what is the Internet? (recipes)

↳ There are some basic hardware and software components that make Internet.

→ Hosts ↴
PC, server, routers, wireless devices.

↳ Links ↴
All the lines or links connecting to the hosts / Copperwire, Coaxial cable, Copperwire, radio, satellite...

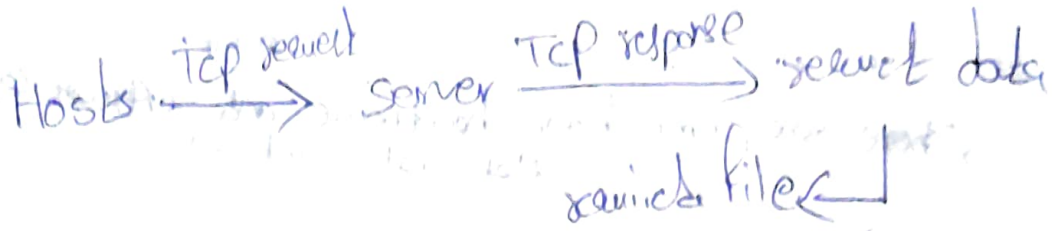
↳ Routes ↴
makes path for data, etc.
exits in the core of network.

↳ protocols ↴
Implements by the hosts and useful in many ways to communicate, & data transfer.
- etc

↳ Internet provides communication services eg: web, games, email, file share ---

↳ Best way of transferring data delivery. Ex: TCP is reliable data delivery protocol, IP, HTTP ---

↳ what is protocol? → rules for interaction between the hosts.
↳ format of messages
↳ order of messages
↳ Actions of messages.
↳ send and receive messages.



1.2

↳ Network edge interface.

↳ The network devices that are accessible to end-users. ex: Applications, skype, webbrowsers phones, laptops, Desktops etc---

1.3 ↳ Network core.

↳ is all routers and backbone tier-1 networks that are doing transfers between different networks

↳ Endsystems are .. nothing but hosts. that are accessible to their own.

↳ Two models

↳ client/server model. ~~provides server~~.

The server that provides resources @ services for the client.

Ex: web → web servers

chrome → Apache.

↳ peer-to-peer model.

Everybody is a client and a server.

Anybody can talk to anybody.

Ex: Skype, Bit torrent etc---

↳ Access Networks and physical media.
↳ the network that is physically connects end system and router
↳ How to connect end systems to router ?

Ans: By Access networks like residential access network. Ex: cable, fibre etc--

↳ Ex: modems, 3g, 4g, college internet access etc :--

↳ Wireless Access networks
↳ wireless LAN's → 11/54/108 Mbps
↳ wider area → 3G, 4G, LTE / 10's Mbps

↳ Two Medias
↳ Guided Media → Wired.
Ex: Twisted pair (TP) (LAN)
Coaxial cable. (T.V)
fiber optic cable (light)

↳ un Guided media → wireless.
Ex: Terrestrial Microwave
LAN (wiFi)
Wide-Area (cellular)
Satellite.

Network Core.

↳ How is data transferred through the net?

Ans: ↳ circuit switching.

A there is a seperate circuit for every connection b/w A and B (sender and reciver)

↳ packet switching.

All the data is chunked and divided into these discrete chunks of data and those chunks of data are Independent of each other, we send each that data seperetly along each router.

↳ End to End → all the resources are reserved so that they can quickly transferred.

Resonation → reserving the speed ahead of time (continously)

↳ Two simple access techniques that are used in circuit switching.

↳ FDM and TDM

FDM → Frequency division multiplexing

TDM → Time division multiplexing

↳ getting frequency from particular channels
Ex: Radio

↳ getting frequency from particular channel for particular time.
Ex: phone calls

packet switching

Advantages

→ No bandwidth Division.
→ No dedicated Allocation
→ No Reservation

disadvantages

↳ Resource contention
↳ Congestion (buffering)
↳ store and forward.

↳ packet switching is also called as statistical multiplexing because if A and B are two hosts sending packets to other clients then if A host sends packets with 100 mb/s and B host sends less than 100 mb/s then there will be A hosts packets that will be more.

↳ packet switching is a store and forward

$$\text{Transmission Delay} = N\frac{L}{R}$$

L → No of bits
R → speed of links
N → no of links

* Network of Networks.

↳ All types of hosts connects into the internet via an access ISP (Internet service provider)

↳ They can provide by DSL, cable FTTH, wi-Fi, and cellular.

↳ To connect billions of hosts to make up Internet the access ISPs themselves must be inter connected. this is called network of net

(14)

How do loss and delay occur?

↳ Transmission delay occur while router puts the packets in the line or air.

↳ queueing delay occur while queuing the packets in the line. (waiting)

↳ loss occur when the queue is full then router will drop the packets.

↳ propagation delay is the short amount of time that will take across the medium for the signal.

↳ processing delay the time that will take to find the address that packets will go.

Nodal delay

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

↳ if average queuing delay Increase then traffic Intensity Increases.

↳ what if buffer of queue is full?

(Ans) packets will drop.

↳ what if packet is lost?

(Ans) packets will be retransmitted by the previous node. or original source in host.

Throughput is the rate (bits/time unit) at which bits transferred between sender/receiver

Instantaneous → in given point of time.

Average → long period of.

why layers ?

↳ If we layer the layers then it will be well defined, seperate and Neat interfaces, Exchange information, Hiding the Details, Similar functions to know themselfs.

↳ Internet protocol stack. @ ISO/OSI

→ Application → Implimating network Applications
→ presentation (encryption) } Intermediate layer
→ section (syn)
→ transport → procasing the data transfer TCP / UDP.
→ Network → routing data. / IP
→ link → one node to next node. error control / mac address
→ physical → bits conversion.

OSI → open system Interconnection

~~ISO~~ :

TCP → Transmission control protocol

UDP → User datagram protocol.

↳ End to End → Client, Server

↳ point to point → switch, Routers
Client, server.

switch → link, physical
router → network, link
physical

↳ Application layer
→ End to End. layer
→ Implimenting network Application
→ user services like, web, Email, file transfer.

↳ Trasport layer
→ end to end layer
→ deletes the duplicate data
→ loses data will be resent
→ reorder the data.

↳ network layer
→ point to point layer.
→ IP Addressing the destination data
→ Routing the data the " "
→ provides Administration functionality.

↳ Link layer.
→ point to point layer.
→ Error control.
→ flow control
→ MAC (Medium Access control layer)
↳ transfer data fairly.

↳ physical layer
→ point to point byte
→ Transmission of Raw Bits.

↳ Encapsulation means adding a Additional information to a packet so that the packet can be used by the next layer. (section layer)

* datagram is a packet at network layer.
* segment is a packet at transport layer.
* frame at link layer.

Malware

then :

- Virus
- Worm
- Trojan Horse
- Spyware
- Botnet
- Denial of services Attack.
  - By putting fake addresses that increase the traffic in the network it will get you banned.
- packet Sniffing.
  - wired attack.
- Ip spoofing.
  - pretending to be some one's addresses.
- Record and playback attack.
  - Recording the packets.

↳ Network topology is the arrangement of the various components of a computer network.

1) Bus.
Data transmission in this topology is a single cable.
↳ cheaper.
↳ used in small networks.
↳ only in one direction.
Ex:- LAN.

2) Ring.
↳ the hosts are connected in the ring form.
↳ cheap to install and expand
↳ Unidirectional.

3) Star.
All the hosts are connected to a single host/hub with single cable in star shape.
↳ Fast performance and low network traffic.

④ Mesh topology :-

↳ point to point .. Connection to other hosts @ fully connected

↳ not flexible and costly.

↳ more cables used.

↳ more security.

⑤ Tree topology :-

↳ It has a root node and all other nodes are connected to it forming a hierarchy.

↳ Wide Area Network - WAN.

↳ easy to detect error.

↳ more costly.

⑥ Hybrid Topology

↳ A network structure @ contains more than one topology is said to be hybrid topology.

↳ flexible, error detecting.

2.1

Network Apps

↳ e-mail, web, instant messaging, remote login, P2P file sharing, multi user network games, streaming, social networks video conferencing, grid computing...

↳ How to create a network app?

→ No need to write program

↳ No need to add code to Network Core Devices

↳ Application layer protocols works on end-to-end layer

↳ Client server Architecture.

↳ Google Data centers

↳ IP address are pominent

↳ P2P Architecture

↳ No server

↳ Direct communication

↳ Intermittent connection

↳ IP address may change.

↳ Difficult to manage

↳ Highly scalable.

* Hybrid of C-S and P2P

Ex: Skype.

Socket:

Socket is programming interface for communication between two processes. (endpoint)

Transport layer is a layer that transports data that comes out from socket to the other receiving process.

* How can we identify a process.

↳ tasklist (windows)

↳ ps -aux or top (Mac/Linux)

↳ Identifier → IP Address, port number

To identify the process.

↳ Application layer protocal Defines.

↳ Message Type is a general type of message written by users.

↳ Message syntax
- subcomponents of message.

↳ Message ~~syntax~~ semantics.
meaning of packets.

↳ Message Rules.
when and how processes respond to the messages.

protocols

public domain
- ↳ HTTP.
- ↳ FTP
- ↳ SMTP
- ↳ Bit Torrent

proprietary
- ↳ Skype
- ↳ ppstream.

↳ Application layer provides communication between two processes.

↳ Transport services
  ↳ recover Data loss
  ↳ low delay for Transporting packets
  ↳ bandwidth should be more
      Ex: youtube ; streaming.
  ↳ security.

↳ Internet Transport protocols
  ↳ TCP
    ↳ Connection reserved.
    ↳ Flow control
    ↳ entry of datapackets to network
    ↳ reliable service
    ↳ Congestion control
  ↳ UDP is opposite to TCP

* http is a hyper text transfer protocal.

↳ Non persistent
  ↳ one object per TCP connection
↳ persistent
  ↳ Multiple object per TCP connection

* In Non persistent HTTP
  ↳ 2RTT's /object.
  ↳ overhead in Terms of TCP connection

* In persistent HTTP
  ↳ Reuse open Connection
  ↳ send ~~requests~~ requests immediately

↳ Electronic Mail.
  ↳ User Agents are used
  ↳ Mail servers ~~are~~ stores mails and sends mails to user Agents.
  ↳ SMTP (simple Mail Transfer protocol) is used to transfer the mails between Mail servers.

↳ SMTP

↳ It will use TCP to reliable data transfer.

↳ port is 25

↳ SMTP allowes the messages between sending server and recieving server.

↳ There are 3 phases

↳ Handshaking (greets the server)

↳ transfers b/w servers.

↳ cyclic.

↳ 7-bit ASCII

↳ persistent connections

↳ HTTP

Command/Response

ASCII

8-bit

pull from server

single part message

SMTP

Command/Response.

ASCII

7-bit

push to server

multi part message.

↳ pop (post office protocol) and IMAP (Internet mail Access protocol) are older protocols used to pull the data from server.

pop3	IMAP
Download and delete	server folders
Download and keep	

↳ DNS → Domain Name system.

* How do we map ip addresses and names?
  ↳ DNS is to map between names and ipaddresses.

↳ Distributed everywhere.
↳ Hierarchal distribution.
↳ It is one type of database.
↳ port is 53.

* why DNS is Distributed?

↳ To decrease the traffic volume.

↳ It could not be disbayed.

↳ easy maintenance.

↳ one server goes down that there will be no problem for othe servers.

↳ DNS : Features.

↳ DNS will return the IP Address to that Hostname.

↳ DNS supports common names. Ex: www.co.netflix.com, netflix.

↳ DNS gives unique ip addresses and gives different ones to different ones.

↳ Load distribution.

TLD → Top level Domain
Ex: com, org, net, edu.

↳ DNS Records is a database story
Name, value, type, TTL (Time to live)

↳ Record Types:

Type A → Name → Hostname.
Value → IP Address.

Type NS → Name → Domain.
Value → Hostname of domain server.

Type cname → Alias name.
↳ value → canonical name.

Type MX → Name → Hostname.
Value → name of mail server.

3.1

Transport layer is providing logical communication between processes running on different hosts.
↳ end to end protocol.
↳ TCP and UDP protocols are used.

↳ Transport layer is nothing but
· Collects and Distribute to
the users.

↳ Multiplexing is about taking
data and pushing on to one channel.

↳ Demultiplexing is about to take
data and pulling on to one channel.

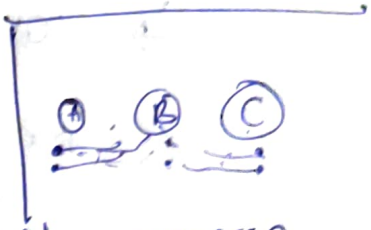
↳ port number is used to fix the
destination on the other host to get
the data.

↳ In transport layer of demultiplexing
↳ IP datagrams · Stores Source
IP Address, Destination IP Address.
and segments
↳ segment will store source port
and destination port.

Demultiplexing :

↳ UDP identifies the 2 things

1) dest → ipaddress
2) dest → port



↳ IP UDP segment - If the source port & ipaddress are same and if the Dest port & ipaddress are different.

↳ In UDP socket consists of dest ipaddress and port number as if to UDP - segment have different same ipaddress and port number. but have same dest ipaddress and port number. then two segment will be directed the same destination socket.

↳ port numbers are arranged in 0 - 1023

Ex: port 80 → web.

FTP → 21

↳ This is how demultiplexing happens in UDP.

↳ TCP socket identified by 4 things

- ↳ Source IP Address
- ↳ Source port
- ↳ Dest IP Address
- ↳ Dest port.

↳ ~~Different source IP address and port card.~~

↳ TCP connection pairs are unique.

* why is UDP there?

1) No Delay
2) Simple & cheap
3) Less wasted Bandwidth
4) No speed limits.
5) Lost data
6) inorder.

Ex: ↳ streaming multimedia.
↳ because Loss Tolerant
Rate sensitive.

↳ DNS
↳ SNMP

↳ Length → segment + header in bytes.

↳ ~~checksum is the number that is computed based on data in the packet~~

↳ The number is used to check whether the packet is matched to the packet that is received.

↳ How checksum works?

checksum is a simple error detection mechanism to determine the integrity of data transmitted over a network.

↳ Add up all the integers or by 1's complement.

↳ And receiver will also do same.

↳ If the computed sum is not equal to stored sum then there must be a corrupted bit. Then Error detected.

↳ All zero's will be No Error.

↳ why checksum is at transport level?

↳ Link layer has no gaurantee that reliable

↳ It can be corrected. In router if we don't use checksum in transport layer.

↳ End to End principle.

Communication protocol operations occur at the end points of communication system.

* principle of Reliable Data transfer

↳ transport layer has to implimet the protocal to create a reliable channel.

↳ rdt_send() is the function. that does reliable data transfer. by reliable data transfer protocol.

↳ After sending it will divided into packets; (rdt_send() and that and sent to data over unreliable channel. and rdt_rcv will recive the data from unreliable channel and deliver to the deliver_data() function.

↳ How these Function calls will be implemented?

↳ Finite state machines (FSM)

↳ when these events and action happed then we will move from 0 state to next state.

↳ we Incrementally develop the Rdt.

↳ rdt 1.0 is the first version of the protocol ↳ this is perfectly Reliable over Reliable channel.

↳ No Bit Errors

↳ No packet Loss

↳ on sender side. (rdt_send)

↳ wait for call from Application layer and it will make the packet.

packet = make_pkt.

and it will send the packet

udt_send.

↳ on receiver sid (rdt_rcv)

↳ wait for call from the sender. and it will extract the packet and deliver_data.

* rdt2.0

↳ Bits Errors.

↳ No packet loss

↳ How to detect errors?

↳ checksum.

↳ How to recover from errors?

↳ By receiver Feedback.

↳ There are Two types:
  ↳ Acknowledgements: Acks
  ↳ Negative Acknowledgements: NAKs

↳ Then Retransmission takes place.

↳ what happens if Ack/NAk is Corrupted?
  ↳ sequence number to the packet.

↳ Rdt2.1
  ↳ Implements sequence numbe to solve Ack/NAk corrupted files.
    ↳ sequence is added to packet
      ↳ 2 sequence (0, 1).

↳ rdt 2.2.

↳ Ack's only. (same as rdt 2.1)

↳ rdt 3.0

↳ Bit Errors

↳ packet loss

↳ Error Detection

↳ checksum, Acks seq #, retransmission

↳ How to detect packet loss?

↳ wait for "Reasonable" time period.

↳ retransmit the packet if no Ack received.

↳ stop-and-wait operation

$t = RTT + L/R$.

↳ Here sender waiting most of the time to get Ack for that pipeline protocol is used.

1000 bytes

$\frac{8000}{1000} = 8$ 23

$\frac{8000}{R} = \frac{}{1,000000}$

$\frac{8000}{1000000}$ $\frac{8000 \times 1000}{8 \times 1000}$

↳ pipelined protocols.

↳ Go - Back - N

↳ selective Repeat

sender
↳ up to N packets in pipeline

Receiver
↳ sends cumulative Acks
↳ If there is a gap then it won't Ack.

sender
↳ set the time
↳ if timer expires then retransmits.

sender
↳ up to N packets in pipeline

Receiver
↳ Individual packs of Acks

sender:
↳ Timer for unAcked.
↳ if time exp then retransmits.

↳ TCP reliable data transfer.

↳ TCP creates rdt service on top of IP's unreliable service.

↳ goes in pipelined segments

↳ cumulative Acks

↳ retransmission time.

↳ when retransmissions occur

↳ timeout event

↳ duplicate Acks

↳ what TCP does in this scenario:

↳ ignore flow control

↳ ignore congestion control

↳ ignore duplicate Acks.

If Ack test from first packet and directly sent second Ack then it will assume first ack's also successful

↳ ~~if the packet lost then the receiver~~

↳ sender does not wait for timeout.
It will wait for triple Ack's and
It will ~~send~~ the lost packeet is
retransmit
called fast retransmit

↳ point to point
↳ connection oriented
↳ reliable data ; in-order
↳ Congestion and flow control.

↳ TCP seq number is the number of the
first ~~byp~~ byte in the segment.

↳ how to set TCP timeout value?
↳ RTT estimation
↳ too short → retransmission waste.
↳ too long → segment loss..
time waste.

↳ how to estimate RTT?

↳ By sample RTT.

EstimatedRTT = (1-α) * EstimatedRTT

+ α * sampleRTT

here α = 0.125

↳ Timeout interval = EstimatedRTT + 4 * DevRTT

Here DevRTT = (1-β) * DevRTT +

β * |sampleRTT - EstimatedRTT|

here β = 0.25

* Approaches to Congestion control.

↳ End to End help to control.

↳ Routers.

↳ By ATM ABR control.

* ↳ How TCP 'conbol' congestion.

↳ end to End. Connection oriented.

↳ decreases the send rate.

*↳ How to perceive Congestion?

↳ Feedback mechanism Implimentation

↳ Acks

↳ Acks not Received.

* ↳ How to limit send Rate?

↳ limity the unAcked bytes in "pipeline".

cwnd = LastByteSent - LastByteAcked

sendRate = $\frac{cwnd}{RTT}$

↳ when sending Rate increases and Acks being received and By that packet will loss so we must decrease sending rate.

↳ First we have to start slowly and then we Double Cwnd every RTT It is called slow start threshold.

4.1

↳ Transport segment from sending to receiving host

↳ sender encapsulates segments into the datagrams.

↳ network layer protocols in every host, router and server.

↳ two network layer function:

↳ forwarding

↳ routing.

↳ Forwarding → moving packets from router's input to router output.

↳ Routing → finding the route taken by packets from source to dest.

↳ routing algorithm determines route to route path through network.

↳ forwarding table determines local forwarding at this router.

↳ Datagram networks.

↳ packets forwarded using destination host address.

↳ source ip address

↳ dest ip address.

↳ Datagram forwarding table consists of dest ip address and route

↳ how routers are deciding ipaddress and store them?

↳ There will be address ranges for every interface.

↳ routers the packets will be forwarded using destination address and longest address prefix.

↳ First 21 bits will match and last bits will decide the ipaddress

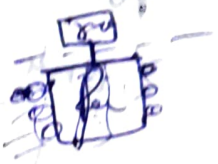
↳ If the two ipaddress are same then longest address prefix will decide where should it go.

↳ Router architecture

- ↳ Input ports → datagrams Enter
- ↳ output ports → datagrams exit.
- ↳ routing determines the routing path By routing algorithms like (RIP, OSPF, BGP)
- ↳ forwards datagrams from input to output ports By high-speed switching fabric.

↳ routing and forwarding data plane is logical seperater.

↳ routing processor will compute the routes and tells high speed switching fabric how to forward the packets from Input to output port.



↳ In Input port

↳ line termination (physical layer)

↳ link layer protocol: (data link)

↳ queueing

↳ forwarding table works

↳ if line speed is speed and forwarding speed is slow then packets will drop.

↳ switching fabric. transfer the packet from input port to output port.

↳ Three type of switching fabrics

↳ memory ↳ bus ↳ crossbar.

4 memory switching

↳ traditional method

↳ packets copied to system memory and then moves to output port.

↳ limited by memory bandwidth

↳ bus switching

↳ by shared bus. it will buffer.

↳ limited by bus bandwidth.

↳ crossbar switching

↳ fragments datagram into fixed length cells and switches through fabric

↳ limited by crossbar bandwidth

↳ output ports

↳ Buffering when datagrams arrive from fabric.

↳ queuing if packets loss occur

↳ Head of the line (HOL) blocking.

↳ what work will done in between routing and forwarding?

↳ routing Algorithm will decide the node. to move from which node to output link.

↳ Forwarding table determines local forwarding at this router.

~~* Routing algorithm classification:~~

↳ IP datagram format.

↳ consists of ↳ IP protocol version
↳ header length
↳ time to live.
↳ datagram length
↳ checksum
↳ source IP address
↳ dest IP address.

↳ If the datagram is large then the IP datagram is divided into several datagrams and they will reassemble only at final destination. (MTU)

↳ IP address is a 32 bit identifier and it is stored in two 8 bits.

↳ Interphase is a connector between host and physical link.

↳ routers have multiple interfaces

↳ hosts have one or two interfaces.

↳ maximum value of 8 bits is 223.

↳ Subnets

↳ Isolated network is called subnet

↳ detaching interfaces and creating the islands.

CIDR → classless Inter Domain Routing

ip address format is a.b.c.d/x

↳ x is subnet portion.

Ex:

11001000.00010111.00010000.00000000

subnet port host port

200.23.16.0/23.

↳ here 23 is the bits in subnet portion

↳ how does host get ip address?

↳ By DHCP.

↳ Dynamic Host configuration protocol.

↳ IPv4 and IPv6 are ip versions.

↳ The length of IPv4 is 32 bit.

↳ In IPv_4 has 4 billion unique ipaddresses available.

↳ The range of IPv_4 addrs is 0 to 225
Ex:- 192.225.108.253

↳ It consists of 4 octets each has 8 bits.

↳ octets are separated by dot.

↳ 5 classes

· IPV6 :

↳ Th length of IPV6 is 128 bit

↳ In IPV6 has 340 trillion unique ipaddresses available.

↳ The range of IPV6 address is 0 to
1=FFF (65535)
Ex:- 2001:DB8:1904:FB83:1800:175C:CBA:0000

↳ It consists of 8 octets each
has 16 bits

↳ octets are separated by (:) Colon.

↳ No classes.

DHCP

↳ It allows host to dynamically obtain its IP addrs from network.

↳ allow reuse of addresses.

↳ can renew.

↳ support for mobile users

↳ DHCP client server.

↳ DHCP discover

↳ DHCP offer

↳ DHCP request

↳ DHCP ACK.

↳ name and addrs of DNS server.

↳ indicate network mask.

↳ How does network get subnet part of ipaddr?

A: gets allocated portion of its provider's ISP's address space.