INDEX

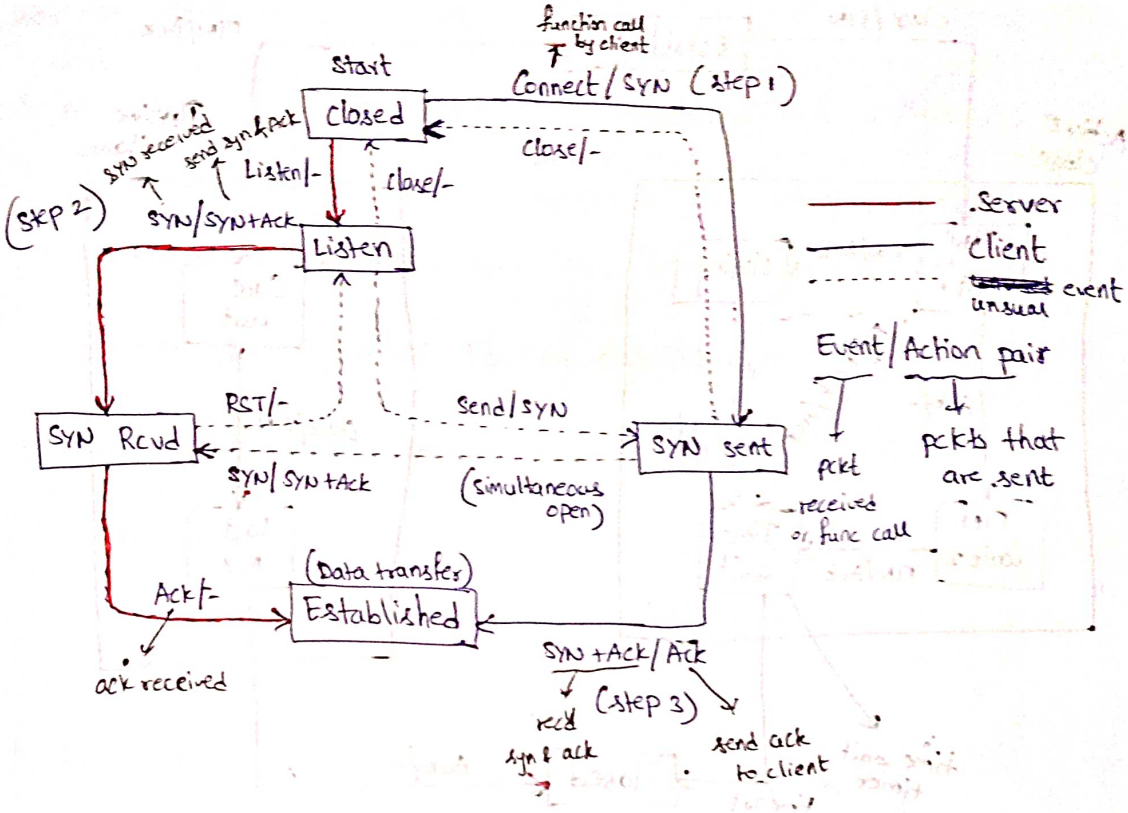
NAME: K. Gowtham STD.: _____ SEC.: _____ ROLL NO.: _____ SUB.: _____

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
		Computer Networks		
		TCP (Continuation)	03 - 04	
		UDP	05 - 06	
		Hardware & other Devices in N/W	07 - 19	
		Application Layer	20 - 25	

Page No. | Teacher's Sign / Remarks

16/12/20

TCP State transition diagram: Connection Establishment



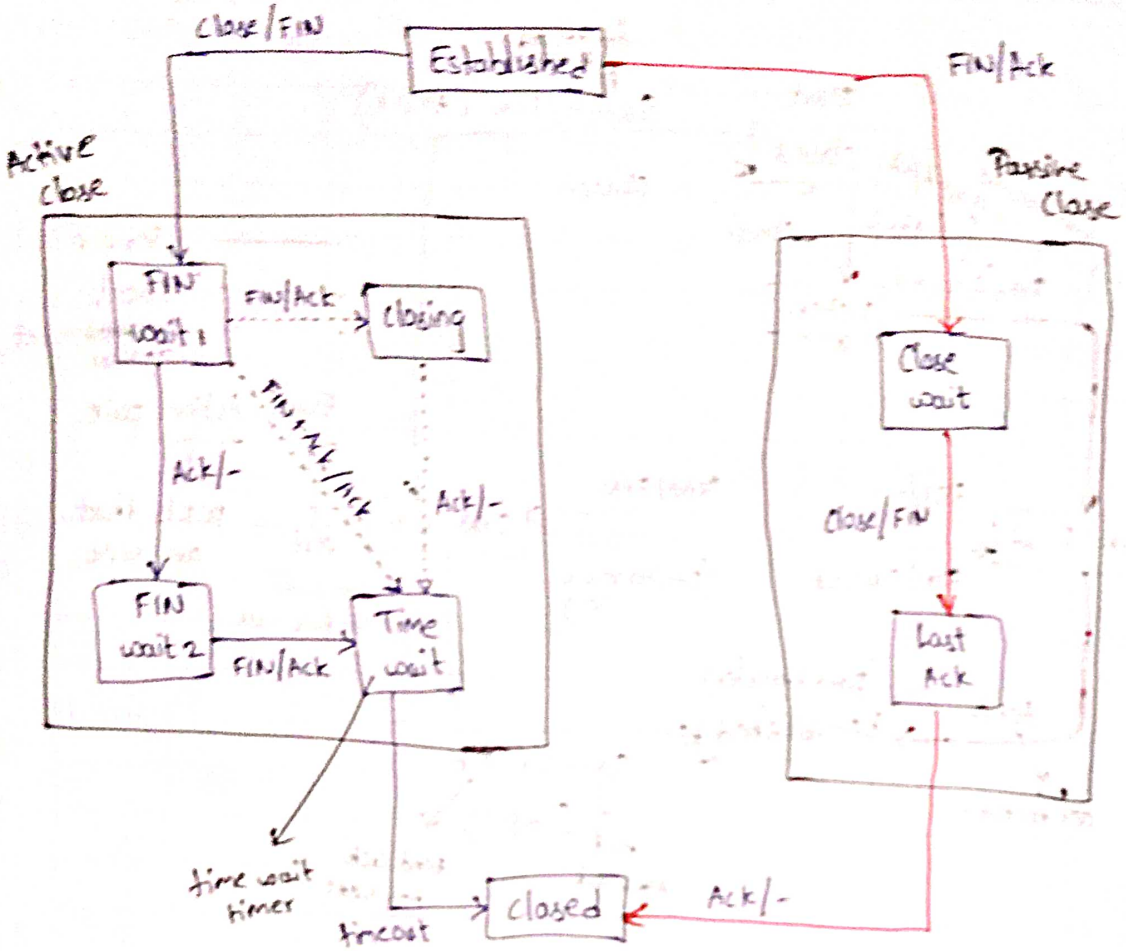
→ At "syn sent" state, if client doesn't receive (SYN+ACK) for long enough time, then client executes close system call and moves to "closed" state.

→ At "Listen" state, if server doesn't receive any connection requests then server executed close and moves to "closed" state.

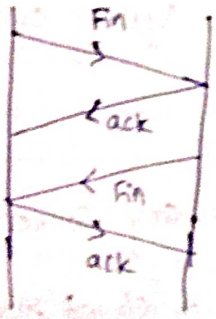
→ At "SYN RCVD", if server finds any problem, it ~~call~~ calls rest and terminates the connection.

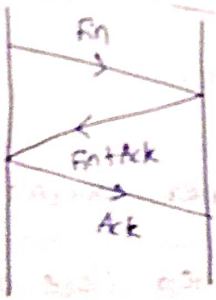
→ Sometimes server may initiate connection. In this case server at "Listen" state sends SYN. Then client at "SYN sent" sends (SYN+ACK) on receiving SYN from server. This is called Simultaneous open as both server & client wants to open connection at a time

TCP state transition diagram: Connection Termination

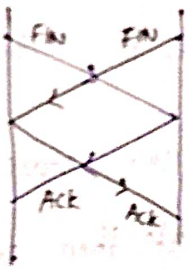


Termination of connection may happen in any of below way





FIN & ACK are sent at a time by server



Both client & server sends FIN at a time

All the above cases are shown in the transition diagram.

UDP

→ Need for

some

* app

when

connec

Eg: DN

* Broadca

Hea

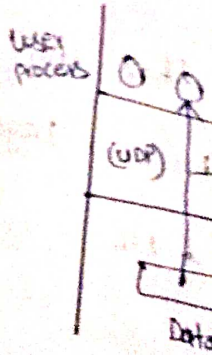
with

∴ w

* Some ap

which

Eg:



→ UDP header

urgent Pointer

04

UDP (Null Protocol)

→ Need for UDP:

* Some applications need one request & one reply.

when we need only one request & one reply, the connection establishment, connection termination are overheads

Eg: DNS, BOOTP, DHCP, Routing algorithms (for sharing DV)

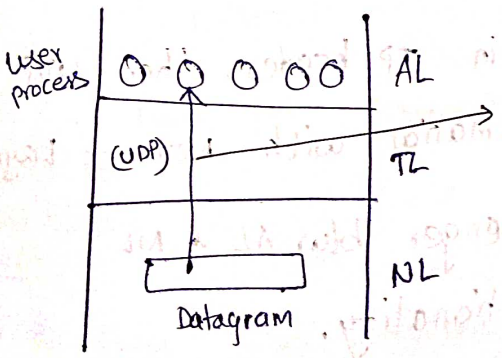
* Broadcasting & Multicasting applications.

Here if we use TCP, we need to establish connection with each host and reserve buffers for each connection.
∴ we use UDP.

* Some applications require speed than reliability.

Using TCP applies end to end congestion control due to which speed fluctuates.

Eg: Multimedia applications (watching videos)
Online Gaming etc..

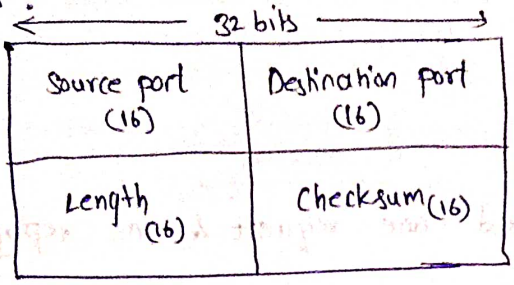


The received datagram is directly given to processes in AL bypassing the services of transport layer.

→ UDP header doesn't need seq num, ack, window size, flags, urgent, pointer, options.

∴ no flow control

UDP header:



Header size = 8 bytes

UDP has fixed header size

length: length of datagram including header

Checksum: Checksum is computed on UDP header, UDP data, psuedoheader of IP.

→ Since UDP is unreliable, checksum may not be used. when checksum is not used it is set to (000...0)

→ However, if the computed checksum itself is 000..0 then we store 111...1 in checksum.

Checksum is stored in 1's complement form

∴ If 0 is represented as 000..0 ⇒ Checksum not used

0 is represented as 111...1 ⇒ Checksum is used

Other functionalities of UDP:

→ If AL wants to set options in IP header, then UDP has to communicate this information with Network Layer.
i.e., UDP has to act as messenger blw AL & NL
TCP also has this functionality.

→ ICMP pckts received at IP, has to be informed to AL through UDP

Hardware and other devices used in Networking

→ 2 types of channels

Base band	Broad band
→ only one station (i.e., 1 signal) can be transmitted at once	→ Multiple hosts can transmit at a time
∴ No multiplexing	Multiplexing is possible

→ The wires come in different types. Few are shown below.

* 10 Base T (10 Mbps, No multiplexing, 100m)
* 10 Base 2 (10 Mbps, " , 200m)
* 10 Base 5 (" , " , 500m)
* 100 Base T (100 Mbps, " , 100m)

→ The third field is distance for which the wire can run.
i.e., It is max distance for which signal can be transmitted without attenuation.

∴ LAN Segment with 10 Base T can range only for 100 m
↳ LAN connected to one wire

Note:

(i) All these wires operate at PL.

(ii) Attenuation is possible for all.

(iii) Collisions are possible.

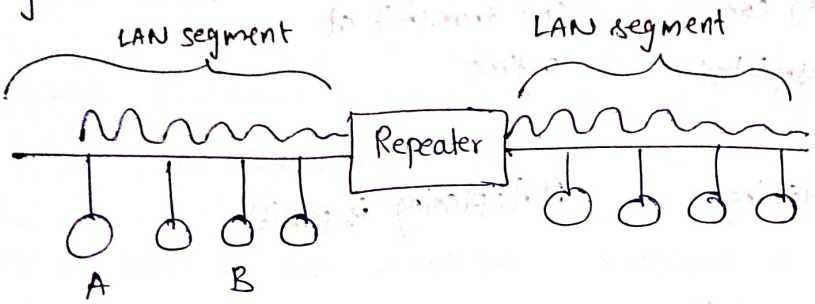
Collision Domain is n.
i.e., no of stations involving in collision

Here length of LAN depends on the type of wire.

Repeater:

→ This is used to increase the length of LAN by connecting 2 LAN segments.

→ Repeater takes in signal which is about to die and regenerated it



→ Even if A want to send data to B, repeater still sends data to other side.

→ The LAN segments connected to repeater must be of same type. i.e., both should be ethernet or both should be token ring.

→ Repeater is different from amplifier. Amplier increases the amplitube where as repeater just regenerate the original signal

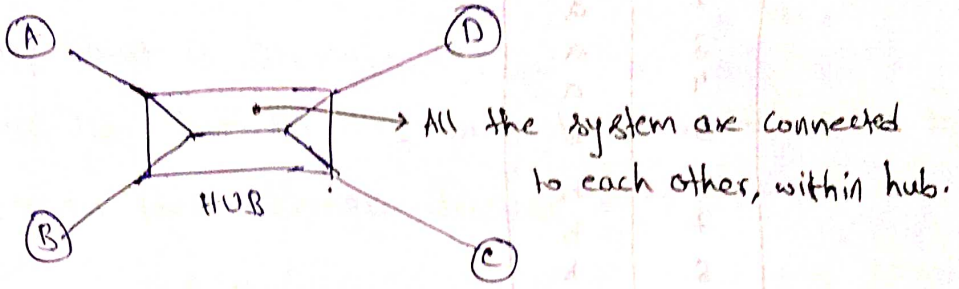
→ Repeater runs at PL
i.e., repeater is purely hardware.

→ Collisions are possible in repeater and hence all the stations in the network are in one collision domain.
∴ Repeater doesn't reduce collision domain. i.e., no

Hub:

→ Hub is a multi-port repeater.

→ For example, a 4-port hub can connect 4 stations



Here if B needs to transmit a pckt to A, the pckt will be transmitted to every station

∴ Traffic is very high

→ HUB has only PL.

→ Collision are possible inside hub.

Collision Domain = n

∴ Hub doesn't decrease collision Domain.

Adv: Hub is cheaper.

Bridge:

→ Bridge is used to connect 2 LANs of different types.

(Repeater connects 2 Lan segments of same LAN)

→ If a bridge has n ports, then it can connect n LANs.

→ Also the 2 LANs connected by bridge may be different

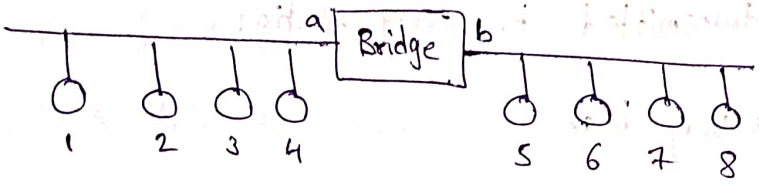
i.e., one may be ethernet and other may be token ring.

→ Bridge operates at PL & DLL

→ Since bridge works at DLL, it can look in MAC addresses.

→ Every bridge has a forwarding table.

MAC	Port
1	a
2	a
3	a
4	a
5	b
6	b
7	b
8	b



Bridges

- Static Bridges
- Dynamic / Learning / Transparent bridge.

Static Bridge:

→ Here the forwarding table is static and whenever we need to make a change in the network we need to change the table manually.

Dynamic Bridge:

→ Here bridge will learn the MAC & port of a station.

→ Whenever a station transmits data, the data reaches bridge and thus bridge will understand the MAC of the station and puts an entry in the forwarding table

→ But dynamic
host which

→ Bridge is
i.e.,

→ Bridge is
i.e., s

→ If a brid
present

→ Bridge is cap
Since
N/w

∴ Collis

Problems with

→ If we c
bridges,
table is
Now if
sent pc
Another
with
a to c

MAC address.

→ But dynamic bridge never knows the MAC address of a host which has never transmitted any data.

→ Bridge is capable of filtering (∵ Bridge has DLL)

i.e., It sends a pckts to LAN in which the req host is present.

If the dest host is present in the N/w of sender, then bridge discards the pckt.

→ Bridge is capable of forwarding.

i.e., sending data from one N/w to other N/w.

→ If a bridge doesn't know where the destination is present, then it floods the pckt.

∴ Bridge is capable of Flooding.

→ Bridge is capable of store & Forward.

Since it can store, collisions blw hosts of diff N/w are not possible.

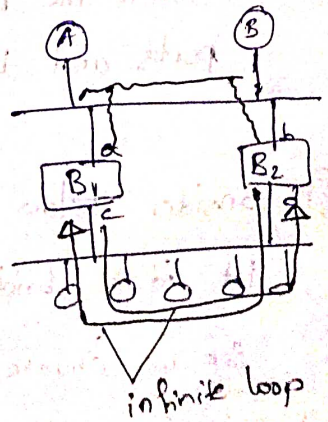
∴ Collision Domain is reduced.

Problems with bridges

→ If we connect two lans using 2 bridges, and assume initially forwarding table is empty.

Now if A send data to B, this sent pckt is fall in infinite loop.

Another problem is that port of A. for B1. keeps on changing from a to c & c to a.



whenever we
we need to

of a station.
data reaches
of the

→ If MTU of LANs connected is different, then bridges cannot do the job of fragmentation.

So the fact that bridges can connect different types of N/w is theoritical but not much practical.

17/12/20

Spanning Tree Algorithm:

→ This algorithm is used to avoid infinite loop that pckts fall into, when two LANs are connected by more than one bridge.

Algo:

(i) Every bridge has a built-in ID. The one with smallest ID is taken as root bridge.

(ii) Mark one port of each bridge which is closest to root bridge as root port.

(iii) Every LAN chooses a bridge closest to root bridge as a designated bridge for that LAN. make the corresponding port as designated port.

(iv) Mark the root port and designated port as forwarding ports and block remaining.

Eg: Consider below N/w with 4 LANs and 5 bridges.

let id of bridge B_i be i.

So we choose B_1 as root bridge.

→ Here closest is defined with some cost parameters.
Here we consider no of hops

root bridge

LAN2

(designated bridge: B1)

→ Root por

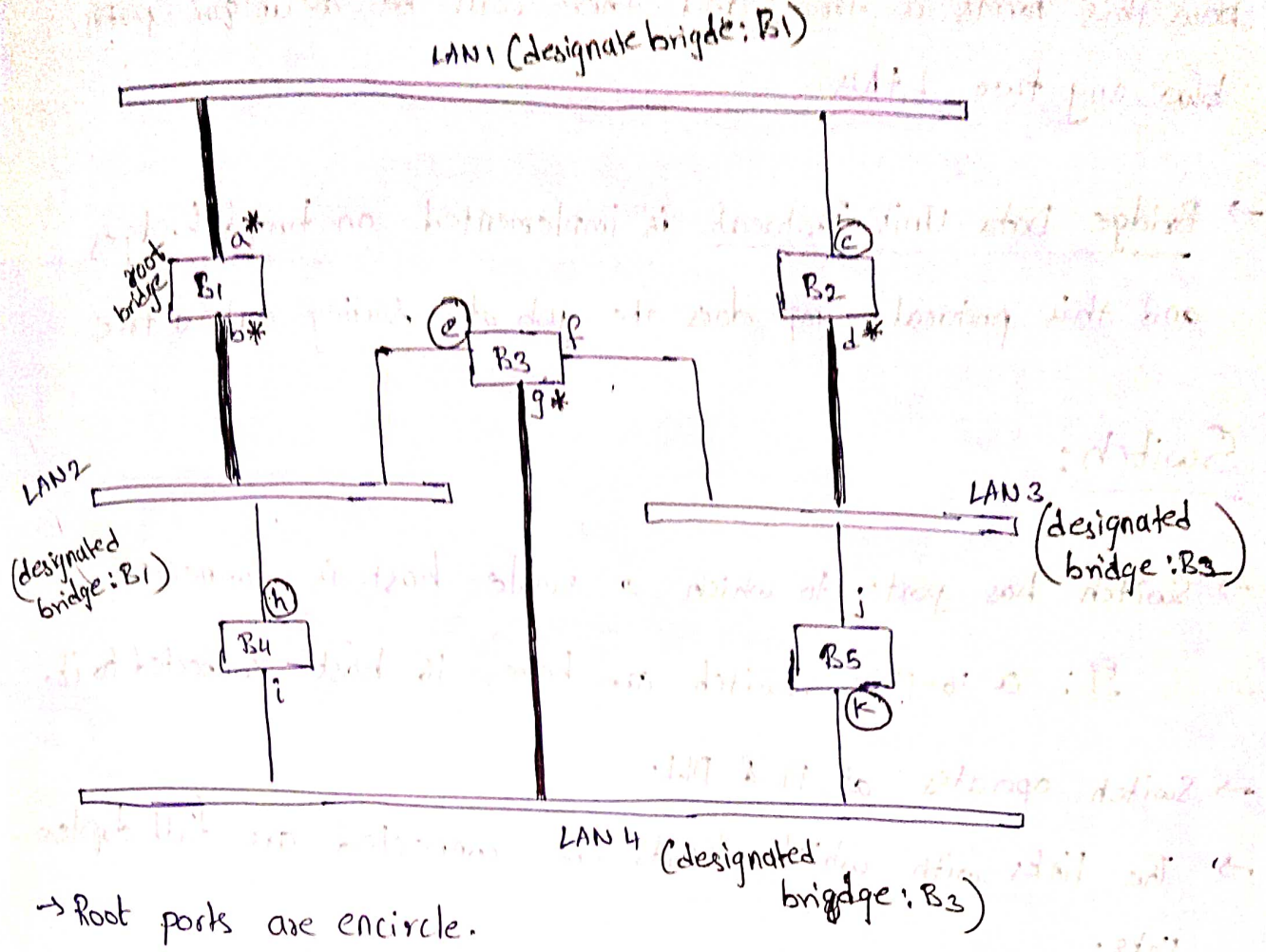
→ when

choose

→ Now w

and bloc

LAN 2

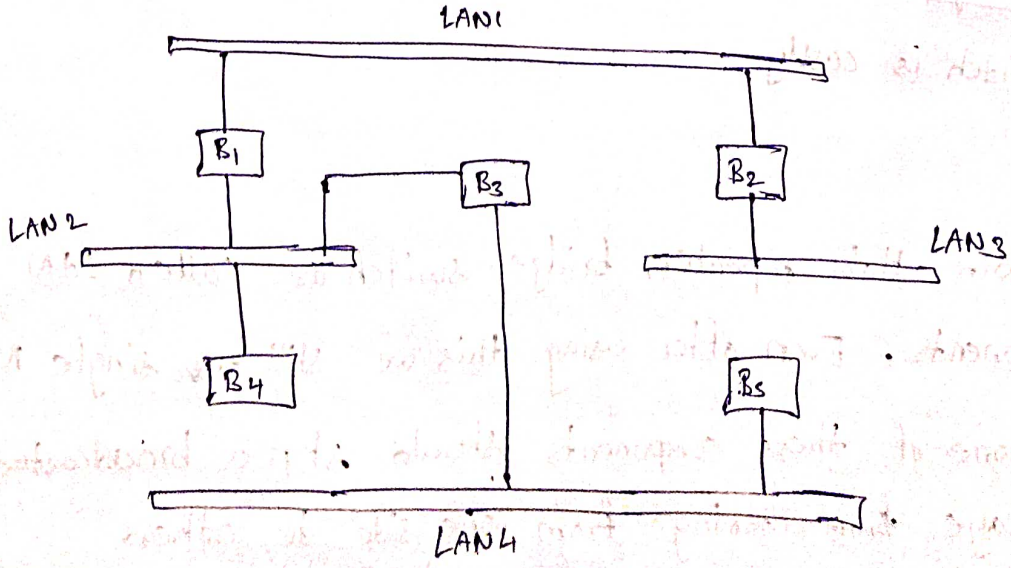


→ Root ports are encircle.

→ when ~~two~~ there is a conflict in choosing designated bridge, choose the bridge with least id.

——— (thick line) – line that ~~two~~ connects LAN & designated port (*)

→ Now we use root port & designated ports as forwarding port and block remaining



how this forms a tree and there will be a unique path btw any two LANs.

→ Bridge Data Unit Protocal is implemented on ~~brigd~~ bridges and this protocal ~~imp~~ does the job of sorting out a tree

Switch:

→ Switch has ports to which a single host is connected.

Eg: a 16-port switch can have 16 host connected to it.

→ Switch operates at PL & DLL.

→ The links with which hosts are connected are full duplex links.

→ Switch is designed in such a way that it can allow more than one communication occur at a time.
Switch can forward pckt to crct host. (∵ it has DLL)

→ ∴ No collisions are possible using a switch.

Collision Domain = 0.

→ Traffic is less with switch.

Disadvantage:

→ Switch is costly.

Note:

→ ~~use~~ wire, Hub, repeater, bridge, switch are called LAN components. Even after using this we still have single N/W. So none of these components should stop a broadcasted message ~~from~~ entering from one side to other

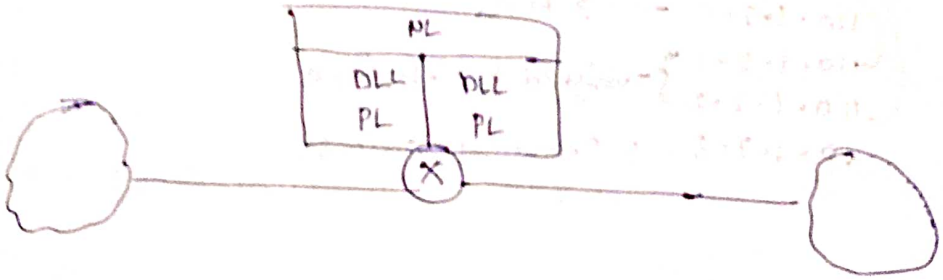
∴ These components doesn't change broadcast domain.

	Broadcast Domain	Collision Domain
Repeater	Same	Same
Hub	Same	Same
Bridge	Same	reduces
Switch	Same	reduces
Routers	reduces	reduces
Gateway	reduces	reduces

Router:

→ Router is a device used to connect 2 networks of same type.

→ Router has PL, DLL, NL



→ Router does the job of filtering

i.e., broadcast, BootP, DHCP, ARP etc. are discarded and are not let to cross other networkes.

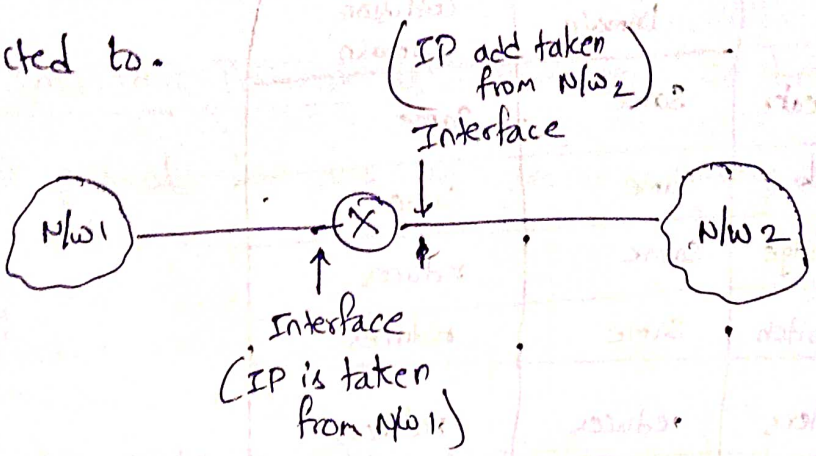
→ Router can store & Forward.

→ Router can do flooding / routing.

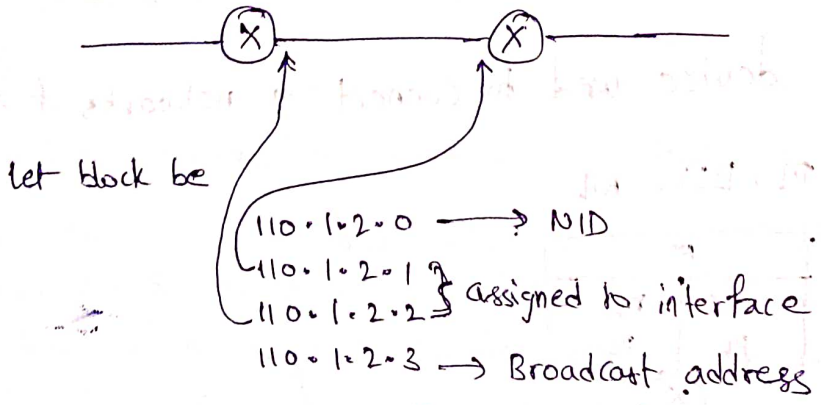
→ No collision are possible within a router

→ Every interface of router has an IP address.

The IP address is taken from ~~one~~ the N/w that router is connected to.



→ However if two routers are connected, then we need to ~~buy~~ buy a block of IP add of size 4.



Gateways

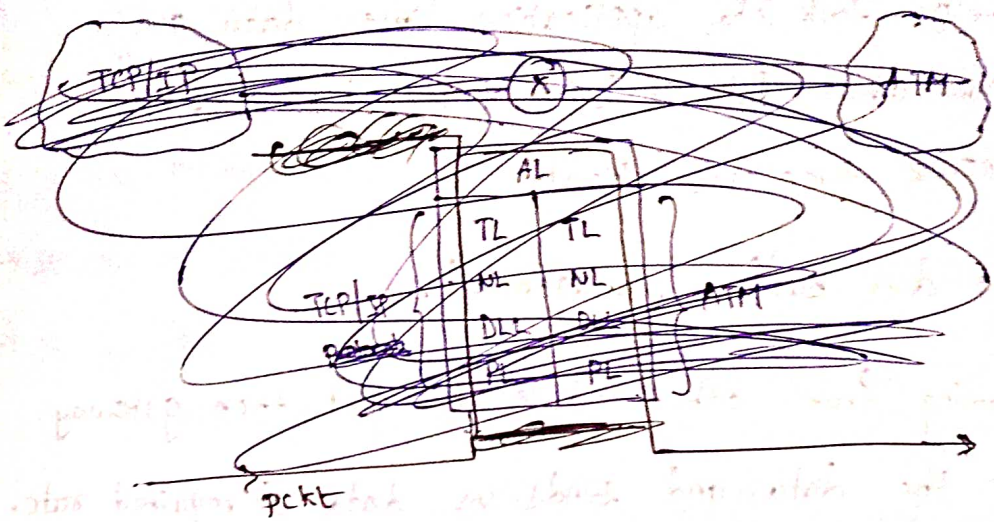
Router is capable of connecting similar type of N/ws.

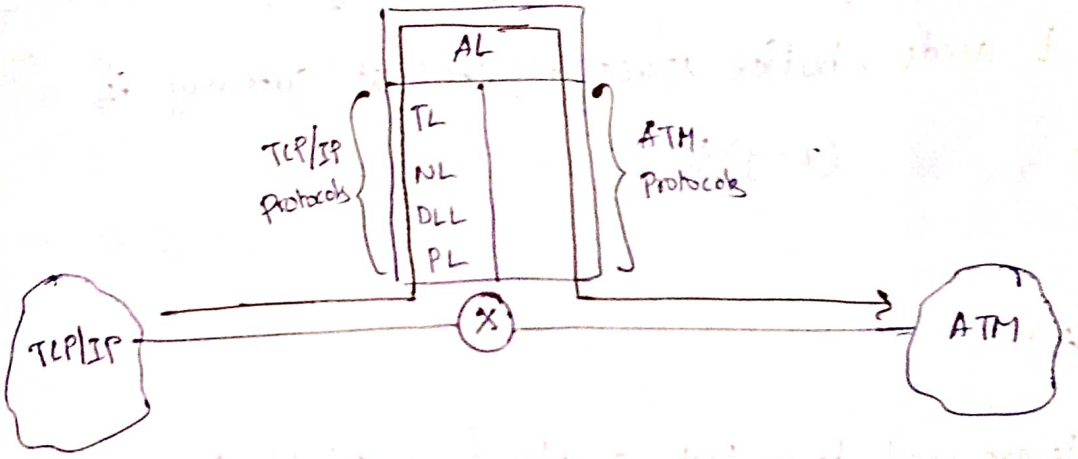
i.e., two TCP/IP N/w or two ATM N/ws

but not a TCP/IP and an ATM N/w

→ ~~In~~ For this need we use Gateways.

→ Gateway is used as protocol converter.





→ Gateway is is used as proxy.

* ie A proxy can monitor every pckt that is going out of the network.

* Proxy also does the job of buffering / caching
  ↓
  i.e. recent accesses are stored.

→ Gateway is use as NAT server
  ↳ used to conserve IP addresses.

→ Gateway is also used as Firewall.

Firewall monitors every incoming & outgoing pckts and can impose some restrictions.

→ Another use of Gateway is Deep Packet Inspection (DPI)

DPI helps look into application layer data and thus know what data is being transmited and can also impose some restrictions.

→ Gateway also does buffer management.

If data coming from other end is too fast, then gateway can buffer the data and sends us data at required rate.

* If x is incoming rate of data & y is outgoing rate then for t sends, buffer space required at gateway is

$$(x-y)t$$

Firewalls:

→ Firewalls are used to protect a n/w from outside in internet.

→ Any incoming/outgoing pckts has to go through firewall.

Firewalls are of 3 types

(i) Layer 3 Firewall
(ii) Layer 4 Firewall
(iii) Layer 5 Firewall / Proxy

Layer 3 Firewall:

→ It has PL, DLL, NL

This firewall can

(i) Block hosts (using SIP, DIP)
(ii) Block certain protocols (TCP, UDP, ICMP, IGMP)
- Block ICMP can save from ICMP attacks.

(iii) A protocal from particular host can be blocked.

Layer 4 Firewall:

→ It has PL, DLL, NL, TL

→ It can do everything a Layer 3 Firewall can do

→ It can:

(i) Block a service (by looking into port number)

(ii) Block a particular service from particular host.

Layer 5 Firewall / Proxy Firewall:

→ It has all 5 layers.

→ It can do ~~any~~ every thing a layer 4 Firewall can do.

→ It can:

(i) give authentication (∵ it can look into username & password using AL)

→ Here a rule table is maintained.

A pckt is discarded if it matches ~~which~~ with atleast one rule.

Application Layer Protocols

→ DNS, HTTP, FTP, SMTP, POP

DNS (Domain Name Service)

→ port No: 53
→ Given a domain name, DNS gives its IP address.

Even if we remember IP add, there is no guarantee that the IP address of a website doesn't change

Types of domains:

(i) Generic Domain:

.com, .edu, .mil, .org, .net

↓ used by commercial organizations

↓ used by non-profit organization

(ii) Country Domain:

.in, .us, .uk ....

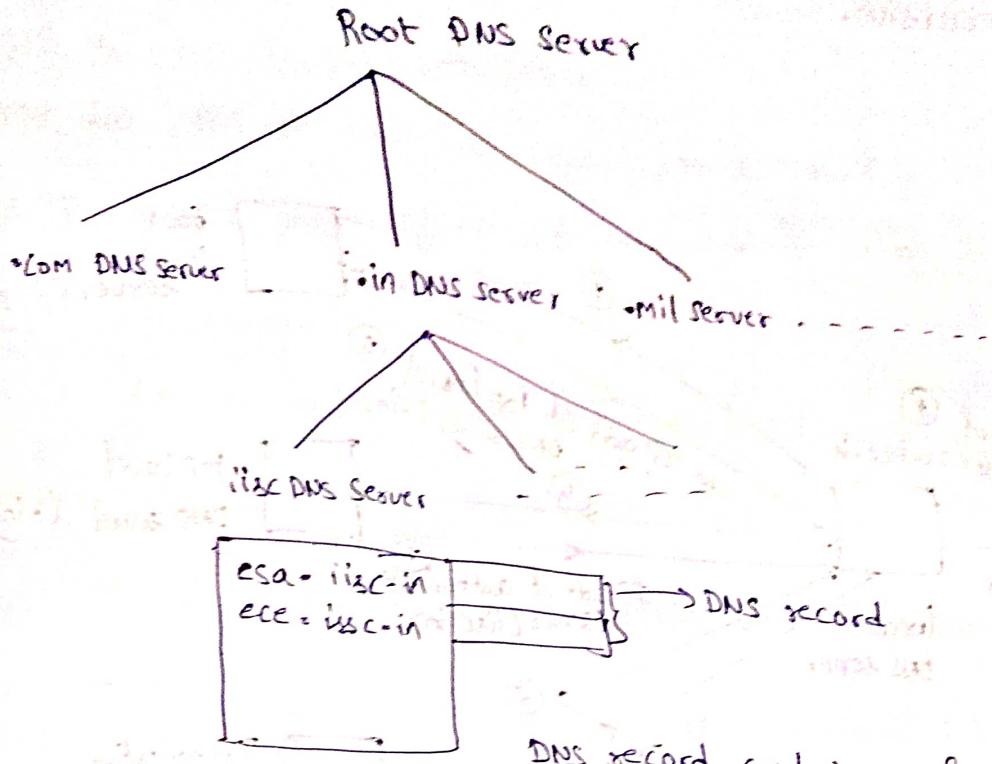
(iii) Inverse Domain:

Given an IP address, DNS can also give domain name.

→ DNS also does load balancing.

i.e., if a website have more than one IP add (more than one server) then each time we access the website it will take us to a different server.

Data organization in DNS



DNS record contains information like how long the IP address will be assigned to that domain, etc.

→ Data of DNS is distributed.

→ There are 13 root DNS server so that even if one server doesn't work there will be no problem

Note:

→ However, we don't need to go to Root DNS server everytime.

→ ISP provides local DNS which has most frequently used websites. If local DNS can't help then we go to root DNS server.

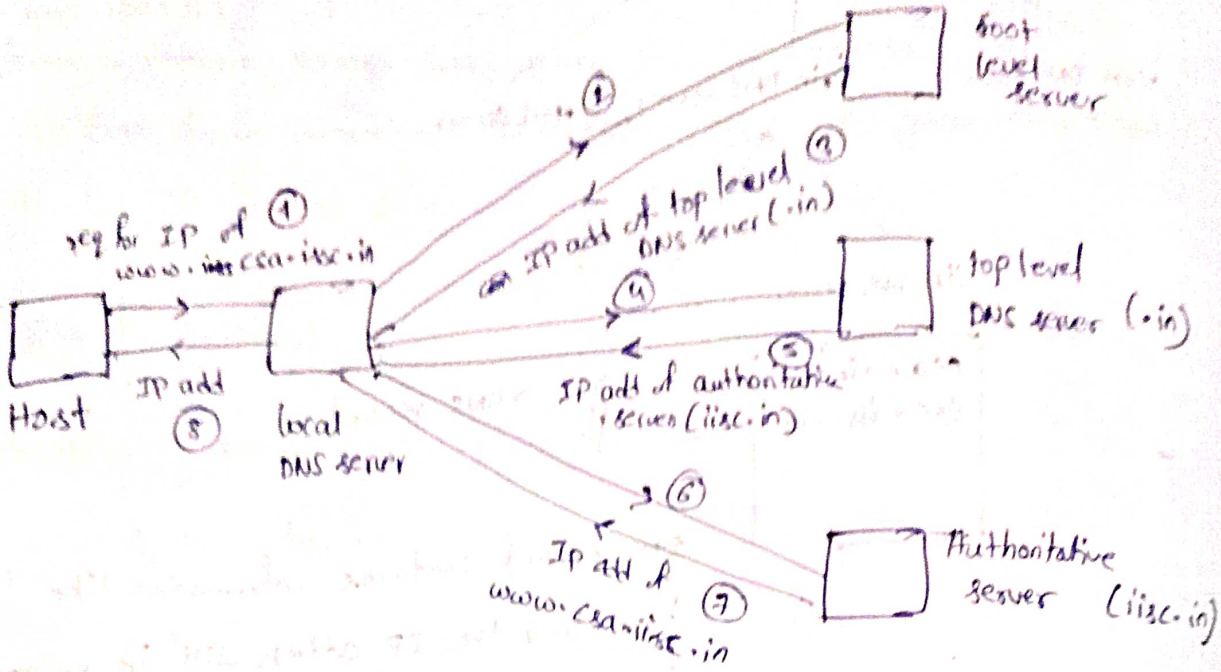
→ The record of local DNS has a field Time to Live. after which entry is deleted.

There are 2 ways through which local DNS contacts root DNS server.

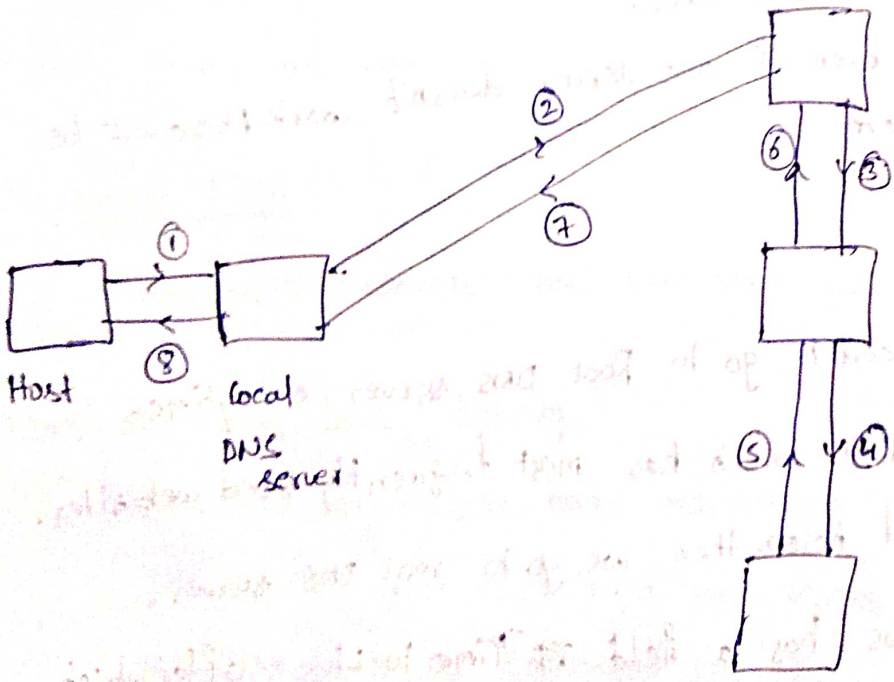
i) iterative

ii) recursive.

Iterative:



Recursive:



→ ~~All the~~ DNS uses UDP (since all the messages are single req & reply)

HTTP:

→ port number : 80

→ HTTP is used to get web pages.

→ HTTP uses TCP at transport layer (for reliability)

HTTP doesn't have any inbuilt mechanism for reliability.

→ HTTP is inband protocol:

ie, commands & data are transfered using same connection.

→ HTTP is stateless protocol.

ie, It is not going to maintain any information abt users.

∴ HTTP uses a cookies to know the previous history of communication. These cookies are stored at users' end.

→ HTTP 1.0 uses non-persistant connection

ie, a seperate connection is used to fetch each object.

For example if a web page has 10 image, we used 11 connections (one for each image & one for web page)

Here server doesn't need to hold connection for long time.

→ HTTP 1.1 uses persistant connection.

ie, a single connection is used to fetch all the data.

Here server holds connection for long time. So here congestion window grows and thus we have high BW.

Methods used in HTTP:

(i) Head: used to obtain header (meta data) of a webpage.

Usage: Gateway caches frequently used data.

So by getting head, gateway knows whether the data is modified or not. If not modified, data from gateway cache can be sent.

There are many other uses

(ii) Get: Used to get the webpage

(iii) post: post is used to send data when we fill po forms

(iv) Put: used to upload something

(v) Delete: used to delete an object.

(vi) Trace: Used to know the servers involved in sending data.

(vii) Options: used to know whether the above functionalities are provided by a server or not.

(viii) Connect: used for security (Authentication).

FTP (File Transfer Protocol)

→ ~~Protocol~~ port number: 21

→ Example applications: Techia, Filezilla

→ FTP uses TCP

→ FTP uses out of band connection

i.e., seperate connections are established for commands & data.

Control Connection (used for commands) (port: 21) (Persistant)

Data Connection (used for data) (port: 20) (non-persistant)

∴ seperate connection is opened for each data transfer

FTP server: data of FTP server is used by FTP client.

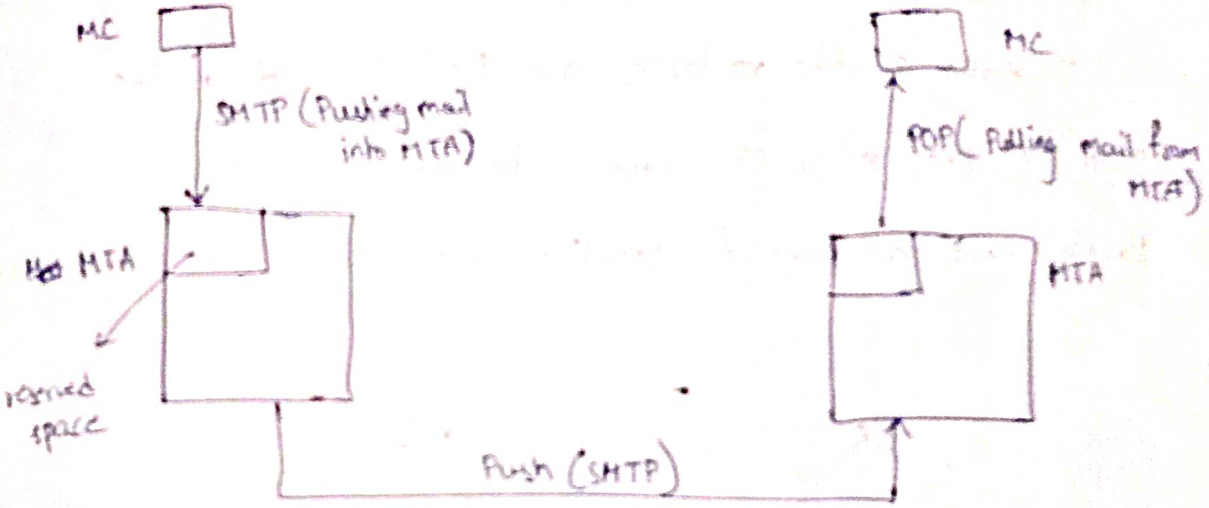
→ FTP is stateful protocol.

∴ FTP is going to log every activity.

SMTP (Simple Mail Transfer Protocol) and POP (Post office Protocol)

→ FTP requires both the stations to be online while file transfer.

→ But SMTP doesn't need this. So we don't use FTP for emails.



MTA: Mail Transfer Agent

MC: Mail Client

→ SMTP & POP uses TCP for reliability.

→ To send data other than text, this non-text data has to converted into text format on sender side and it has to converted back into non-text format on receiver side.

The protocol used for this conversion is MIME

MIME: Multipurpase Internet Mail Extension.

→ SMTP & POP are inband protocols.

28/12/20

Routing : Shortest Path Algorithm

→ This works on dijkstra's algorithm

→ This is a fixed routing algorithm (static)

→ Here we find shortest path based on dijkstra's algorithm.

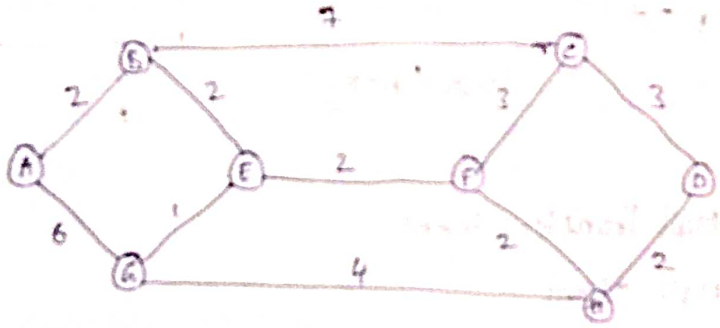
→ Here cost may be no of hops (or) physical distance (or) it may depends on bandwidth traffic etc.

→ A label may be tentative or permanent.

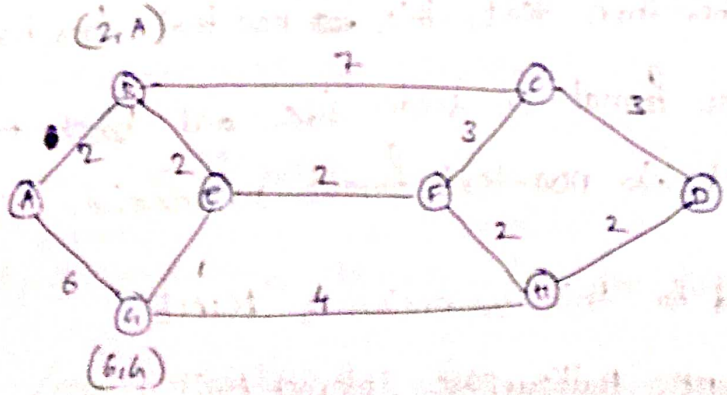
Initially all the no lables are tentative and as the algorithm proceeds few labels become permanant.

Labels that are part of shortest path are made permanant.

Eg:

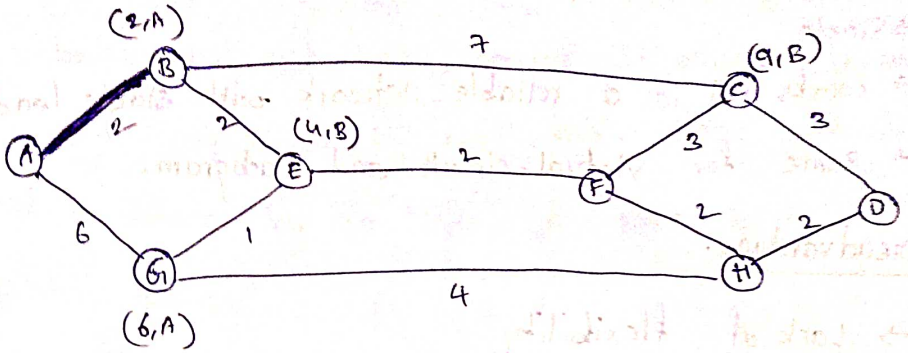


{A}



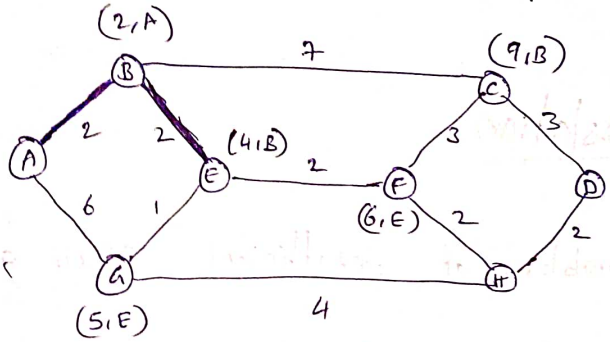
Include B in the set and now edge AB becomes permanant

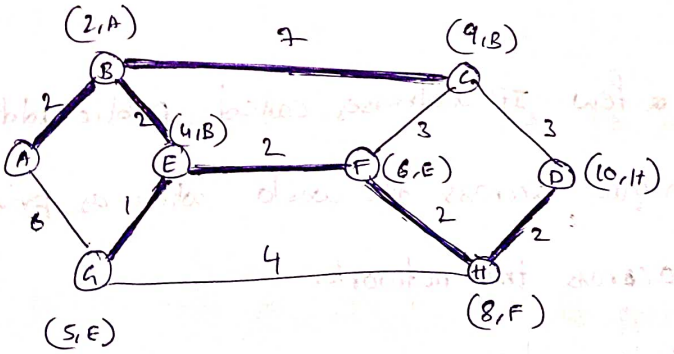
{A, B}



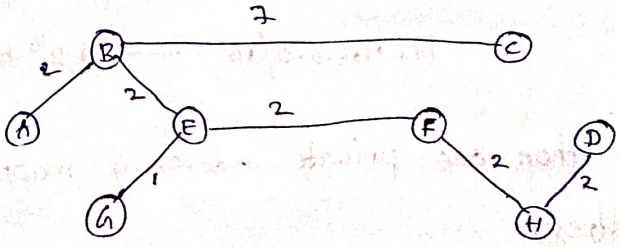
Include E and now BE becomes permanant

{A, B, E}





∴ Routing is done with below tree



Advantages of Fixed Routing

→ Simple

→ works well in a reliable network with stable load

→ Same for virtual-circuit and Datagram.

Disadvantage:

→ Lack of flexibility

→ Does not react to failure or network congestion.

29/12/20

Network Address Translation

→ NAT addresses the problem of insufficient IPV4 addresses.

→ Every network is connected to a NAT box.

→ Systems within the network have unique IP addresses called private address

→ NAT has one or a few IP addresses called public addresses.

→ Public address is unique accross the world where as private address is unique acaross the network.

The range of private addresses given are:

$10.0.0.0/8$ — 2^{24} hosts

$172.16.0.0/12$ — 2^{20} hosts

$192.168.0.0/16$ — 2^{16} hosts

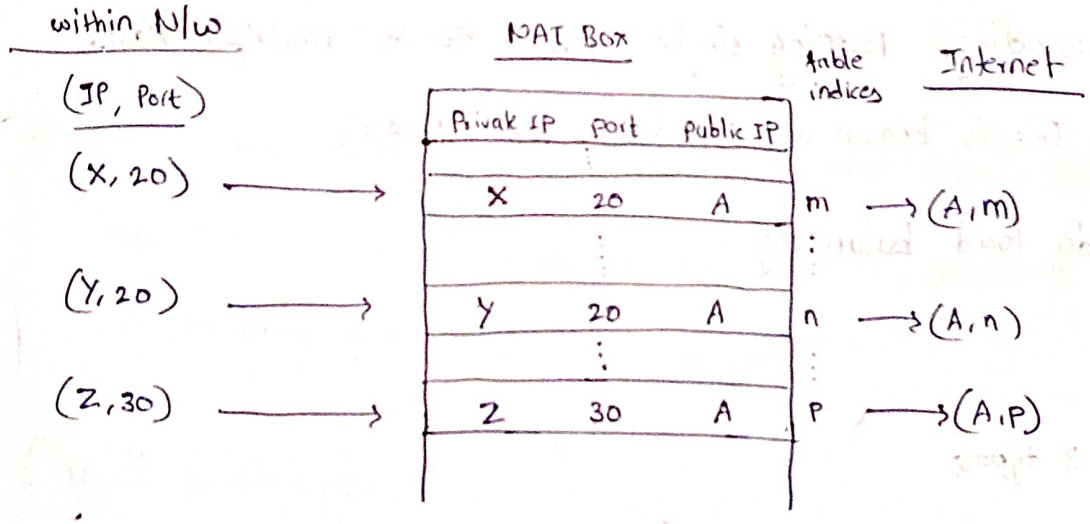
→ Now, more than one private address is mapped to a single public address

→ Working:

→ When a pckt is being sent out, the source IP, which is private IP, is replaced with public IP by NAT box. Along with this, source port field is replace by an index of NAT box's 2^{16}-entry translation table.

→ By replacing source port we will be able to distinguish two machines using same source port

→ When a packet is incoming, the destination port is used to index into translation table to get the private IP and port of the station in the network.



Say packet with dest IP = A and dest port = n is incoming, using port=n, we index into table and get the pair (Y, 20)

Issues with NAT:

→ Since mapping in NAT box is built by outgoing pckts, incoming pckts can't be accepted until after outgoing ones.

→ NAT maintain information even for connectionless service.

→ NAT breaks independence blw layers.
i.e., for example if TCP is upgraded and no of bits in port change, then NAT fails.

→ If a protocol other than ~~NAT~~ TCP or UDP (for example, for some multimedia application) then NAT won't able to locate TCP port correctly

Advantages:

→ Supports more systems.
→ Allow easy interchange blw ISPs by changing IP addresses in NAT boxes without changing internal system address.
→ IP masquerading: Mapping single public IP to multiple hosts
This is known as port based NAT (PNAT)
→ NAT can do load balancing.

Note:

NAT are of 3 types:

(i) Static NAT: one to one mapping from private to public addresses
~~(ii)~~ It doesn't solve the problem of less no of IP addresses

(ii) Dynamic NAT: one to one mapping from active private IP address to Public address. Solve the problem of less IP address to some extent.

(iii) PNAT: The NAT that's been discussed till now is PNAT and it solves the problem of less no of IP addresses.

Circuit Switching & Packet Switching

Packet Switching:

→ No reservations of resources is made.

→ Store & forward mechanism is used.

→ Packetization is done.

→ Queuing Delay is present.

For every outgoing link of a switch, a queue (o/p buffer) is maintained. The amount of time packet spends in this queue is called queuing delay.

Queuing delay depends on congestion level in the network.

→ If arriving pckt has no space in the queue then pckt loss occurs. when queue is full either arriving pckt or an existing pckt may be dropped.

Circuit Switching:

→ For every connection, reservations are made and path is set. The time involved in this is called setup time.

→ Circuit switch N/ws are implemented in 2 ways:

(i) Frequency Division Multiplexing (FDM)

(ii) Time Division Multiplexing (TDM)

→ FDM:

* The available frequency is divided among the connections established.

TDM:

→ Here time is divided into slots of fixed size.

→ Each connection (sender-receiver pair) is given a slot to transfer. The set slots of the connections forms a frame.

* In both FDM & TDM the allocated bandwidth or time slot remains unused if the host doesn't want to transfer. This leads to wastage of resources.

Circuit Switching	Packet Switching
* suitable for real time services. (telephone calls etc.)	* provides better sharing
* more costly	* simple & efficient
	* less costly.

Computer Networks

Classfull IP address Classification:

Class A: (0 ___)
* NID: 8, HID: 24
* $2^7 - 2$ N/w
* $2^{24} - 2$ Hosts
(0+1-126)

Class B (10 ___)
NID: 16; HID: 16
* 2^{14} N/ws
* $2^{16} - 2$ hosts
(128-191)

Class C (110 ___)
NID: 24; HID: 8
* 2^{21} N/ws
* $2^8 - 2$ Hosts
(192-223)

Class D (1110 ___)
No NID & HID
* used for multicasting
2^{28} addresses
(224 - ~~240~~ 239)

Class E (1111 ___)
No NID & HID
* Reserved
* 2^{28} address
(240-255)

→ NID 0 & 127 of class A are not used.
→ For every N/w starting add is n/w id

→ LBA: 255.255.255.255
→ DBA: NID.(All 1's)

Subnetting

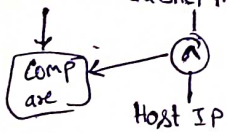
* Divide N/w by borrowing sufficient no of bits from host id part.
* Think abt confusion in interpreting certain IP add by insider N/w and outside N/w
* no of hosts = no of IP add − (2 * no of subnets)

Subnet mask: Nid + subnet id → All 1's
Hid → 0's

Subnet mask ∧ Host IP → subnet ID

Routing table structure

(Subnet id | Subnet mask | Interface)



→ If more than 1 match then subnet id with more 1's.

→ Fixed Length Subnetmasking
Variable Length Subnetmasking (VLSM)

→ Larger the subnet, smaller the ~~size~~ value of subnet mask.

CIDR (Classless Inter Domain Routing)

* IP add representation: a.b.c.d/n
n is no of bits in NID
* CIDR block forming rules:
→ IP add are contiguous
→ Block size in power of 2.
→ Start IP add / block size.

* Subnetting in CIDR is also same.

Supernetting / Aggregation

* Rules: → Contiguous blocks
→ equal sizes can be combined at once.
→ Start IP / total size

→ Using subnet mask a host determines whether a given IP add is in the same N/w or not (Think how).
If yes, then pckt will be sent directly otherwise through router.

FLOW CONTROL

Delays:

Transmission Delay: $\frac{L}{B}$; Propagation Delay = $\frac{d}{v}$

→ Queuing delay; processing delay.

Stop & wait:

→ pckt is sent and sender waits until ack is received

$$\eta = \frac{T_t}{T_t + 2T_p} = \frac{1}{1+2a}$$ (assuming queing, processing delay, ack transmission time are negligible)

$a \rightarrow \frac{T_p}{T_t}$

distance ↑ ⇒ η ↓

size of pckt ↑ ⇒ η ↑

no of pckts transmitted = $n + np + np^2 + \dots$ → same for SR also

p → probability of error.

Sliding Window Protocol / Pipelining:

$\eta = \frac{1}{1+2a}$ ⇒ Min window size for 100% efficiency = $\lceil 1+2a \rceil = \lceil \frac{1}{\eta} \rceil$

no of bits of seq num = $\lceil \log_2 (1+2a) \rceil$

→ But this not posible practically.

→ GBN, SR are implementations of sliding window.

Go Back N ($N > 1$)

Sender window size = N ; Receiver window size = 1;

→ out of order pckts not accepted. So for retransmission we need to go back N and retransmit

$$\eta = \frac{N}{1+2a}$$

→ GBN uses cumulative ack.

→ Buff = N+1; Seq num = N+1; Bandwidth req = high

CPU req = low; Implementation difficulty = easier than SR.

Selective Repeat:

Sender window size = Receiver window size = N

→ out of order pckts accepted.

$$\eta = \frac{N}{1+2a}$$

→ uses independent ack.

→ Buff = 2N; Seq = 2N; Bandwidth req = moderate

CPU req = high; Implementation is complex.

Note:

→ seq num ≥ sender win + receiver win

→ sender window = $\min(1+2a, 2^n)$ (n → no of bits in seq. num) → This is for sliding window (not for GBN, SR)

→ Ack
- Cumulative: ack timer is started (when a pckt is received) and ack is sent for group
- independent: 1 per pckt. SR sends negative ack and allows early retransmission.

Problem	Soln
Data pckt lost	Timeout timer
Duplicate pckt problem (ack lost)	Seq number
Delayed Ack (Missing pckt problem)	Seq num for Ack

Capacity of channel: Max no of bits present on the channel.

∴ capacity of channel = BW * T_p

High capacity channel is called thick pipe and low capacity channel is called thin pipe.

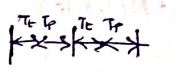
Access Control Methods

If T_p is given in bits then divide with bandwidth " " " " " meters " " " velocity.

→ when channel is broadcast channel we need access control methods

Time Division Multiplexing:

$$\eta = \frac{T_t}{T_t + T_p}$$



If a station doesn't transmit, then the slot is wasted.

Polling:

$$\eta = \frac{T_t}{T_{poll} + T_t + T_p}$$

Stations that are interested participate in polling.

CSMA/CD: (Carrier Sense Multiple Access Collision Detection)

→ Every host, before sending, sense carrier. Carrier sensing is done as long as data is transmitted

⇒ $T_t \geq 2T_p$ ⇒ $L \geq 2T_p * BW$

→ No acks are used.

If no sufficient data, extra bits are padded.

$$\eta = \frac{T_t}{c*(2\tau_p)*T_t + T_p} = \frac{T_t}{T_t + (2c+1)T_p} = \frac{1}{1+6.44a}$$

Probability of successfull transmission is $= {}^nC_1\, p(1-p)^{n-1}$ (p is probability to transmit)

This maximum when $p = 1/n$

⇒ Max value = $(1-1/n)^{n-1} = e^{-1}$ (for large values)

Backoff Algo / Binary Exponential back off algorithm:

→ Every pckt has a collision num initialized to 0.

→ If collision occurs, then collision num is incremented.

Now system choose a num b/w $[0, 2^n - 1]$ when n is collision num. Now system waits for this amount of time and retransmits.

→ As collision num increases, probability of collision ↓ exponentially.

→ Drawback: Capturing Effect.

Token Passing:

→ Stations are connected in ring topology
→ Every system transmits whenever it gets a token.

Ring latency = $\frac{d}{V} + N*t$

$t \rightarrow$ delay at every station

Efficiency, $\eta = \frac{N * T_t}{T_p + N * THT}$

Delayed token reinsertion	Early token reinsertion
$THT = T_t + \text{ring latency}$	$THT = T_t$
$= T_t + T_p + 0$ (assuming t=0)	
$= T_t + T_p$	
$\eta = \frac{N * T_t}{T_p + N(T_t + T_p)}$	$\eta = \frac{N * T_t}{T_p + T_t}$

ALOHA:

→ any station can transmit any time. So collision is possible.
→ Acks are used; No collision detection.

Pure aloha

- vulnerable time = $2T_t$

$\eta = G * e^{-2G}$

G → no of stations who wants to transmit in T_t

$\eta_{max} = \frac{1}{2e} = 0.184$ (at $G = 1/2$) i.e., 18.4%

Slotted Aloha

- vulnerable time = T_t

$\eta = G * e^{-G}$

$\eta_{max} = \frac{1}{e} = 0.368$ (at $G = 1$) i.e., 36.8%

ERROR CONTROL METHODS

Error Detection: Detect and req for retransmission
- → Send data twice (DTD)
- → Parity check
- → CRC
- → Checksum

CRC: Both sender and receiver will have CRC generator.
→ An n-bit CRC generator is represented using a polynomial of degree n-1.

For data we append 'n-1' bits and perform XOR division
→ The obtained remainder is called CRC and it appended to original data.
→ Now receiver gets remainder 0 is data is received correctly.

Checksum:

→ To compute n-bit checksum, divide data into groups of n-bits each.
→ Now add all these. (If carray occurs add to the LSB)
→ If checksum field is present in data, consider it 0.
→ Now negate the result and store it in checksum.
→ Now receiver must get 0 when added all these.

Note:
Meaningful errors are possible.

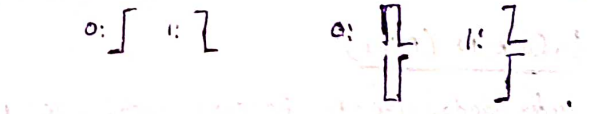
Error Correction:
↳ Hamming Code (Refer Notes)

ISO-OSI Layers

→ A host need not contain Nlw layer but TL is present on every host and NL is present on every router.

Physical layer:

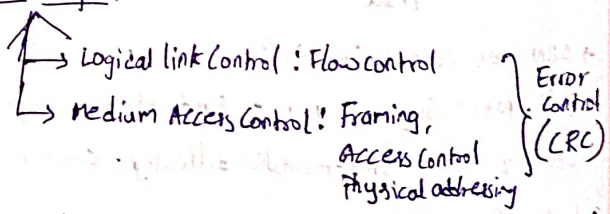
→ signal conversion blw different transmission media.
→ transmission modes: simplex, half duplex, full duplex.
→ Topologies: bus, star, ring, mesh, hybrid.
→ Encoding: Manchester, Differential Manchester Encoding



→ It provides bit synchronization, bit rate control.

baud rate = 2x bit rate

Datalink layer:



Transport layer:

→ End to End connection. Service point addressing.
→ Flow control
→ Error control
→ Segmentation & reassembly
→ Multiplexing & Demultiplexing
→ Congestion control

Network Layer:

→ Host to Host connectivity
→ Logical addressing.
→ Switching
→ Routing
→ Congestion control
→ Fragmentation.

Session Layer

→ Authentication & Authorization
→ checkpointing.
→ Synchronization.
→ Dialog control, logical grouping
→ Connection establishment, maintainence, termination.

Presentation Layer:

→ character translation
→ Encryption & Decryption
→ Compression.

this all layers work together (Pg: 70)

Framing:

→ SFD is used at the beginning of every frame
↳ (10)*11
→ we also need to detect end of frames

Framing
- fixed length (End is implicit)
- variable length
  - using length field
  - End delimiter

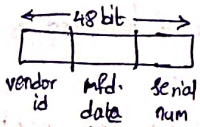
End delimiter may match with data pattern.

Character stuffing
- Add null (\0) as prefix to every null and ED appeared in string

bit stuffing
- If ED is 01111 then add '0' after every 0111.

Physical Addressing:

Logical add: unique globally.
Physical add: unique with Nw.
→ MAC is unique globally and used as physical address.



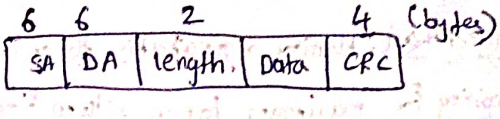
Unicast MAC: LSB bit of 1st byte is 0
Multicast MAC: LSB bit of 1st byte is 1
Broadcast MAC: all bits are 1's.

LAN Technologies

Ethernet (IEEE 802.3)

- Bus topology; CSMA/CD; No acks; Manchester Encoding
- Bw: 10Mbps, 100Mbps (↳ fast ethernet), 1Gbps (↳ gigabit ethernet)

Ethernet Frame Format:



→ PL adds 1 byte SFD (10101011) and 7 bytes preamble 10101010...10
→ Preamble & SFD provides synchronization.
→ Since CSMA/CD is used $T_t \geq 2T_p$

$\Rightarrow \boxed{L \geq 2T_p * Bw} \Rightarrow L \geq 64B$

→ To avoid monopolization, max data = 1500 B

	Min	Max
Data	46	~~64~~ 1500
Frame	~~1500~~ 64	1518

Disadv:

→ not suitable for real time applications
→ not applicable for interactive application ∵ min data size = 46B
→ not good for client server applications (∵ no priorities)

Token Ring (IEEE 802.5)

- Ring topology; token passing; piggy backed acks; Differential manchester

Sender side problems

- orphan pckt problem: sender goes down
- Stray pckt problem: pckts get corrupted and cannot be identified by sender.

Soln: Monitor & Monitor bit (Think)

Destination side Problems:

- Destination Down
- Destination Busy
- Packet corrupted
- Copied.

A	C	
0	0	
1	0	& E=0
1	0	& E=1
1	1	
0	1	→ invalid.

A - available
C - copied
E - Error

→ while retransmitting sender has to reset monitor bit

Token Problems:

a) **Captured token**: a station may hold token for indefinitely long.

soln: impose restriction on max THT

ETP ⇒ T_t = THT

DTP ⇒ T_t = THT − RL

b) **Token Lost**: Monitor regenerates token after waiting for maximum token return time.

max token return time = ring latency + N * THT

c) **Token Corrupted**: monitor identifies this and regenerates a new token after waiting for max token return time.

Monitor Problems:

a) **Monitor goes down**: monitor sends AMP frame as long as it is active. Every station expects AMP frame in regular intervals.

→ If monitor goes down, new monitor is chosen using polling

b) Monitor Hacked

Token ring frame format:

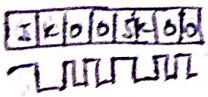
Data frame:

8	7	6	5	4	3	2	1
SD	AC	FC	DA	SA	data	CRC	ED	FS

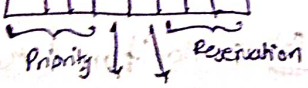
Token frame: (3 bytes)

SD	AC	ED

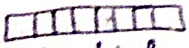
SD:



Access Control:

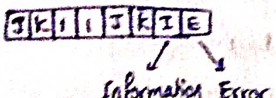


Frame Control:

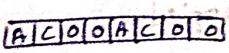


0 0 - data frame (all bits are 0's)
1 1 - control frame (Eg: AMP)

End Delimiter:



Frame status:



Minimum length of token ring:

If all the systems are down, then there is a chance that sent token comes back even before it is fully transmitted.

To prevent this

propagation delay ≥ transmission delay of token

or)

Capacity ≥ size of token.

Switching

Circuit Switching

- For every connection a reservations are made & path is set. Time required for this is called setup time.
- No header is req as path has already been setup
- Data is always sent in order.
- Circuit switching is applied at physical layer.
- Circuit switching is implemented in 2 ways:

Freq. Division Multiplexing
- freq is divided among the connections

Time Division Multiplexing
- time is divided into slots and each connection is given a slot.

Packet Switching:

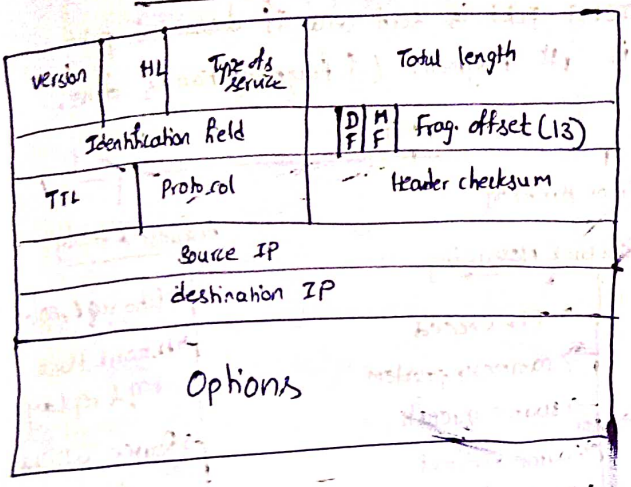
→ No reservation; store & forward is used;
→ packetization is done.
→ Queing delay is present and pckt may be discarded if there is no space in the queue.
→ No setup time is required.
⇒ 2 types of packet switching:

virtual circuit	Datagram
• Connection oriented	• Connection less
• pckts follow same path	• Diff path.
• reservations are made by 1st pckt	• No reservations
• pcks arrive in order	• out of order.
• 1st pckt has global header	• all need global header
• reliable, costly	• not reliable, cheap

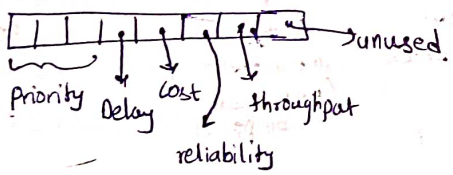
Note:

- Circuit switching may result in wastage of resources if the connection is idle.
- Circuit switching is costlier than pckt switching.
- Circuit switching is more suitable for real time applications

Internet Protocol



- Header length = 4 * (HL field)
- TOS:



- Total length: length including header.
- Identification field: numbering to datagrams helps in ordering received datagrams
- Protocol: to used to discard less imp pckts when there is space.

  ICMP < IGMP < UDP < TCP
- Header checker: divide into 2 byte slots, add and negate.

Options:

(i) Record route: each router put its address

(ii) Source routing < Strict source routing / Loose source routing

(iii) Padding: to make size a multiple of 4

(iv) timestamp: Every router in path adds in_time and out_time.

Fragmentation:

→ Max Transmitable Unit: Max size that can be transmitted (including IP header)

→ Segmentation is done at TL such that frag. will not be need at host. So fragmentation is done only at routers.

→ After fragmentation pckts are routed independently.

→ Max payload at TL = MTU − IP header − TL header

→ original fragment offset = 8 * (fragment offset field)

So payload size at IP must be a multiple of 8 in all the pckts except the last.

→ Reassembling is done only at destination.

Reassembly algo:

- Starting from 1st fragment calculate fragment offset of next pckt and identify next pckt using this calculated value. Continue this until last fragment is met.

MF	offset	
1	0	→ 1st fragment
1	≠0	→ intermediate fragment
0	≠0	→ last fragment
0	0	→ No fragmentation

$$\text{offset of next fragment} = \left(\text{offset of current fragment} + \frac{TL - HL*4}{8}\right)$$

Other Concepts at Network Layer:

Implementation of Broadcasting

Limited Broadcasting:

S.IP = source IP
S.MAC = source MAC
D.IP = All 1's
D.MAC = All 1's

Directed Broadcasting:

S.IP = Source IP
S.MAC = Source MAC
D.IP = NID + (All 1's)
D.MAC = MAC of gateway router.

- once this reaches gateway router then it replace DIP with all 1's and D.MAC with all 1's.

Address Resolution Protocol (used to find MAC given IP)

Receiver on N/w:

ARP req pckt is broadcasted with

S.IP = Source IP
S.MAC = MAC of sender
D.IP = dest. IP
D.MAC = All 1's

Reciever on diff Nlw:

→ In this case sender can never know the MAC of receiver.

→ Here sender just finds ∴ MAC of default router and forwards pckt to the router.

Note: ARP req is broadcast message
ARP reply is unicast message.

Special Address 127 (Loopback address)

→ If we need to check if NIC and os other layers of system we used this address.

→ we can use every address except 127.0.0.0 and 127.255.255.255

→ This pckt will go through all layers except physical layer and comes back to app. layer.

→ This is also used to test working of ~~client~~ server by using different ports.

RARP (Reverse ARP):

→ used to know IP, given its MAC.

→ RARP server is maintained at every Nlw. It contains mapping blw IP add & MAC.

→ RARP req:

S.IP = All 0's D.IP = 255.255.255.255
S.MAC = source MAC D.MAC = All 1's

→ RARP server sends reply.

Disadv: → MAPping is static.
→ Data is not centralized.

BootP (Bootstrap protocol)

→ ~~so~~ same as RARP except that it may not have bootp server at every Nlw. ~~Relay~~ In this case relay agents are used.

- Adv: Data is centralized
  Disadv: Mapping is static.

DHCP:

→ Here we maintain 2 mappings < static / dynamic

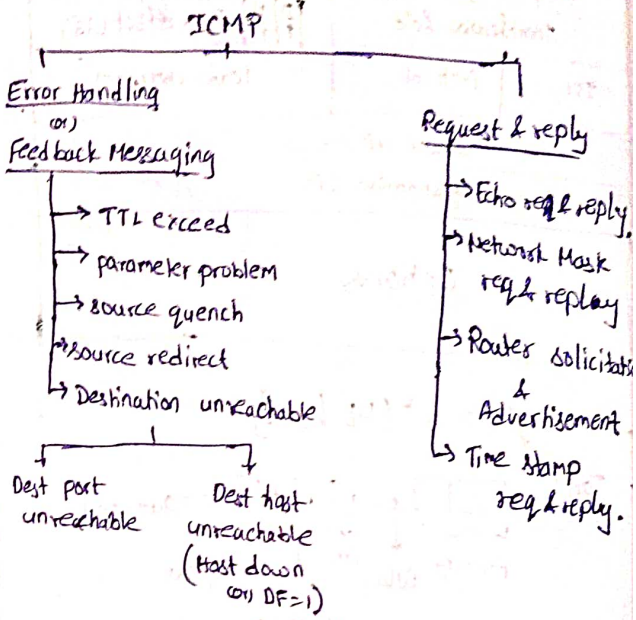
→ IP addresses are dynamically assigned from a pool of address and lease time is given after which renewal request has to be done.

Adv: • Data is centralized
• Mapping is dynamic

Note: to make DHCP compatible with bootp, port of BootP = port of DHCP.

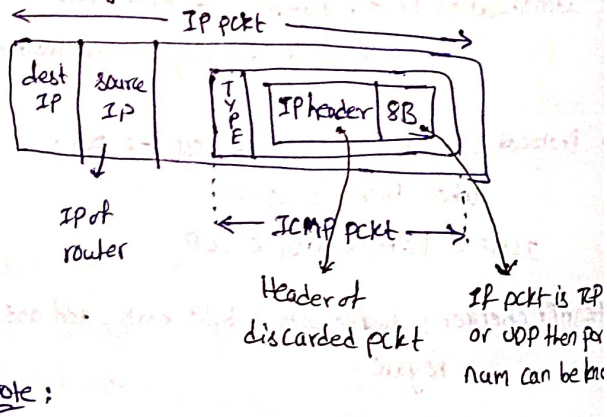
ICMP (Internet Control Message Protocol)

- ICMP is sent only if the discarded pckt is TCP or UDP.
- ICMP pckt is sent only if discarded pckt is 1st fragment (if fragmentation is done.



- ICMP pckt never meets app-layer and TL.

ICMP message format:



Note:
Before any communication starts, host must do below steps:

(i) Getting IP address (RARP, BootP, DHCP)
(ii) Knowing default router (router adv./solicitation)
(iii) Knowing Nlw mask (Nlw mask req & reply)

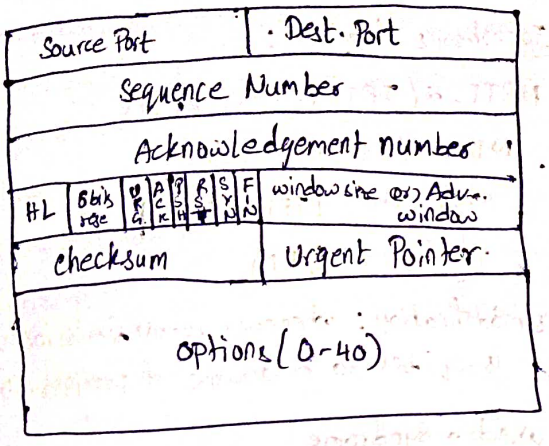
Applications of ICMP:

(i) Trace route
(ii) PMTUD (Path MTU Discovery)

Refer notes for Routing

TCP

- TCP ensures end to end communication



Port number:

0-1023: well known

1024-49151: reserved

49152-65,535: Public usage.

Seq number: Contains number given to 1st byte in the segment.

- TCP is a byte stream protocol.

Ack number: contains byte num of next expected byte.

no of bytes in segment (n) = $TL_{IP} - HL_{IP} - HL_{TCP}$

⇒ Ack num = n + seq num

Note:

- seq num doesn't always start with 0. Instead we start with some random number (think why)

wrap around time: It is time taken to usecp all seq number.

life time: It is max amount of time for which pckt can be in network.

- to avoid conflicts

  wrap around time > life time.

- For given ~~no of bits~~ bandwidth, if no of bits in seq num field are not sufficient to satisfy above condition, then we take extra bits from options field. this option is called timestamp.

Header length: scaled down by 4

Flags:

PSH: pushes data as soon as it is received without buffering. Useful for interactive applications

URG:

RST: Used when there is something wrong in connection.

Ex: receiver receives unexpected seq num.

URG: when this flag is set, the data is sent faster and it will reach dest even before the pckts that are sent earlier.

For this purpose we set priority to 7 in type of service field. (∵ routers can't look into TCP header).

URG pointer: Used when only some part of data is urgent.

last byte of urgent data = seq num + urgent pointer.

SYN: used to synchronize sequence numbers.

Ack: If this is set then only the ack number field is considered.

If pure ack is to be sent then, only header will be sent.

FIN: used when we need to terminate the connection

Note: Refer connection establishment, ~~data~~ connection termination in notes. Also refer state transition diagram.

- TCP is full duplex connection.
- At the time of connection establishment information like initial seq num, MSS, window size are exchanged. (MSS → payload in TL)
- Window sizes are set based on MSS shared.

Note:

- pure ack doesn't use seq num.
- FIN pckt uses a seq num
- If no data is received, still we send ack for old data

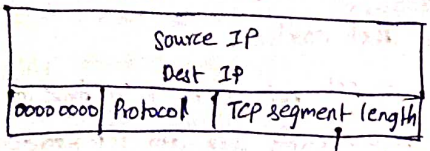
SYN	Ack	
1	0	→ 1st pckt
1	1	→ 2nd pckt
0	1	→ Ack present in header
0	0	→ Invalid

Window size / Adv window:

- used for flow control.
- receiver sends available window size.
- If window size received by sender 0, then sender starts persistant timer and waits until receiver notifies or the timer finisher. If timer finishes sender sends 1 byte of data. If sender gets reply then communication is continue otherwise persistant time is started again.
- window size choosen by sender = min{receiver window, congestion window}
- window size field has only 16 bits, if more than that ~~is~~ 2^{16} is available then window ~~extension~~ size extension option is used.

Checksum:

- checksum is computed on TCP header & pseudo IP header



→ total length$_{IP}$ − HL$_{IP}$

Options:

- timestamp
- window size extension
- Parameter Negotiation Eg: MSS
- padding

Retransmissions in TCP:

- In TCP, if ack num 'x' is received then sender assume all the data before x is correctly received.
- Retransmission is done in 2 cases
  - (i) receiving 3 dup acks (excluding original ack) within TO timer. ---- low congestion. This is known as early retransmission
  - (ii) Time out timer. ---- high congestion.

Congestion Control:

sender window = $\min\{w_c, w_R\}$

w_c - congestion window w_R - Receiver window.

working:

- slow start phase until threshold is reached. Here congestion window size is doubled.
- Congestion avoidance phase after threshold. Here congestion window is increamented by 1. initial threshod ← $w_R/2$.
- Congestion detected due to TO times ⇒ threshold = $\lceil w_c/2 \rceil$ & enter slow start phase.
- Congestion detected due to 3 dup acks ⇒ threshold ← $\lceil w_c/2 \rceil$ & enter congestion avoidance phase.

TCP time Management:

(i) time wait timer: (think of its purpose)

time wait timer = 2 * life time

(ii) keep alive timer

(iii) persistent timer

(iv) Ack time

(v) Time out timer:

- static TO timer leads to problems.

Dynamic TO timer is given by below algorithms

(i) Basic algorithm:

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

$$TO = 2 * RTT$$

(ii) Jackobson's Algorithm:

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

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

where $AD = |IRTT - ARTT|$

$$TO = RTT + (4 * D)$$

Karn's modification: whenever retransimission is req set TO to double of previous TO.

Silly window Syndrome:

- occurs in 3 cases:
  - (i) Receiver advertises 0 window size. But this problem is tempory. If this continues for long time RST flag is used.
  - (ii) occurs when sender transmits less amount of data.

    soln: Nagle's algo:
    - transmits once for each RTT or transmit in advance if data sufficient for 1 MSS is available.
  - (iii) receiver accepts only 1 byte.

    Clarck's soln: don't advertise until half of window is available (or) 1 MSS is available.

Traffic Shaping:

- Another congestion control that controls rate at which pckts enter N/w.
- During connection establishment sender and receiver negotiates traffic pattern.

(i) Leaky Bucket:

- Data comes into Q at any rate but comes out at const. rate.
- If Q is full data is discarded. (think disadv)

(ii) Token Bucket:

- Each pckts gets a token before it goes on the N/w.
- o/p rate = token filling rate.
- If capacity is full then no more token will be added.
- If capacity is c and token filling rate is r then max no of tokens that can be sent in time 't' is

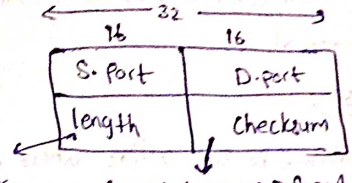
$$\frac{c + rt}{t}$$

- In leaky bucket pckts are discarded when bucket is full but here tokens are discarded.

UDP (User Datagram Protocol)

- Used in application that need single req & reply.
  Eg: DNS, BootP, DHCP, Dist. vector routing etc.
- broad casting & multicasting application.
- used in applications that req speed over reliability
  Eg: multimedia, gaming etc.
- If AL needs to set any priorities then UDP has to communicate this to NL. TCP should also do this.
- ICMP pckts received are informed to AL through UDP.

UDP Header:



- Since UDP is not ~~use~~ reliable, checksum is not used in general. In that case it is set to 0.

Checksum is stored in 1's comp form

000...00 ⇒ Checksum not used
111...11 ⇒ Checksum used.

Hardware in Networking

~~Hardware in Networking~~

- 2 types of channels < baseband (no multiplexing)
  broadband (multiplexing)
- 10 base T (10MBPS, no multiplexing, 100m)
  10 base 2 (" " 200m)
  10 base 5 (" " 500m)

Attenuation is possible here.

Repeater:

- used to increase length of LAN by connecting 2 lan segments of same type.
- Repeater takes dead signal & regenerates it.
- Runs at PL and collision domain remains same.

Hub:

- Hub is a multiport repeater. An n-port hub connects n stations.
- A pckt to be sent is transmitted to every station. ∴ traffic is high.
- Runs at PL and collision domain remains same.

Adv: cheap

Bridge:

- Used to connects 2 lan segments. ~~of~~ Lan segment can even be different types ~~by~~ but MTU must be same since bridge cannot do fragmentation
- Operates at PL & DLL (can see MAC)
- Collision domain is reduced.
- Every bridge has forwarding table. Based on how this table is built we have 2 types of bridges, (i) static (ii) Dynamic.
- Dynamic bridge never knows MAC of a station until it transmits.
- Bridge is capable of filtering, forwarding, store & forward, flooding.

Problems:

- If ~~a~~ connection forms cycle then ~~the~~ pckts may go into infinite loop.

Spanning tree algo:

- Choose bridge with least id as root bridge.
- For each bridge, choose root port.
- For every LAN, choose designated bridge and make corresponding port as designated port.
- Mark root port and designate ports as forwarding ports and block remaining.
- Bridge Data unit Protocol (BDUP) is implemented to sort out this tree.

Switch:

- A n-port switch can connect n station.
- Operates at PL & DLL.
- Switch ~~is designed~~ allows more than one communication at a time.
  So collision within switch = 0.
  traffic is less.
- Collision domain is reduced.

Disadv: costly.

Router:

- Connects 2 networks of same type.
- operates at PL, DLL, NL.
- filters broadcasted pckts.
- collision domain & broadcasted domain are reduced.

- Every interface of router has an IP address. The address is taken from the N/w the interface is connected to. However if 2 routers are connected then we need a /30 CIDR block of size 4.

GATEWAYS:

- capable of connecting N/ws of 2 diff types.
- operates at: PL, DLL, NL, TL, AL
- ~~appli~~ reduces collision domain & broadcasting domain

Uses:

- protocol convertor
- proxy.
- NAT server.
- Firewall
- Deep packet inspection
- Buffers management (think of its use)

Application Layer Protocols

DNS (Domain Name Service):

- port: 53
- given a domain, it gives IP address.
- DNS also does load balancing
- Data of DNS is distributed.
- There are 13 root DNS servers.
- Local DNS server contancts root DNS server in 2 ways (i) iterative (ii) Recursive.
- DNS uses UDP.

HTTP (Hyper Text Transfer Protocol)

- port: 80
- used to get web page.
- Uses TCP for reliability. No any inbuilt mechanism for reliability.
- Inband protocol
- stateless protocol. (HTTP has no any inbuilt security mechanism)

  It rather uses cookies which are stored at the userend.
- HTTP 1.0 use non-persistent connection
  no of connections = no of objects + 1
- HTTP 1.1 uses persistent connection.
  Since connection is held for long time, congestion window grows and high B/w is available.
- Suitable for when objects are small and many in number.

methods: Head, Get, Post, put, delete, trace, options, connect (think of their functionalities)

FTP (File Transfer Protocol)

- port: 21 (commands), 20 (data); Uses TCP
- outband protocol
- Control connection (port 21) is persistent
- Data connection (port 20) is non-persistent.
- stateful Protocol

SMTP & POP:

- FTP req stations to be online where as SMTP doesn't
- Mail Client (MC) pushes mail into Mail Transfer Agent (MTA) using SMTP.
- Mail client (MC) at the other ends takes mail from MTA using POP.
- SMTP & POP uses TCP
- To send data other than non-text (we need to convert it text format and send it. This will be converted back to non-text at receivers end.
- inband protocol.