:

1. Network congestion
2. Weak signal
3. Bandwidth of your connected network etc

Bottleneck link: data travels through multiple links, if one link path has a much lower bandwidth compared to the others, then the entire connection speed is limited. This is known as bottleneck line.

\*traceroute can be used to identify the bottleneck link and it can be fixed to ensure the speed limit is not congested.

#### **identifying Bottleneck Links**

* **Normal Latency:** 0-50ms (local network & ISP backbone)
* **Moderate Latency:** 50-150ms (regional/national routing)
* **High Latency:** 150ms+ (possible bottleneck or long-distance routing)

Tracing route to yahoo.com [74.6.231.21]

over a maximum of 30 hops:

1 10 ms 8 ms 12 ms 172.21.148.1

2 7 ms 9 ms 9 ms 192.168.41.105

3 14 ms 15 ms 6 ms 192.168.8.17

4 7 ms 6 ms 6 ms noc-cr-in.comp.iith.ac.in [103.232.241.2]

5 6 ms 5 ms 6 ms noc-cn-in.comp.iith.ac.in [10.119.254.121]

6 9 ms 8 ms 10 ms 10.160.24.5

7 24 ms 10 ms 10 ms 10.255.221.33

8 31 ms 10 ms 16 ms dsl-tn-085.99.246.61.airtelbroadband.in [61.246.99.85]

9 125 ms 116 ms 115 ms 116.119.49.38

10 \* \* \* Request timed out.

11 132 ms 116 ms \* 130.117.14.54

12 150 ms 193 ms 135 ms prs-bb1-link.ip.twelve99.net [62.115.124.54]

13 142 ms 140 ms 139 ms ldn-bb1-link.ip.twelve99.net [62.115.135.24]

14 347 ms 243 ms 246 ms nyk-bb5-link.ip.twelve99.net [62.115.139.244]

15 \* \* \* Request timed out.

16 359 ms 249 ms 354 ms chi-b23-link.ip.twelve99.net [62.115.126.45]

17 231 ms 230 ms 232 ms yahooholdings-ic-322225.ip.twelve99-cust.net [62.115.36.149]

18 360 ms 258 ms 350 ms ae-0.pat2.nez.yahoo.com [209.191.64.216]

19 407 ms 313 ms 249 ms et-0-1-1.msr1.ne1.yahoo.com [216.115.105.185]

20 358 ms 252 ms 364 ms et-8-0-0.clr1-a-gdc.ne1.yahoo.com [98.138.2.7]

21 330 ms 254 ms 348 ms lo0.fab5-2-gdc.ne1.yahoo.com [98.138.51.4]

22 330 ms 308 ms 250 ms usw2-1-lbd.ne1.yahoo.com [98.138.97.157]

23 389 ms 296 ms 413 ms media-router-fp74.prod.media.vip.ne1.yahoo.co

Hop 10 has a dropped packet

Hop 14 has huge jump

These can slow down the entire speed, reliability and efficiency.

Latency: time taken for data to travel from point to another.

Low latency: fast response

High latency:slow response

**Use a VPN** – Avoid ISP congestion by taking a different route.

✔ **Change DNS Settings** – Google (8.8.8.8) or Cloudflare (1.1.1.1).

✔ **Optimize Wi-Fi & Network** – Upgrade to **5 GHz Wi-Fi** or use **Ethernet** for critical tasks.

✔ **Monitor Traceroute & Speed Tests** – Identify where slowdowns occur.

✔ **Upgrade Network Hardware** – Old routers/switches may limit speeds.

✔ **Contact Your ISP** – If the bottleneck is outside your control.

Section 1.5

PROTOCOL LAYERS AND THEIR SERVICE MODELS.

* Service models have layered architecture embedded with protocols in each layer to control the sending and receiving of the data in an order.
* Each layer plays a specific role in transmitting the data by encapsulating them with headers.
* Why do we need layered architecture?

1. To smoothly deal with a complex system and ensure faster inter-operability.
2. Internet runs protocols to ensure that information is shared from source to destination.

* Modularity :breaking things into manageable parts

1. Each independent parts work together to achieve a larger goal.

* Protocol stack: the protocols of the various layers are known as protocol stack.

THE LAYERS:

7. Application layer: messages are generated at this layer. It allows user and software to communicate over a network. For ex: http protocol requests for web pages, SMTP for sending emails, IMAP is for receiving emails. For each application, there is a certain protocol

6. Session layer

5. Presentation layer

4. Transport layer: it transports the messages by segmenting the data into smaller packets. They are known as segments. Packets are included with port number.

* Transmission control protocol: it guarantees delivery of application layer messages to the destination.
* User datagram protocol: it is a connectionless service to its application.

3. Network layer: it provides an ip address to the datagram.

* IP address vs Port number
* IP : where to send (identifies the destination device)
* Port: who to send (identifies the application or service on that device)
* Each device has it’s own IP address.

1. Data link layer: data travels through a series of routers between source and the destination. So, link layer assigns a mac address to the packet to ensure that it is sent to the designated device. The assigned mac address could be either next hop device or the destination. If it is a hop device, then with each hop, the old mac address is removed and updated with the new mac address until it reaches the destination device.

* MAC: media access control: it works in the local area network and each device has its own mac address.
* The link layer packets are called frames

1. Physical layer: they are the hardware components through which the packets are sent.

ENCAPSULATION: the process of adding header when the data is moved through the layers, up or down of the service model.

1. Message - A
2. Segment - T
3. Datagram - N
4. Frame - L

HEADER FIELD AND PAYLAOD FIELD:

Payload field is typically from the layer above

Chapter 3

TRANSPORT LAYER:

1. Services
2. Transport and network layer services
3. UDP
4. TCP

Section 3.1

* The transport layer is responsible for establishing a connection between two entities.
* It converts the messages generated in application layer into segments(packets) by adding a port number identifying the application or service to send exactly
* TL has two protocols:

1. TCP: TRANSMISSION CONTROL PROTOCOL
2. UDP: USER DATAGRAM PROTOCOL

* TCP

1. It establishes a reliable connection between two entities. It is done through 3 way handshake.
2. Syn, syn-ack, ack. There after the connection between two entities is established
3. It guarantees delivery of the packets without loss.
4. It is reliable as even if the network layer loses or corrupts packets, tcp can re-transmit lost data.

* UDP:

1. It is connectionless, the protocol might lose the data. What matters is the delivery but not reliability. (the process might lose packets but the process will still continue)

Ex: buffering causes low quality video as some of the packets were lost yet the video continues, the same with the zoom calls.

* UDP is for fast delivery of packets, best for speed sensitive applications.
* Lets talk about the internet delivery system which is network layer protocol, it is known as IP address(the best effort delivery service), internet protocol that provides a logical communication between host systems. Yet IP does not guarantee segment delivery and orderly delivery.
* Every host has at least one network layer address i.e. IP address.
* The fundamental responsibility of TCP and UDP is to extend the IP address delivery service i.e. to deliver the packets in order and in segment. So they can be reassembled at the application layer.
* It extends the delivery service from host to host system to process to process system, by using port numbers.
* The speed is slow as it does error checking and re transmits the lost data packets.

KEY MECHANISMS:

* Flow control: each receiver has a window size that tells how much data it can handle at a time.
* The sender must not send more data than its limit, preventing the data loss due to buffer overflow. Avoiding glitches
* Sequence numbers: each byte in TCP ocnnection has a unique sequence number, the receiver uses this and reassemble out of order segments and detect lost packets.
* Ack: to confirm whether the receiver has successfully received the data. For each data received, the receiver sends acknowledgement back to the sender expecting next sequence number to arrive. If the sender does not receive ack in a given time period then it retransmits the data.
* Timers & re transmissions: they ensure ack is received within a specified time. Or it re transmits the data to avoid loss.

1. Re transmission timer:
2. Persist timer
3. Keep alive timer
4. Time wait timer

* Congestion control: it prevents excessive traffic from overwhelming the network, by controllilng the rate at which data is sent.
* Bandwidth sharing: ensures fair use of network.

MULTIPLEXING AND DEMULTIPLEXING.

Section 3.3

UDP: USER DATAGRAM PROTOCOL

* It facilitates a process to process communication. It is used where speed is prioritized over reliability. It provides a connectionless transport of packets as it does not perform handshake. it does not guarantee the delivery of packets. It does establish connections like TCP through handshake.

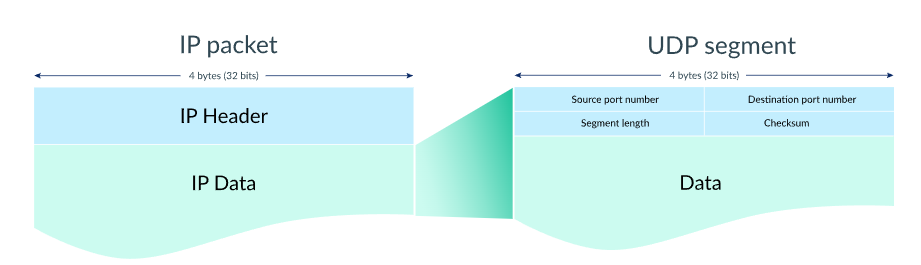
WORKFLOW:

1. A message is created at the application layer.
2. It adds source and destination port numbers.
3. Adds error checking field
4. Sent to the network layer, which puts it inside an IP packet and sends it to the receiver
5. The receiver extracts UDP data and delivers it to the correct application.

Examples: domain name system. It sends a quick UDP request to the server, if the request is timed out then it sends another UDP request instead of waiting.

UDP can tolerate some packet loss

UDP SEGMENT STRUCTURE:



* UDP uses ip to deliver its messages across the network.
* Each ip data has UDP segment in its data portion
* Udp segment consists a header and a data section, the header is fixed to 8 bytes but the data field can vary depending on what is being sent
* Udp segment has source, destination port number
* Segment length has the total length of the udp segment
* Checksum: used for error checking, if the data is corrupted then it is discarded
* IP address takes 4 bytes if it is IPv4 and 16bytes if it is IPv6

ERROR HANDLING IN UDP:

* It uses checksum as its header.
* The sender calculates the checksum and adds it to the UDP header.
* The receiver verifies the checksum for corrupted data
* If an error is found, the packet is simply discarded. There is not request for retransmission of the corrupted packets.
* Check sum is of 16 bit length.

How is a checksum calculated?

* It uses binary addition rule
* For example, consider 16 bit words of 3
* First two words are added together
* Then the sum is added to the next 3rd word
* If there is an overflow, the carry is considered for the next step of binary addition (wrap around)
* 1s complement : the 0s are flipped to 1s and the 1s are flipped to 0s

How is the checksum verified?

* First, the receiver gets the transmitted data(final sum sent by the sender)
* It extracts checksum from the final sum
* And performs binary addition
* If all are 1’s then data is transmitted without any error.

Section 3.5

TCP: TRANSMISSION CONTROL PROTOCOL.

* It is a connection oriented protocol providing reliable, ordered and error checked delivery of the data between the applications running on hosts communicating via an ip network.
* Why is TCP reliable?

1. It establishes connection between the client and server before the data exchange through three way handshake.
2. Three way handshake ensures that both client and server are synchronized for the data exchange.
3. The data can be sent and received at the same time.

TCP CONNECTION ESTABLISHMENT: THREE WAY HANDSHAKE.

* Synchronize: it sends SYN message to the server, requesting an establishment of a connection. It includes a sequence number to track data segments.
* SYN-ACN: it sends synchronize acknowledgement message back to the client, it ack the clients sequence number and includes its own sequence number.
* ACN: client sends an ACK message to confirm that the connection is established, and both devices can start sending and receiving data.

After establishing a connection, data transmission begins

HOW DATA IS SENT IN TCP.

1. The data is segmented into pieces, the segmented has its own sequence number. Ack num, checksum and other data control flags. It is now called as TCP segment.’
2. This tcp segment is then placed inside a IP packet sent over the network.
3. Then the IP packet is transmitted throughout the networks devices, they do not track tcp connections but only forwards the packets.
4. The receiving system extracts the tcp segment and re arranges it if the data segments are received in out of order.
5. Sends an acknowledgement to confirm successful reception

Flags: flags are control bits used to manage connections and manage transfer.

A little about urg flag:

* It is managed at the application level instead of tcp
* It transmits data in smaller size first as they do not need a larger bandwidth compared to a video playing in the background, which used tcp or udp for data transfer for sustained amount of time.
* FLOW CONTROL:

1. Tcp uses flow control to ensure the sender does not overwhelm the receiver.
2. Sliding window protocol: each window has a specific size (how much data it can accept at a time), every sender must wait for an ack before sending more data.
3. If the receiver is slow, the sender adjusts its speed to avoid its packet loss.

* Reception : the process of receiving, processing and acknowledging incoming tcp packets.

1. IP delivers the packets
2. Tcp segment is extracted
3. Verifies the data using checksum field, if the data is corrupted then it immediately requests for re transmission of the data.
4. Arranges out of order packets using sequence numbers.
5. Send an ack msg back to the sender to confirm successful reception

Notes: flow control and congestion control

1. If the network is slow, the data stored in the buffer decreases to avoid packet losses, the sender adjusts its speed according to the receivers capacity.
2. If the network is congested, then the rate of sending packets drops to avoid packet losses, it is based on the network traffic.

SOCKET: it is the gateway to establish a connection between two systems.

TCP SEGMENT STRUCTURE:

* It has two fields inside it, one is header and another is data field.
* Data field has the actual data being transferred.
* Sequence and ack number: it used for reliable data transfer
* Sequence number identifies the position of the data from the sender.
* Ack number: confirms reception from the sender

MSS: maximum segment size: it defines the largest amount of data a segment can carry. Large files are broken down into smaller chunks according to the segment size.

* If an syn is lost during the handshake, it does not receive syn-ack from the server.
* Then, the client re transmits the syn after a timeout(tcp re transmission mechanism)
* If multiple syns fail, then the client assumes the server is unreachable and stops after a timeout.

**TCP segment is inside an IP packet.**

✅ **MSS controls the data field size but is negotiated in the header.**

✅ **If a file is smaller than MSS, TCP sends it as it is.**

✅ **Ports ensure the right application receives the data.**

✅ **Acknowledgments confirm data was received.**

✅ **Reception happens on both sides.**

✅ **Sequence numbers ensure ordered data delivery.**

✅ **Flags control the connection and data flow.**

✅ **If SYN is lost, TCP retries.**

✅ **Urgent pointer is rarely used due to modern app-layer handling.**

* Each application has a system for managing traffic, which is known as quality of service. For example, whats app prioritizes messages over a video or a sticker being sent at the same time. When the network is congested or the internet is slow, messages are sent

First.

SEQUENCE NUMBERS AND ACKNOWLEDGEMENT NUMBERS:

* Tcp treats the data as unstructured but ordered stream of bytes.
* Tcp is full duplex - it can send and receive data at the same time.
* First segment has bytes from 0-999
* Second has 1000 to 1999
* It continues until 500 segments
* Ack numbers are used to identify the sequence of bytes it should receive for the ordered reassembly.
* For ex: host a received bytes of 0-535 then the next is 536 - 899, next is 900 - 1000
* If 536 is lost then, ack will send ack = 536 until it receives 536-899 bytes.
* ISN: initial sequence number. The new connection initial sequence starts with any random number to avoid any confusions with previous old connections.

1. It is established before exchanging data between two hosts.
2. The sequence numbers start with isn chose by the servers.
3. First, the sequence are used to establish the connection, after successful connection, the data is exchanged.
4. Sequence numbers track both control segments (syn, syn-ack, ack) and data segments.
5. Control segments has only the sequence numbers to perform the handshake.
6. Data segments has both sequence numbers and data.
7. Echo: echo in telnet is when the server sends back each character you type to confirm it was received.

* Telnet is an application layer protocol which you can use to connect to remote servers or hosts. It can be accessed using telnet session. It echoes back each character you typed.
* It has a disadvantage of encrypting the data making it vulnerable to eaves dropping so SSH(secure session layer) is preferred over telnet. The connection is closed using the control segment(FIN, where no data is exchanged)

ROUND TRIP TIME AND TIME OUT ESTIMATION

* RTT is known as the time taken for a segment to travel form sender to receiver and vice versa.
* How is it measured

1. Tcp starts timer
2. Segment sent
3. Ack received
4. Records the time taken.

* RTT should be calculated as to set a time out period for requesting re transmission of data.
* Estimated rtt

Chapter 4

THE NETWORK LAYER: DATA PLANE.

1. Data plane
2. Control plane

DATA PLANE

* It is responsible for host to host communication
* It operates in every host and router - determines how datagram is forwarded from one router input to an output link.

CONTROL PLANE:

* Controls how packets are routed between routers alond an end to end path.
* It uses routing algorithms to find out the best path to forward through routers between the sender and receiver.

SOFTWARE DEFINED NETWORKING.

Section 4.1

OVERVIEW OF THE NETWORK LAYER.

* It is a simple network where two hosts communicate with each other through multiple routers.
* Role is to ensure datagram travel fronm the sender to the receiver efficiently.

WORKFLOW:

1. Encapsulates the segment generated at the transport layer.
2. Sends it to the nearest router
3. At the receiver side, the NL extracts the transport layer segments and send them to the transport layer.
4. The TL then delivers the data to the application.

* Network layer encapsulates and forwards datagram while network layer is considered with handling end to end communication.
* Forwarding (data plane function): when a packet arrives, the router must move it to the correct output link
* Routing (control plane function): it determines an end to end path by using routing algorithms
* A forwarding table has next hop address, when a packet arrives at a router. It examines the packet header and looks up in the forwarding table and determines the correct output link.
* It is a local operation.
* Forwarding is done in hardware while routing is done in software.
* Forwarding also filters packets, blocking malicious packets and packet duplication(sending copies to multiple links)
* Who sets up the forwarding table?

1. Routing decides the best path for packets and creates a forwarding table based on routing algo, and forwarding moves the packets based on it.
2. It is a dynamic operation.
3. Automated routing protocols ensure fast, accurate and dynamic updates for forwarding tables.

* There are two approaches for forwarding packets. One is the traditional approach and another one is the software defined networking
* Traditional approach:

1. It is distributed approach where each routers make their own independent tables dynamically and exchange information with the neighbours
2. It is slow and uses distributed protocols such as OSPF, BGP, RIP.

* Software defined approach:

1. Where a centralized system makes a pre computed forwarding tables are forwarded to all the routers, updates are made by the centralized system.
2. It is fast and scalable for changing routing policies via software updates
3. Used in modern cloud data centers isp and 5g networks.
4. Used controller based protocols such as openflow, netconf, bgp ls.

NETWORK LAYER SERVICES.

* Different types of services are provided by the network layer, affecting reliability, speed and order of packet transmission.

1. Guaranteed delivery
2. Guaranteed delivery with bounded delay
3. In order packet delivery
4. Guaranteed minimal bandwidth
5. Security

* BEST EFFORT SERIVE
* ALTERNATIVE SEVICE MODEL

Section 4.3

IP (internet protocol)

* It is the fundamental component in the internets architecture. It is responsible for addressing, routing and delivery of packets between different devices.
* Workflow

1. Addressing - assigns an unique ip address
2. Packetization - data is broken down into packets before the transmission of data.
3. Routing- the best path to send data from source to destination
4. Fragmentation and reassembly - large packets are divided into smaller packets to fit the network constraints and later reassembled at the destination.

* IPv4 and IPv6 are the two version of the internet protocol, they define different ways of addressing and routing of data across networks.
* Subnetting is a ip address management technique used to divide larger IP network into smaller subnetworks.

IPv4 DATAGRAM FORMAT.

* It has a header and payload, it plays an important role in routing and delivering packets.
* Version number - 4 bits
* Header length - 4 bits
* It differentiates the type of traffic.
* Datagram length is - 16 bits.
* It ensures proper reassembly at the destination, after splitting the datagram into smaller fragments.
* Time to live
* Protocol - indicates the transport layer protocol that should be kept/ receiving the payload.
* Header checksum: detects errors, if error is found then it is discarded by the router
* Source and destination IP address - 32bits each.
* Options- allows the header to add extra information if needed.
* data(payload) - the main content of the datagram which is usually occupied by tcp or udp segment.

Notes:

1. Voice over internet protocol - this protocol allows us to make phone calls over the internet ex: whats app calls
2. File transfer protocol : used to upload and download files from a server.
3. Real time traffic - data must be delivered immediately
4. Non real time traffic - data does not need to be delivered instantly and can handle some delays
5. Time to live: a number assigned to the packet that controls how long it can stay in the network. The no. is decreased by 1 and when the packet reaches 0, the packet is dropped from the network.

ADDRESSING:

* It defines how devices are identified and communicate over the internet.
* Ip address is an unique identifier assigned to each device connected to a network. Ipv4 addresses are 32 bit numbers, written in dotted decimal notation. The format divides the address into four 8 bit segments called octets.
* Octets are crucial for determining the network and host portions of the address

Subnet: in a network multiple interfaces and one router forms a subnet. An ip address is assigned to this subnet, left most ip address will indicate the subnet it is connected to. Any other device connecting to the subnet will have the same form as assigned the subnet.

* Lets say you have 3 routers and each router is connected to group of computers, now it has a total of 6 subnets, 3 routers connected different local computer and router to router connecton.

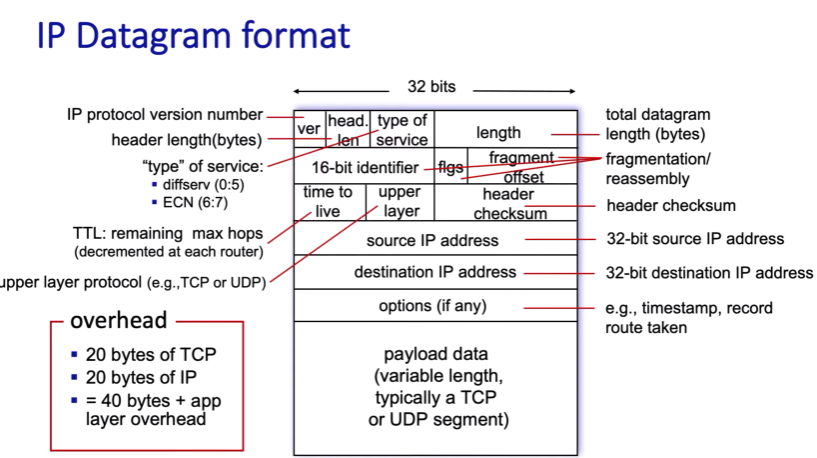
CLASSLESS INTER DOMAIN ROUTING: it is an internet address assigment strategy.

* It allocates bits in an ip address by dividing them into two parts

1. Network portion: prefix bits (identifies the network)
2. Host portion: suffix bits (identifies devices within the network
3. Ex: 192.168.1.0/24 represents that 24 bits is allocated to the network portion and the remaining 8 bits are for the host portion
4. It allows a variable length subnet masks to efficiently allocated ip address unlike classful addressing, it has a fixed size of network bits and wastes ip addresses, classful addressing cannot create custom subnet masks.
5. Cidr has a subnetting flexibility where multiple small networks are combined into one entry.

Ip protocol

* Datagram format
* Addressing
* Packet handling conventions



How does a host gets it IP address.

1. It uses DHCP(dynamic host configuration protocol) it dynamically gets address from the router
2. DHCP server is used to allocate ip addresses to the incoming hosts.
3. If a dhcp server is not in the subnet then, it has a dhcp relay agent that knows the address of a dhcp server for that network is needed.

WORKFLOW:

* DHCP server discovery - the incoming host does not have an ip address assigned, so it send a dhcp server discovery message
* Dhcp server offer - once the dhcp receives it, as the host does not have an ip address, so several dhcp servers in the subnet responds and send dhcp server offer.
* Dhcp request - the new client chooses from one of the many offers.
* Dhcp ack - the server responds request with an ack. Confirming the requested parameters.

NAT: NETWORK ADDRESS TRANSLATION.

* It allows multiple devices in a private network to share a single public ip address
* Changes the source ip and port and sends the request, it translated back when a response is received.
* Hides internal ip address but does not encrypt data

IPv6:

* It used 128 bit system making it limitless, addressing the spacing issue in ipv4.
* It introduced a new types of addressing known as anycast addressing, which allow a packet to be delivered to the nearest host among a group of hosts.
* It has 40 byte header.
* By removing fragmentation, header cheksums and options, it significantly reduces the router workload leading to faster packet forwarding.

Section 4.4

GENERALIZED FORWARDING AND SDN:

Generalized forwarding:

* Traditional destination based forwarding involves two key steps

1. Match - look up the destination ip address in a forwarding table
2. Action - forward the packet to the corresponding output port.

* Instead of simply forwarding the packets based on destination ips, generalized forwarding allows packets to be processed based on multiple header fields across different protocol layers.
* Sdn enables generalized forwarding. It implements the match plus action using flow tables in network switches.

SOFTWARE DEFINED NETWORKING ;

* Normally, network has its own built in system (control plane) that decides where to send the packets and the data plane that actually forwards the data.
* Sdn separated the control plane from the data plane
* Control plane is centralised in SDN controller, where the best routes are constantly updated.
* Openflow is a popular sdn protocol.