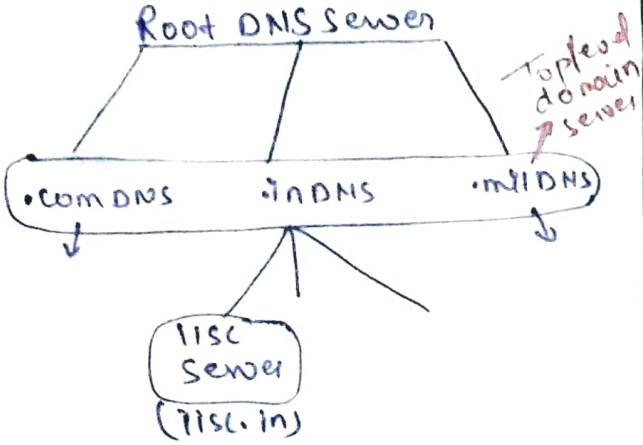
→ DNS have record for each server service arrange in form of tree.

→ Arrange of DNS server is like the distributed database-



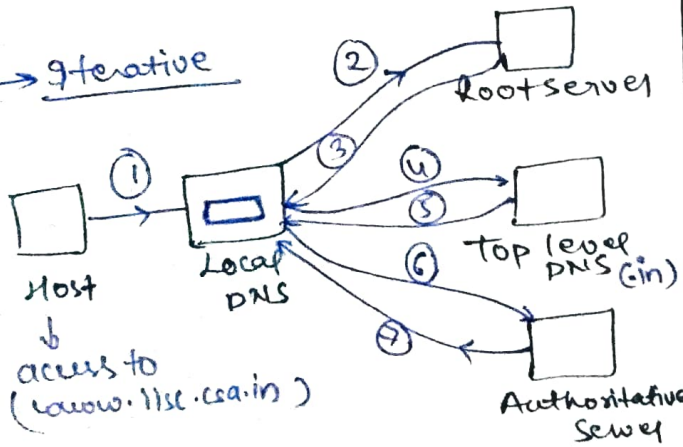
→ We have lot of root servers so, that if any of root servers gets down, then we can get access from another one.

→ India's nearest root server in Tokyo (Japan).

→ Each ISP provide local DNS server to use.

→ If entry is not in DNS then DNS will get the data from above server called as DNS overhead.

→ Iterative

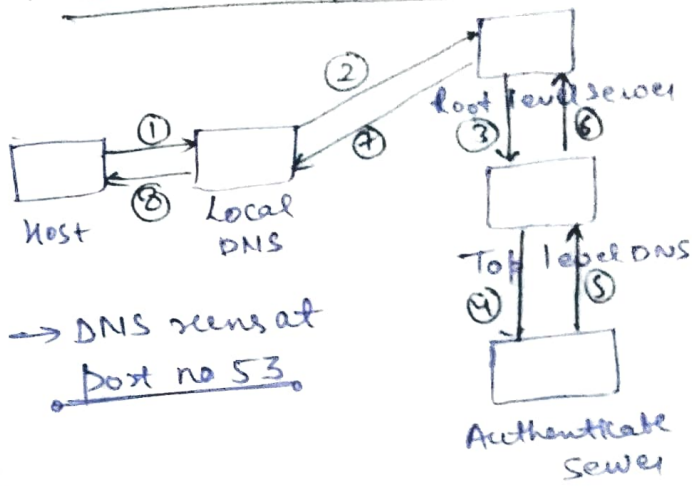


→ If entry is not is table and Local DNS ask to Root server it replies that entry is not in root server contact to top level DNS.

Similarly if Top level DNS doesn't have any info it will responde and say to contact with authoritative server.

And finally it will get the IP Address of required domain name.

→ Recursive Approach



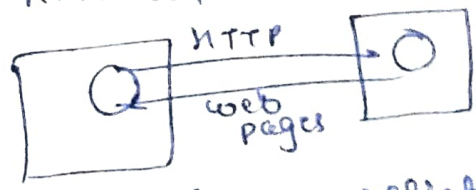
→ DNS runs at port no 53.

→ Increase local DNS cache at max as possible to reduce traffic.

→ DNS uses UDP at TL.

HTTP : Mainly used for getting web pages.

→ Run at port no-80



→ Always relies on reliability from the layers below AL.

→ It uses TCP at transport layer.

→ It is 'in band' protocol means both command and data sends in one connection

→ HTTP is stateless (not going to maintain info.)

→ Cookies always saved as client side.

→ HTTP 1.0, uses non-persistent connection. (Ravi sir Module)

→ HTTP 1.1 uses persistent connection. (Gmail)

→ In non-persistent server have no need to establish conn^n for long time.

→ Bandwidth is very High in persistence conn^n.

Methods [Read from Internet to get logic of them]

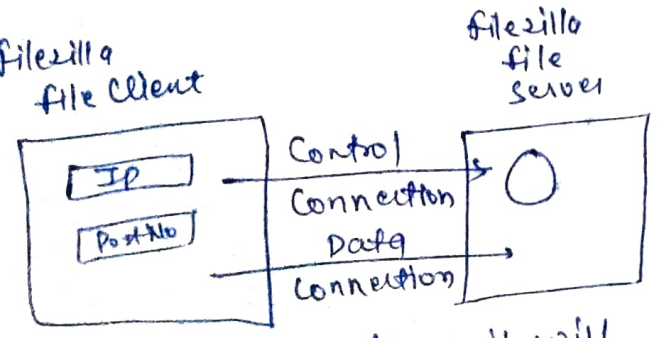
① Head → To get header of webpage or metadata of webpage.

IETF (Internet Engine Task force)

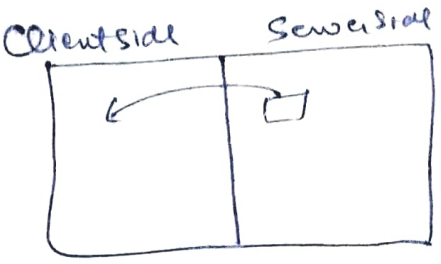
② Get
③ Post
④ Put
⑤ Delete
⑥ Trace
⑦ options
⑧ Connect.

FTP : File Transfer Protocols

Ex → Tectia, Filezilla (open source)



After above procedure it will create a window as-



→ You can just drag and drop files from server to client.

→ Data Connection is non-persistent but control connection is persistent.

→ FTP requires reliability from other layers, so FTP uses TCP.

→ FTP is stateful protocol

→ Uses
Control Conn^n : Port no. 21
Data Conn^n : Port no. 20

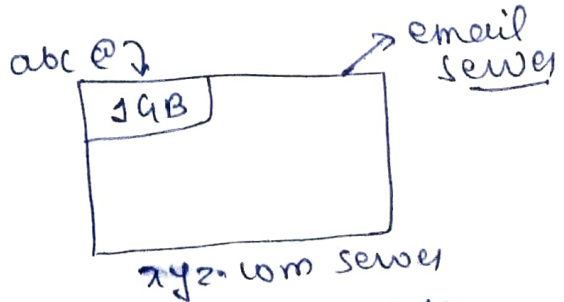
→ It follows "out of band".

SMTP and POP:

Email transfer through SMTP (Simple mail tranfer Protocol) and POP (Post office Protocol)

→ In FTP both client & server should be online at the same time but in SMTP it is not necessary to online every time.

→ abc @ xyz.com

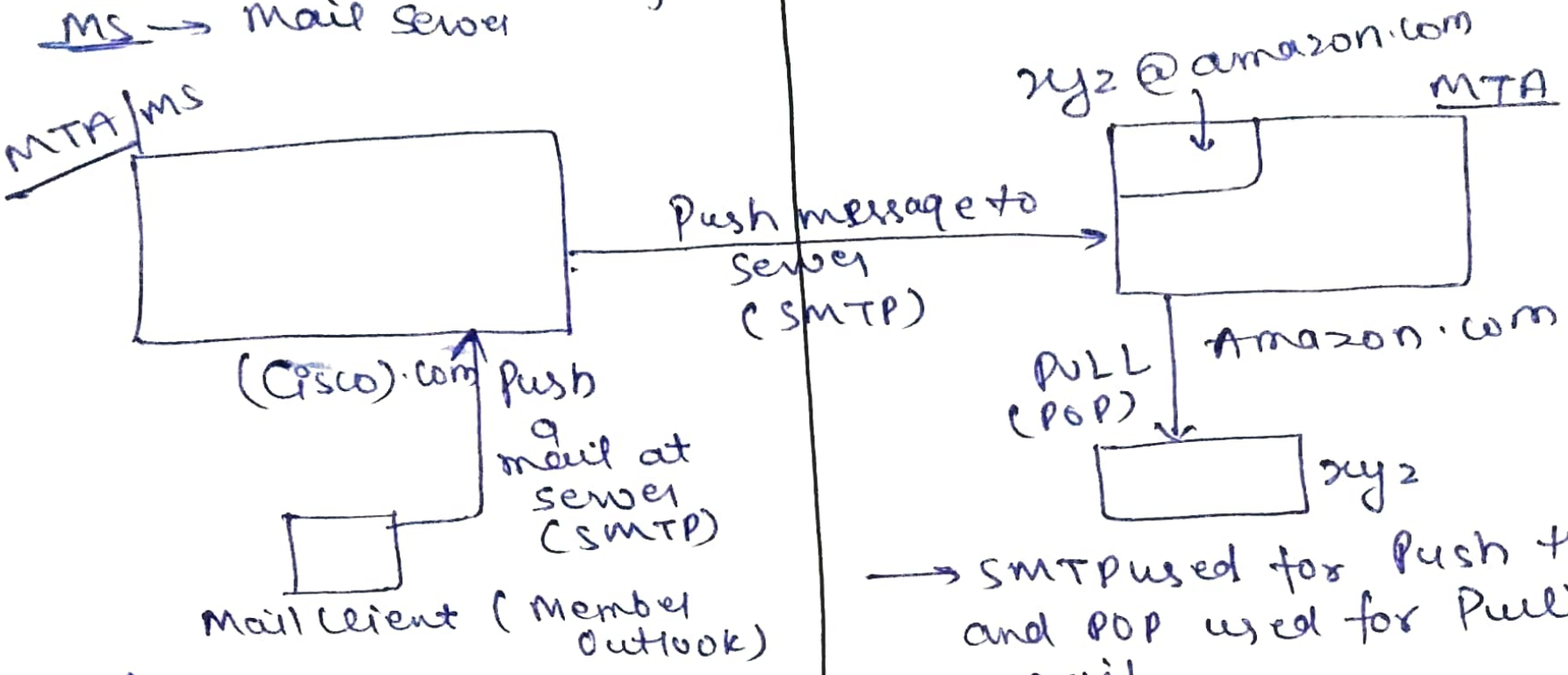


1GB space alloted to abc in xyz.com server.

→ hotmail was first email service for general public.

MTA → Mail Transfer Agent

MS → Mail Server



→ SMTP used for Push the mail and POP used for Pulling the mail.

Convert

Non-Text to Text
Text to Non-Text

using → (MIME) Multipurpose Internet mail extension.

→ Gmail converts the same process easyly to good looking web pages. Background process is same but gmail form a web page and makes the thing user friendly.

→ So gmail is webbased email.

→ IP address from domain name using DNS is an overhead.
→ for web services by default port no is 80.
→ IP address has prefic code for different classes as 0(A), 10(B), 110(C), 1110(D), 1111(E)
→ Total no. of IP addresses possible 2^{31}(A), 2^{30}(B), 2^{29}(C), 2^{28}(D & E).
→ Class A :- (1 - 126) : No of N/w :- 2^7 : No. of host :- $2^{24} - 2$
→ But we can't use all 2^7, we uses $2^7 - 2$ (126 N/w only).
→ Class B :- (128 - 191) : No of N/w :- 2^{14} : No of Host :- $2^{16} - 2$
→ Class C :- (192 - 223) : No of N/w : 2^{21} : No of host :- $2^8 - 2$

Broadcasting → Limited : Same network : 255.255.255.255
→ Directed : Different N/w : NID.255.255.255

→ In subnet mask NID & SID takes 1's and HID takes 0's
→ More the network size less 1's in subnet mask. (more no. of host present)
→ Each routing table have default entry as 0.0.0.0.
→ With subnet mask we can find only no. of host, to find no. of subnets we need class and subnet mask both.
→ CIDR: All IP should be continous, block size in 2^n, first address should be divisible by 2.
→ We can find VLSM in both CIDR and classfull addressing.
→ for supernetting size of all networks (hosts) should be same.
→ Private IP : 10.0.0.0 → 10.255.255.255, 172.16.0.0 → 172.16.255.255
192.168.0.0 → 192.168.255.255.
→ In case of data 1k = 1024 but in Bandwidth 1k = 10^3.
→ $T_p = \frac{d}{v}$, $T_t = \frac{L}{B}$ ⇒ OR $\frac{L}{\text{Total Time}}$

Stop & Wait : $\eta = \frac{1}{1+2a}$, Throughput = $\eta * B$, RTT = $2 * T_p$

→ If probability of packet lost is p then = $N\left(\frac{1}{1-p}\right)$
packet will transmito

→ Capacity = $2 * T_p * B$ [Full Duplex]

To acheive max. (η) : window size = $(1+2a)$ = seq. no.

→ (Avail. seq. No.) $\geq W_S + W_R$.

→ In Go-Back N : (N) packet will retransmit Cumulative alk.

	S & W	GBN	S/R
η	$1/1+2a$	$N/1+2a$	$N/1+2a$
Size	$W_S = 1$, $W_R = 1$	$W_S = N$, $W_R = 1$	$W_S = N$, $W_R = N$
Retransmission	1	N	1
Bandwidth	Low	High	Moderate

Access Control ① TDM : $\eta = \frac{1}{1+a} = \frac{T_t}{T_t + T_p}$

② Polling $= \frac{T_t}{T_t + T_p + T_{polling}}$

③ CSMA/CD : $T_t \geq 2 * T_p$: $L \geq 2 * T_p * B_W$

$\eta = \frac{1}{1+6.44a}$

④ Back off Algo. (CSMA/CD) : The packet get transmit first then that sender have high chance to send the packet again [Capture effect].

Token passing : (time)$_{sec}$ = (time)$_{bit}$ / Bandwidth

(time)$_{sec}$ = (time)$_{metre}$ / velocity

Ring latency = $\frac{(\frac{d}{v})}{sec} + \frac{(N * b)}{bits}$

$\eta = \frac{NT_t}{T_p + N * (T_{HT})}$ N → No of stations in ring.

→ In Delayed token reinsertion : TRT = $(T_t + T_p)$ (66)

→ In early token reinsertion : TRT = T_t.

So, $(\eta)_{ETR} > (\eta)_{DTR}$

Aloha → Slotted : Time division (T_b) : $\eta = G * e^{-G} = 36.8$

→ Pure : Max collision : $\eta = G * e^{-2G} = 18.4\%$

CRC : If CRC generator is k bits then add (k-1) 0's in the data and do (XOR). XOR always done with the no. followed by MSB as 1.

Checksum : Devide no. in chunks of (8, 16, 32..) and find its decimal values and add them.

Physical Layer → Bit rate control and Bit synchronization. → Transmission Cable (Duplex), topologies, encoding, wire or wireless, point to point connection.

DLL → framing, MAC address, error/ flow / access control. we add SFD (1 to 0)*. (node to node) (LLC, MAC)

→ Bit and character stuffing: In character stuffing we add '10' and in bit stuffing we add 0 to break sequence.

→ NL → host to host, router table, switching, fragmentation, Congestion control, IP addressing.

→ TL → End to End connectivity, Segmentation. error control

IP are connectionless. So no acknowledgements

Session, Pre. L, Appl :→ Authenticity, Checkpoint, dialogue control, logical grouping, encryption, compression.

Ethernet : (IEEE 802.3), Bus topology, CSMA/CD, connectionless, manchester encoding.

Unicast : One host to other : LSB is 0 in 1st byte.

Multicast : one host to many : LSB is 1 in 1st byte.

Broadcast : one to all : All 1's in all bytes.

Max data = (1500 B), min data = 46 B
frame = (1518 B), min fram = 64 B.
(DA, SA) MAC = 12 B, Length = 2 B, CRC = 4 B

→ Simple but not applicable for real time & client server application (no priority). Backoff Algo is used.

→ Switching: In ckt switching NO header is required. Packet Switch is good for small data transfer. Header may or may not required. Packet size should be less but greater than size of header and to reduce (T_t) we can use pipelining method.

Virtual Ckt: 1st packet global header, all other has local header, voice call, max routers, costly, connection oriented, highly reliable.

Datagram: No global header, no buffers, connection less, all packets have different header, less, reliable, cheap, message transfer, data calls.

IPV4: Min. Header = 20 B, max header = 20 + 40 = 60 B

1st row 4B: ① Version (4b): ($v_1, v_2, v_3, v_4, v_5, v_6$)
② HL (4b): header length, ③ Total length (16 b)
④ Type of service (8b). (Discussed in TCP after URG flag)

2nd row 4B: ① Identification (16b) ② Fragment offset: (13b)
③ Donot fragment, more fragment, 0: (3 bits)

3rd row 4B: TTL (8b), Protocol (8b), header checksum (16 b)

4th: Source IP (32b)

5th row: Destination IP (32b)

(5th): Optional (0 → 40)

→ To identify HL we devide HL by 4 and keep it inside HL field. If not in multiple of 4 do padding.

→ Identification no is used to identify the datagram uniquely.

→ fragment offset keeps info of no. of fragments ahead.

→ TTL decreases as it passes one station or router. It is useful in case of looping path or infinite loop.

→ Protocol: TL (TCP/UDP), NL (ICMP, IGMP).
ICMP < IGMP < UDP < TCP

→ Header checksum changes at each station bcoz of change in TTL value.

→ Options: ① Record Route: It can record at max 10 IP address (4×10=40B) but if you separate each IP by some bit as max 9 can be hold. It store the IP of each router in path.

② Source Routing → Give the Routing path in header. Also called strict source routing.

→ Segmentation (TL) occur at host sides but fragmentation can be done at each routers.

→ In fragmentation ID remains same but (offset and MF) bits gets changed.

→ If MF=0 means, it is end fragments.

→ If $(MF)_{parent} = 1$ then all child have (MF)=1, but If $(MF)_p = 0$ then all child will have (MF=1) except last child with $(MF)_c = 0$.

→ Do fragmentation and set fragment offset value accordingly.

→ In value of fragment offset we use scaling factor method.

→ If packet is middle then we can add padd bits to make it multiple of 8 or if packet is last one then no issue.

→ If packet is multiple of 8 then child will also multiple of 8 if parent is followed by some more packets.

→ Due to fragmentation more header bits are required to $(\eta)_{new} = \frac{\text{Useful data}}{\text{Useful data} + [(\text{Header bits}) \times \text{no of headers}]}$

→ In case of different (B_W) routers path we choose $\min(\eta * B_W)_0$ (Bottom neck)$_0$.

→ Reassembly of fragments (Algo):

first packet:	$(MF=1)\ (FO=0)$
Middle packet:	$MF=1\ FO\neq 0$
Last —— :	$(MF=0\ FO\neq 0)$

MF = FO = 0 means no fragmentation →

Protocols (NL): Broadcasting is a concept at OLL

If Limited Broadcast then router see the packet and doesn't allow to go forward.

If directed broadcast then router will send the packet to corresponding routers.

→ Network is limited to the router.

→ ARP (IP → MAC): Address resolution Protocol

You want to send packet to $(IP)_x$ but you don't know about the (mac) router address of that network.

→ ARP request is broadcast but $(IP)_x$ reply is unicast

IB	?	MA	255.255.255.255.255.255

→ Special address: 127.—.—.— is used to check Internet is working or not. It is also called loop back address.

→ RARP (MAC → IP): Mac address present in ROM of NIC so it is permanent.

→ In RARP we have a mapping table contains mac address correspond to IP Address.

Ma	0.0.0.0	MR	Broadcast

↳ Unicast

We get IP from the router of your own network.

→ Mapping table is static (Disadvantage)
→ We need mapping table at each routers (Disadv.)

Bootp: Same as RARP but runs on application severs

~~Mapping~~ table is static but only one RARP severs

DHCP: (Dynamic Host Configuration Protocol).

→ Dynamic mapping for private IP and static mapping for public IP's. (Very popular).

ICMP [Internet Control message Protocol].

Part 1: (Error Handling or feedback msg)

Part 2: (Request and reply).

To avoid infinite loop we come with a concept that ICMP request can't be generate on discarded ICMP packet.

Part 1: ① TTL Exceed : Packet doesn't received and in btw TTL=0

② Source Quench: If receiver buffer size < sender buffer then receiver will say to quench sender (stop).

③ Parameter problem: Checksum, route problem in case of strict routing.

④ Destination Unreachable: Host Unreachable, Port Unreachable
→ Host is down
→ DF = 1 (can't fragment)

⑤ Source redirect: ICMP is warning here. In case of strict routing if we have better path than choosen path then intermediate router generate warning that we have a better path.

⑥ Requirement of sender: To discard any packet and send back we need sender's info.

To discard any ICMP we create another ICMP packet.

⑦ Icmp never generate for Icmp packet.

⑧ In case of fragmentation, Icmp generate for 1st packet.

⑨ Icmp Generated for both TCP / UDP.

ICMP (Part B):

① **Echo Req. & Reply**: To verify N/w layer of destination & routers are working or not.
Ex: ping. [Packet Internet Gopher]

② **Router Solicitation**: Broadcast an ICMP request to get info. about nearby routers

③ **Router Advertising**: Router generates an ICMP that I am new here and available to use

④ **Time stamp**: To synchronize clocks in different time zone. Not in use and highly unreliable

⑤ In traceroute we identify the route to destination so it's different from record route

→ To check packet reaches destination or not we use dummy port no. because after that destination sends packet with host not found.

→ If IP packet get lost then TTL will become 0 in between the path & we can get ICMP in return.

→ **Path MTU Discovery**: To get min(MTU) in the path. So that we can avoid fragmentation.

→ Send packet with DF=1 and you will get info of MTU of particular router in return.

Routing

→ Send to particular router using routing tables

→ No. duplicate packet] Adv.
→ Not much traffic
→ Multicasting service

→ Routing table wasted space.] Disadv.
→ Not much reliable if link is failed

Flooding

→ Send to all routers

→ No routing table] Adv.
→ Shortest path guaranteed
→ Highly reliable in case if any router is down

→ Max traffic] Disadv.
→ Packet Duplication is possible

Routing Algo.

Static: Manual work, updation is tough if link is down, can't change if traffic or topology changes.

Dynamic: No manual work, done by router automatically, changes dynamically according to traffic & topology. [DVR, LSR]

DVR: Distance Vector Routing

→ Edge wt can be due to distance, traffic or queing delay.
→ Every routing table has 3 entries [destination, distance, next node]
→ If nodes are not in direct contact then, distance = ∞.
→ A node can take Distance Vector (DV) from only in direct contact nodes.
⇝ this arrow means indirect contact in graph
At A if direct contact at B and D then find AB & AD first then find all paths using DV.
In graph with n-vertices we will go for (n-1) rounds.
→ In every round we don't use update value of DV during computation of DV for another router.

Drawbacks: ① Count to Infinity:

① If new network added is a good news and it spread fast.
② If a network gets detached out, is a bad news and it spreads slow.

Split Horizon: Soln of count to infinity problem.
→ Other drawbacks are presence of loops and Convergence is slow.

LSR (Link-state Routing): Each router has link state packet instead of routing table in the begining.
→ In LSR every node floods the information.
→ DVR based on local knowledge but LSR based on global knowledge.
→ After getting distance information we use Dijkstra.

→ It converges faster than DVR.

Disadv.: Heavy traffic due to flooding.

→ We maintain info. of latest packet to discard old pack.

→ To avoid ∞ loop we use TTL.

→ To avoid errors we use concept of lifetime.

Transient Problem ① Blackhole problem: If link is down all packet come to Blackhole.

→ Loop is transient problem here but persistent prob. in DVR (short time) (long time)

DVR: Low Bw, local know., Bellman ford, less traffic, periodic updates, slow convergence, count to infinity, persistent loop, (RIP) [Route Info. Protocol]

LSR: High Bw, global know., Dijkstra, more traffic, periodic updates, converge fast, no count to infinity, transient loop, (OSPF) [Open shortest path first]

RIP and OSPF are implemented version of both.

→ In RIP we consider 16 instead of ∞ in DVR. (UDP)

→ LSR have complex computations. Devide routers in regions and restrict the flooding to particular region in OSPF.

→ EIGRP is hybrid of both RIP and OSPF.

Tcp Header

1st row: Source IP(16), Destination IP(16)

2nd row: Sequence no. (32)

3rd row: Acknowledgement no. (32)

4th row: HL(4), Reserved (6), URG | Ack | PSH | RST | SYN | FIN (1), window size (16).

5th row: Checksum (16), URG pointer (16)

6th row: optional [0→40B]

Min Header length = 20, Max. Header leng. = 60

→ Tcp is end to end protocol and able to do MUX & DEMUX.

Port no. (0-1023): well known (1024 - 49151) → Reserved
(49152 - 65535) → Public.

- TCP is connection-oriented and apply at TL.
- Complete address is (IP + Port) → socket (48); Unique
- TCP is Byte stream protocol. IP is packet stream
- seq. no (of Byte), Ack no. (last Byte seq no. + 1)
- TCP uses random initial sequence no. to avoid collision.
- WAT depends on Bw and range of seq. no.
- seq. no. will not create collision problem till WAT > Lifetime
- Bw ↑ ⇔ WAT ↓
- To avoid collision either ↓ Bw or increase seq no size
  Decreasing Bw is not desirable but we increase seq no. size using optional feild bits.
- Min. size of seq. no = (Bw × Lifetime of packet) (If Lifetime is given else WAT = $2^{32} \times 8$ / Bw)
  Extra bits : N − 32
- To represent HL we use modulo 4 concept
- Acknowledge no: TL − (Ip header size) − (TCP header size)

Flags: Each SYN = 1 consumes 1 seq. no. Pure Acknowledgement (Ack = 1) consume acknowledge no. but not seq. no.

- FIN = 1 consumes 1 sequence no
- SYN flag used for connection establishment only
- Ack flag used in connection establisment, data transfer and connection termination.
- We use FIN flag during connection termination.
- Connection start and close both done by client only.
- We have timeout and ack. timers at both sides
- If no packet to send then send pure Acks

SYN	Ack	
0	0	→ Not possible
0	1	→ Ack no in packet
1	0	→ Request
1	1	→ Reply

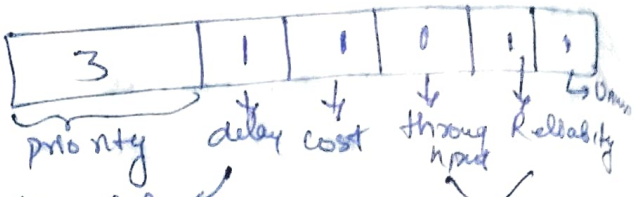
- After closing connection, client can send pure acknowledgement only. But server can send data as well as piggback acks
- 3 packets are required for connection establishment and 4 for termination, but we can terminate it in 3 also.

→ Push flag is used if data to send is very small to achieve good efficiency.

→ Urgent flag: Used to move prior instruction to execute.

→ Priority (URG=1) is max upto 7 and set at N/w layer.

→ type of services (IPv4)



Delay: choose path with less delay

Cost: less cost

→ URG Pointer: It shows upto this point data is urgent to send.

→ RST flag: It is used to terminate the connection in abnormal condition.

FIN vs RST

FIN: desired termination, one side connection abrupt, no data loss, receiver is allow to communicate

RST: abrupt termination, both side stop, data loss, receiver also stop the connection.

Window Size: Used to manage flow control. If at sender side persistent time passes then sender sends 1B packet to check WR is free or not.

→ If WR is greater than $(2^{16}-1)$ then we can use some bits of optional feild too.

Checksum: Compute on (IP header + TCP header + TCP data)

Take 5 feilds of IP (SIP, DIP, Protocol, 8 bit 0's, TCP segment length) called as pseudo header.

Options: (1) Timestamp: useful if (WAT < LT)

(2) Window size extension.

(3) Parameter negotiation

(4) Padding in case of header length.

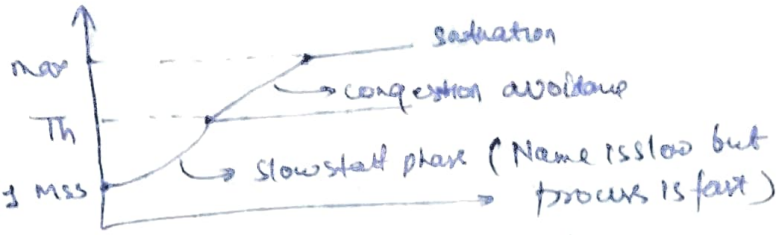
Retransmission in TCP: Uses both SR ($W_S = W_R$) and GBN (Cummulative Ack.)

If one packet get lost all the transmit pack will send the seq. no. of lost packet as ack.

- We need congestion control bcoz, we might make flow control by chosing min (W_S, W_R) but if network buffer is less than (W_R) then, we need congestion control.

→ Threshold = $\frac{W_R}{2}$

→ Sometimes we start with 1 MSS.



→ upto Th power of 2, then linear and saturate after W_R.

Congestion avoidance Algo: ① Slow start phase ② Cong. avoidance phase ③ Cong. detection phase → Time out (Congestion is severe) → 3 duplicate Acks (congestion is mild)

ICMP can also be used for congestion control.

→ If congestion due to timeout then $(Th)_{new} = \frac{W_C}{2}$ and enters in slow start phase.

→ If congestion due to 3DA then $(Th)_{new} = \frac{W_C}{2}$ and enters in congestion avoidance phase.

W_C → current window size.

TCP Timers:

① **Time wait Timers:** After connection termination don't release all resources, wait for (2 * Lifttime) for any delayed packets.

② **Keep-Alive Timer:** Close idle connection (connections which are not sending the data) only. If any connection (not idle) not sending data then wait for (KAT), after this send a probe msg if no reply from client then server closes the connection.

③ Acknowledgement timer

④ **Persistent timer:** Used for periodic probing. If packet needs to send at regular period of time. [Time btw Ack. recieved & packet send]

⑤ **Timeout timer:** RTT = 2 * T_p, TO = 2 * RTT (static)

But static TO is not useful in TCP bcaus there can be any no. of routers in between.

To needed to be changed according to traffic

Algo. Used ① Basic Algo.: $(NRTT) = \alpha(IRTT) + (1-\alpha)ARTT$

ARTT → Actual RTT , IRTT → Initial assume RTT

NRTT → Next packet RTT.

α → Smoothing factor. If $\alpha=0$ NRTT = ARTT

$\alpha=1$ NRTT = IRTT

So here, TO = 2*IRTT (D is advantage) why 2 only?

Jacobson's: We need IRTT & ID (deviation) both.

$TO = 4 * ID + RTT$ — (1)

Given, (IRTT & ID), (ARTT) then $AD = |IRTT - ARTT|$

$NRTT = \alpha(IRTT) + (1-\alpha)ARTT$

$ND = \alpha(ID) + (1-\alpha)AD$

Each time calculate new (NRTT & ND) and used in TO (eq.1)

Karn's modification: If Ack. doesn't come in TO time then next time just double the TO times duration.

Silly window Syndrome: ① If sender is sending packet but W_R is full. so receiver is discarding continuously is a persistent problem.

③ Receiver is consuming only 1 bit at a time. So sender can send only 1 B at a time is another problem.

Nagles Algo: Soln for 2nd problem: Don't send 1 B. Collect all bits is buffer and send after 1 RTT if packet gets full in between then you can send it.

Clark: Soln for 3rd problem; Receiver can't advertise 1 until he had to wait to wait till ($W_S = \frac{1}{2}$ buffer)

for 1st problem use RST flag and restart it again.

2nd problem: If send is sending only 1 bit.

Traffic-shaping: It decide the rate of packet sent to. During connection establishment N/w and sender negotiate at a traffic pattern.

① **Leaky Bucket:** Input and Output is fixed. Constant rate of sending a packets

② **Token Bucket:** Max. no. of packets = $(C + rt)$

Max. avg rate = $\frac{(C + rt)}{t}$ /sec (Out rate)

C → capacity of bucket in tokens
r → rate of coming tokens.
t → time intervals or burst time

UDP:- UDP is useful where TCP is overhead as-

① 1 Req / 1 Reply [Chat]: DNS, BootP & DHCP, NTP/time & News protocol

② Broadcasting and multicasting.

③ Application required speed over reliability. (Games).

UDP is used whenever you need not require all-facilities of TCP.

UDP Header

SP(16)	DP(16)
Length(16)	Checksum(16)

Length = UDP (Header + Data)

→ RIP, ~~OSPF~~, TFTP (Trivial) also uses UDP, DNS.

Internet Protocols: (Appln layer) ① **DNS:** NSlookup: command used to get IP address.

→ DNS keeps record in form of tree in distributed manner.
→ If entry is not in local server you will get it from global server.
Uses UDP at TL. Increase local DNS Cache to avoid traffic.

HTTP: Port no 80, uses TCP at TL, stateless, relies on reliability of layer below AL. (in-band).
Cookies saved at client side. HTTP 1.0 (non-persistent), HTTP (1.1) persistent. Bw is very high is persistent.

Methods used: Head, Get, Post, Put, Delete, Trace, Connect.

FTP: Client make 2 type of connection with server.
① Control (20) (persistent) ② Data connection (21) (Non-persistent)
→ Uses TCP and its stateful.
→ It is (out of band) means both control and data send in different connections

Stateless: HTTP, UDP, DNS
Stateful: TCP, FTP, Telnet, POP3, HTTPS

SMTP (Simple mail transfer Protocol)

Unlike FTP client / server don't need to be online at the same time.

MTA → mail transfer agent

MS → Mail server

SMTP used to Push the mail and POP used to pull out the mail.

MIME → Non text to text or vice versa

→ Max. data rate possible is Throughput.

→ Routing (NL), Bit Sync. (PL),

→ MAC sublayer of DL has 3 types of protocol [Random, Controlled and channelized].

→ Using CRC we can detect only odd no. of errors. if (x) is not a factor of g(x), but (1+x) must be factor of g(x).

Always take care of unit of the answer.

→ Spanning tree algo. consist of 3 mechanism.
① Frame forward ② Add. Learning ③ Loop resolution

→ IEEE 802.11 is used for Lan than uses CSMA/CA for access control.
↳ collision avoidance

Spanning tree protocol: (802.1D) created to prevent loop, using probe message of switch receives its own probe msg back, it means loop.

→ works on the concept of root bridge. All other switches finds a way to reach root bridge.

Cables: 10 Base 5 ! means Range of LAN is 500m with Bw of 10Mbps.

→ PL, collision occurs, to increase LAN range we join two or more cables using regenerators.

→ Repeater regenerates signal and used at PL.

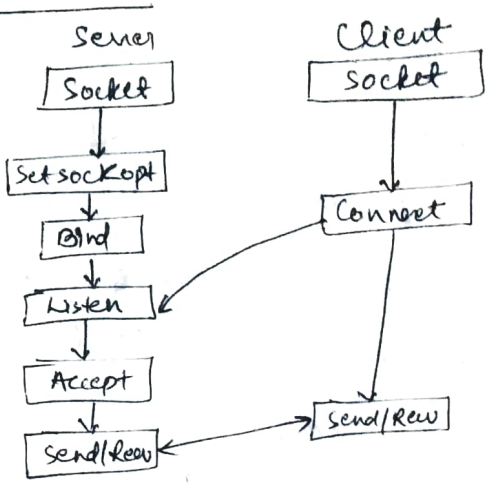
HUB: It is a multiport repeater, high traffic, PL, cheaper, collision occurs, we can do broadcast also.

Bridges: PL & DLL both, forward the msg, have a bridge table, store and forward packet, no collision inside a bridge.

Switch: Host and switch connects, NL & DLL, both m/m & processor, no collision, less traffic, costly, full duplex cables.

→ Repeater and HUB have same Broadcast & Collision domains.
→ Switch and Bridges have same broadcast domain but CD gets reduces.
→ If each host has separate Tslot and 'n' host are present with probability of sending 'p' and only 1 host can transmit at a time then, $\boxed{\text{throughput} = (p)(1-p)^{n-1}}$
→ In ethernet interface $\boxed{T_t \geq 2 * T_p}$
→ FTP and POP3 are stateful AL protocols
→ In case if IP address got verified by 2 subnet masks then go for subnet mask with more no. of 1's.
→ Both TCP and UDP uses datagrams
→ ICMP generated in TCP/UDP both the cases.
→ AL $\xrightarrow{\text{anysize}}$ TL $\xrightarrow{2^{16}-1}$ NL $\xrightarrow{2^{16}-(1+20)}$ DL $\xrightarrow{1500B}$ PL.
→ TCP handles both congestion and flow control but UDP doesn't
→ Slow start mechanism deals with both congestion and flow control.
→ Fast transmit deals with congestion but not flow control.
→ An UDP socket can't connect or communicate with multiple peers simultaneously.
→ A UDP socket can call connect() for same connection for two reasons –
  (1) To specify new IP or port address
  (2) To disconnect socket connection
→ In TCP $WAT = \frac{2^{32} \times 8}{Bw(\text{in bps})} = \frac{2^{32}}{Bw(\text{in BPS})}$

Socket fenctions in TCP :



→ In reverselookup IPaddress → domainname (Host).

→ FTP [PORT, PROMPT] SMTP [RCPT, HELO]

→ fermat's Little Theorem : $a^{P-1} = 1 \pmod P$ where P is a prime no. and 'a' is not divisible by P.

→ PPP is a protocol at DLL and BGP and NL.

→ Non-persistent vs Peristent HTTP :→ In Peristent HTTP we can send multiple objects in just one connection, but in non-persistent we need separate connection for every objects.

→ Error control is mandatory for TL. Presentation layer is responsible for compression and decompression.

→ Datagram fragment during routing (if required) and reassembled at destination.

→ Datagram at source need not be of smallest MTU size of whole network.

→ At DLL trailer usually contains bits used for error detection.

→ Traceroute doesn't always gives right path to destination.

→ Link-State : Intra domain or interior protocol not inter domain.

→ max file size that we can transmit over a network is 2^{32}

for next state according to inputs.

CN Imp:

- To detect 'd' bit errors in a n-bit number, min hamming distance should be d+1 and to detect and correct min hamming distance = (2d+1).
- To obtain the hamming distance do ex-or of code given.
- Both router and bridges selectively forward the packets
  A bridge (2 layer) uses MAC Addr and router (3 layer) uses IP.
  A bridge build its routing table by inspecting incoming packets.
  A router can operate between a LAN and a WAN.
- TCP doesn't guaranteed min. Communication rate but ensures in-order delivery, reacts to congestion by reducing sender's, employs retransmission for compensation of packet losses.
- In congestion control if value is given then slow Th $\left(\frac{cwr}{2}\right)$ follow slow start upto 'Th' and then congestion avoidance upto cwr. If any thing from slow start or congestion control is given then follow that phase from beginning to end.

Ex: cwr = 64KB, MSS 1KB, Th = 32 MSS (Ending is given)

1 2 4 8 16 32 33 34 (timeout Th = 17) 1 2 4 8 16 17 18 19 20 (3DA (Th = 10))

10 11 12 13.

- To solve question of propagation delay from a PT correctly.
- Connect ACK is same as Initial SEQ, accepted at the time of connection establishment.
- In Broadcast each all the host bit should be 1, and in NID all host are zeros.

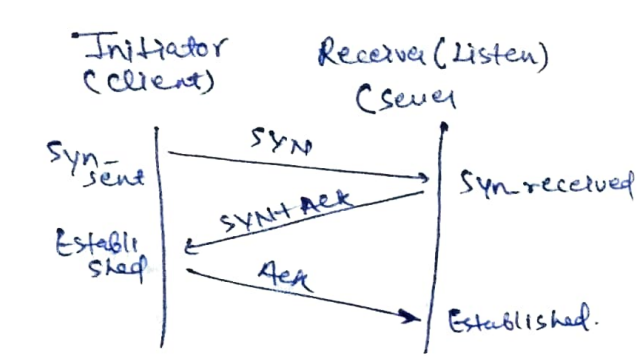
→ Connect() system call used to send SYN Packet to

→ AL can send any amount of data to TL.

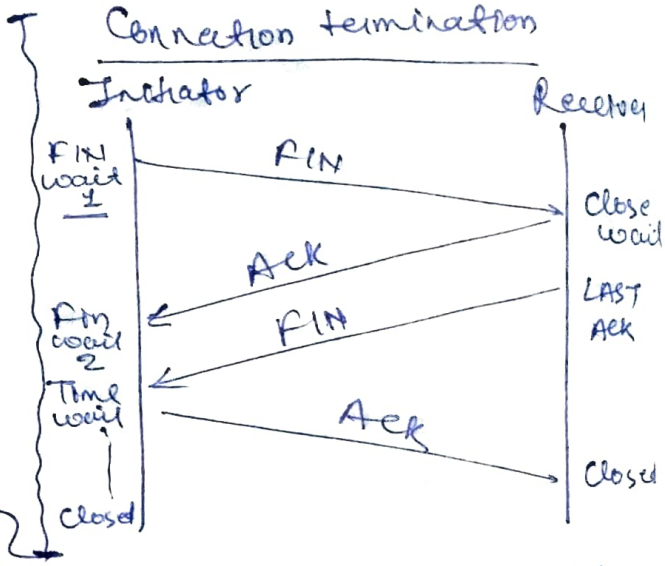
→ TCP have both and UDP have none of congestion or flow control mechanisms. -fast transmit and recovery is part of TCP. Slow start mechanism deals with both congestion and flow control

→ If in option info. given give (Tr) of Acknowledgment Include it

Connection establish



Connection termination



Bind(): After creation of socket, it binds Socket to address and port no.

Listen(): It puts server in passive mode, and server waits till response from client

Accept(): Acceptance of first connection request and established it

Connect(): Connects the socket referred by the file descriptor

⇒ 2, 4, 6, 7, 8, 16 After 5 RTT congestion ws will be 8 and advertise $co_s = 16$.

→ AL and TL both are different (take care).

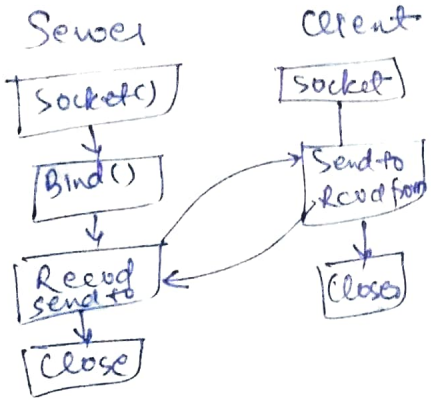
→ In congestion control take care of MSS during addition

→ Remember the concept packet switching with pipeling.

→ In UDP, TTL, Checksum and FO all can changed before reaching destination.

Cont 9mp:

Socket fn in UDP:



UDP Header

SPort 16 bits	DPort 16 bits
Length 16 bits	Checksum 16 bits

→ Datagram length = (Data + HL)

→ Value of retransmission timeout ≥ estimated RTT

→ If Bw = x bps then RTT = $\frac{(2^{16}-1)\times 8}{x}$ or $\frac{(2^{16}-1)}{x}$

if x in Bps. In TCP max $w_s = (2^{16}-1)$ Bytes.

→ TCP has both selective and cummulative acks and full duplex links

→ In N/w 200.10.11.144/27 the fourth octet of last IP add. assigh to host is 200.10.11.[100][11110] → 158
NID — no change in this. — not 11111

→ OSPF uses both TCP & UDP.

→ The congestion window double every RTT.

→ Packet switching has better utilization of Bw, less variation in delay and lead to reordering but doesn't require per packet processing as compare to circuit switching.

→ Prompt: FTP, Head: HTTP, RCPT: SMTP.
PORT: FTP HELO: SMTP

→ FTP needs 2 port on server side. One for data (20), one for control (21).

→ HTTP is not a state sensitive protocol.

→ IMAP can operate with more than one server.

CN Points:

SMTP → AL, BGP → NL, TCP → TL, PPP → DL

→ Send mail: SMTP, received mail: POP, check email in web browser: HTTP.

→ UDP used in real time applications

→ The computational overhead is more in LSR.

→ HTTP and FTP can use multiple TCP connections between same client and servers

→ Completed