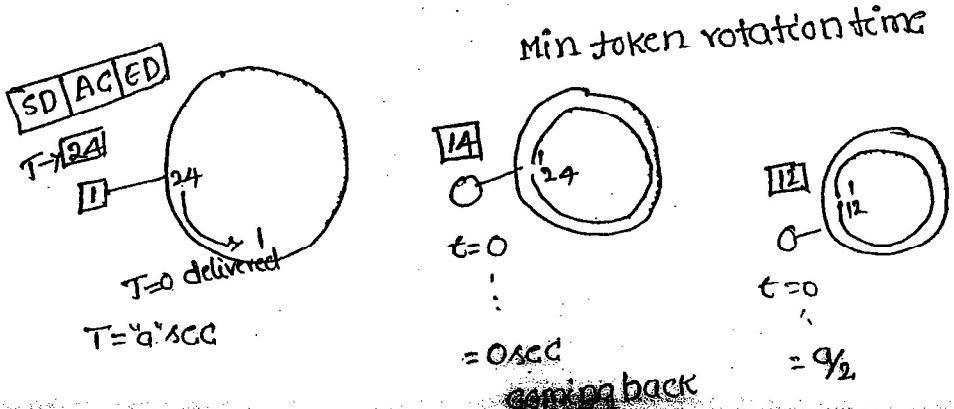
Made Easy Class Computer Network Vol 2

Copyright©Theorypoint.com

Modes of operation:

* Transmission mode
* Listen mode
* Receiving mode
* Bypass mode.

Calculation of minimum size of token ring:



T_{Prop} = "a" sec = trans. delay

$T_{Prop} = T_{Trans} \Rightarrow \frac{T_{prop}}{T_{trans}} = 1$

$\frac{T_{Prop}}{T_{trans}} = 1 \Rightarrow$ min.

$> 1 \Rightarrow$ Max

$< 1 \Rightarrow$ overlap.

* Token rings used for big networks.

* Token ring: **propagation delay = Transmission delay**

$$\frac{d}{V} = \frac{L}{B}$$

Eg: $V = 2 \times 10^8$ m/sec

$L = 24$ bits

$B = 4$ mbps

$d = ?$

$\Rightarrow \frac{d}{V} = \frac{L}{B}$

$d = \frac{L}{B} * V$

$= \frac{24 \times 2 \times 10^8}{4 \times 10^6}$

$d = 1.25$ km.

Eg: if Bandwidth of ring is 10 mbps, & frame size 200 bits velocity = 2×10^8 m/sec find min size token ring

$d = \frac{L}{B} * V$

$= \frac{200 \times 2 \times 10^8}{10 \times 10^6}$

$= 4000$ km.

=

Ring latency = Min TRT = Propagation delay in the ring + no. of active stations * Delay at each station

Propgation delay = $\frac{d}{v} + mb$

($\frac{d}{v}$ → sec, mb → bits)

Bid delay $b = 1/B$

d = Total length of ring

b = bit delay at each station

V = Velocity propagation

R = bandwidth of ring

L = latency.

$$L = \frac{d}{V} + \frac{mb}{R} \text{ sec}$$

$$RL = \frac{dR}{V} + mb \text{ bits}$$

Various Token re-insertion strategies:

Delayed Token strategies	Early Token strategy.
* Token is released after getting entire data packet.	* Token is released after data is transferred.
* Efficiency is low	* Efficiency is high
* Reliability is high	* Reliability is low.
* it is used under load conditions.	* it is used under high load conditions.
cycle time = $(a+b+c+d)$ sec	cycle time = $(a+c+d)$ sec

where a → data transmission

b → Ring latency

c → Token Transmission time.

d → propagation delay b/w station.

Problems:

① $d = 2500$ mts

$V = \frac{60}{100} \times 3 \times 10^8$ m/sec

$B = 10$ mbps

$L = ?$

RTT = Transmission delay

$2 \times \frac{d}{V} = L/B$

$L = 2 \times \frac{d}{V} \times B$

$= \frac{2 \times 2500 \times 10 \times 10^6}{1.8 \times 10^8}$

$= 277$ bits.

② $B = 1$ Gbps

$d = 1$ km

$V = 200{,}000$ km/sec

$L = ?$

$\frac{L}{B} = 2 * \frac{d}{V}$

$L = 2 * \frac{1}{200{,}000} \times 1 \times 10^9$

$= 10{,}000$ bits or 1250 bytes.

③ $B = 10$ Mbps

Propagation delay $= \frac{d}{V} = 225$ bit times

Bit delay $= b = 1/B = \frac{1}{10 \times 10^6} = 0.1$ μsec

Transmission can be considered as either 1 bit delay or 0.1 μsec

at $t=0$ A & B started their communication.

At $t = 225/2$ there is a collision

At $t = 225$ the A & B will come to know about the collision through propagation

④

⑤

Assume 'A' started producing jam signal.

∴ At $t = 273\ (225+48)$, A finishes producing jam signals

⇓

Jam.

④

⑤ $B = 10$ Mbps

Slot time = 51.2 μsec

$L = 512$ bytes

No. of slots = 1.716

$$\eta = \frac{T.P}{T.P + C.P + I.P} \qquad (I.P = 0)$$

Transmission time = L/B

$$= \frac{512 \times 8}{10 \times 10^6}$$

$$= 40 \text{ msec}$$

C.P = no. of slots * slot time

$$= 1.716 \times 51.2$$

$$= 87.8 \text{ μsec}$$

$$\eta = \frac{40 \times 10^{-3}}{40 \times 10^{-3} + 87.8 \times 10^{-6}}$$

$$=$$

≈ 0.1 μsec

.ik.com

(6) B = 10 Mbps

d = 2.5 km

V = 2.3×10^8 m/sec

L = 128 bytes ⇒ 98+30

overhead = 30 bytes

$$\eta = \frac{1}{1+6.44a} = \frac{1}{1+6.44(0.10)} \quad \because a = \frac{T_{prop}}{T_{tran}} = \frac{1.086 \times 10^{-5}}{1.024 \times 10^{-4}} = 0.10$$

$$= 57\%$$

$$T_{prop} = \frac{d}{V} = \frac{2.5\ km}{2.3 \times 10^8} = \frac{2.5\ km}{2.3 \times 10^5\ km/sec} = 1.086 \times 10^{-5}$$

$$T_{trans} = \frac{L}{B} = \frac{128 \times 8}{10 \times 10^6} = 1.024 \times 10^{-4}$$

(7) B = 4 Mbps

Token holding time = 10 msec

4 bits — 1 sec — 4×10^6

? — 10 msec — ?

longest frame = $4 \times 10^6 \times 10^{-3}$

= 4 Kbps.

(8) R = 4 Mbps

m = 20

d = 20 × 100

b = 2.5 bits

V = 2×10^8 m/sec

$$RL = \frac{dR}{V} + mb$$

$$= \frac{20 \times 100 \times 4 \times 10^6}{2 \times 10^8} + 20 \times 2.5$$

= 90 bits

R = 16 Mbps

m = 80

$$RL = \frac{dR}{V} + mb$$

$$= \frac{80 \times 100 \times 16 \times 10^6}{2 \times 10^8} + 80 \times 2.5$$

= 840 bits

(10)

(11)

A. Ean

B. Sek

(10) Station = m

$d = m \times 100$ $\quad V = 2 \times 10^8$ m/sec $\quad$ Transfer time $= \frac{L}{B} = \frac{1250 \times 8}{25 \times 10^6}$

$B = 8$ bits

$= 400$ μsec

$L = 1250$ bytes

$R = 25$ Mbps

$RL = \frac{d}{V} + \frac{mb}{R}$ sec

$\frac{RL}{Tr.time} = 1$

$= \frac{m \times 200}{2 \times 10^8} + \frac{m \times 8}{25 \times 10^6}$

$= 3.33 \times 10^{-3} m = 1$

$= m\left(\frac{200}{2 \times 10^8} + \frac{8}{25 \times 10^6}\right)$

$m = 300$

$= m$

1 sec $\rightarrow 25 \times 10^6$

? $\quad 1250 \times 8 \Rightarrow \frac{1250 \times 8}{25 \times 10^6} =$

(11) $m = 32$

$L = 1000$ bit packet

$B = 10$ Mbps

latency/adapter = 2.5 bit

$d = 50$ m

$V = 2 \times 10^8$ m/sec

Data transmission $= a = \frac{L_D/B}{}$

$= \frac{1000}{10 \times 10^6} = 10^{-4}$

$b = RL = \frac{d}{V} + \frac{md}{R}$ sec

$= \frac{32 \times 50}{2 \times 10^8} + \frac{32 \times 2.5}{10 \times 10^6}$

$= 880 \times 10^{-8}$

A. Early Token $= (a + c + d)$ sec

$= 10^{-4} + 24 \times 10^{-7} + 25 \times 10^{-8}$

$= 10$ msec

Token transmission $= c = \frac{L_T}{B} = \frac{24}{10 \times 10^6}$

$= 24 \times 10^{-7}$

Propagation delay $= d = \frac{d}{V}$

$= \frac{50}{2 \times 10^8}$

$= 25 \times 10^{-8}$

B. Delay token $= (a + b + c + d)$ sec

$= 10^{-4} + 880 \times 10^{-8} + 24 \times 10^{-7} + 25 \times 10^{-8}$

$=$ 11 msec

9) if is heavily loaded, we suppose to use early token strategy

$C.T = a + c + d$ $\quad a = \frac{LD}{B} = \frac{256}{10\times10^6} = 2.56\ \mu sec$

$d = 1\ km$

$B = 10\ Mbps$

$L = 256\ bit$

$c = \frac{LT}{B} = \frac{8}{10\times10^6} = 0.8\ \mu sec$

$d = \frac{200\ m/\mu sec}{2\times10^8\ m/sec} = 2.81\ \mu sec$

bit delay = 8 bits

Propagation speed = 200 m/μsec

$C.T = 26.5\ \mu sec$

$26.5\ \mu sec \Rightarrow 224$

$1\ sec = \frac{224}{26.5\times10^{-6}} = 8.5\ Mbps$

$\eta = \frac{8.5}{10}\times100 = 85\%$

(15)

10.

3) $d = 200\ km$

$B = R = 100\ Mbps$

$V = 200{,}000$

$L = 1024\ bytes$

$a = \frac{LD}{B} = \frac{1024\times8}{100\times10^6} = 81.24\ \mu sec$

$b = RL = \frac{d}{v} + \frac{mb}{R}\ sec \quad \left(\because \frac{mb}{R} = 0\right)$

$= \frac{200\ km}{200{,}000} = 1\ msec$

$c = \frac{L_{TOKEN}}{B} = \frac{24}{100\times10^6} = 0.24\ \mu sec$

$d \Rightarrow$ not given ignore

$C.T = a + b + c$

$= 81.24\ \mu sec + 1\ msec + 0.24\ \mu sec$

$= 1.08\ msec$

$1.08\ msec - 1024\times8\ bit$

$1\ sec - ?$

$= \frac{1024\times8}{1.08\times10^{-3}} = 7.6\ mbps$

$\eta = \frac{7.6\ mbps}{100\ Mbps}\times100 = 7.6\%$

(14). propagation speed = 200 m/μsec

B = 1 Mbps

bit delay = $1/B = \frac{1}{1 \times 10^6}$ = 1 μsec

1 μsec ⟹ 200 mts

B = 40 Mbps Bit delay = $1/B = \frac{1}{40 \times 10^6}$ = 0.025 μsec

1 μsec – 200 mts

0.025 μsec – ?

⇒ 200 × 0.025

= 5 mts.

(15) B = 5 Mbps

Propagation speed = 200 m/μsec

bit delay = $\frac{1}{5 \times 10^6}$ = 0.2 μsec

1 μsec – 200 mts

0.2 μsec – ?

⇒ 200 × 0.2

⇒ 40 mts.

=

c_4

$*$ c

$*$ c

$*$ c

c_1

$*$ Π

$\log$

$\mathcal{H}$

$*$ $\equiv$

$\mathcal{I}$

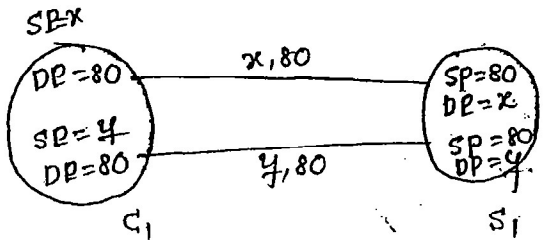
$*$ Ω

Transmission Control Protocol.

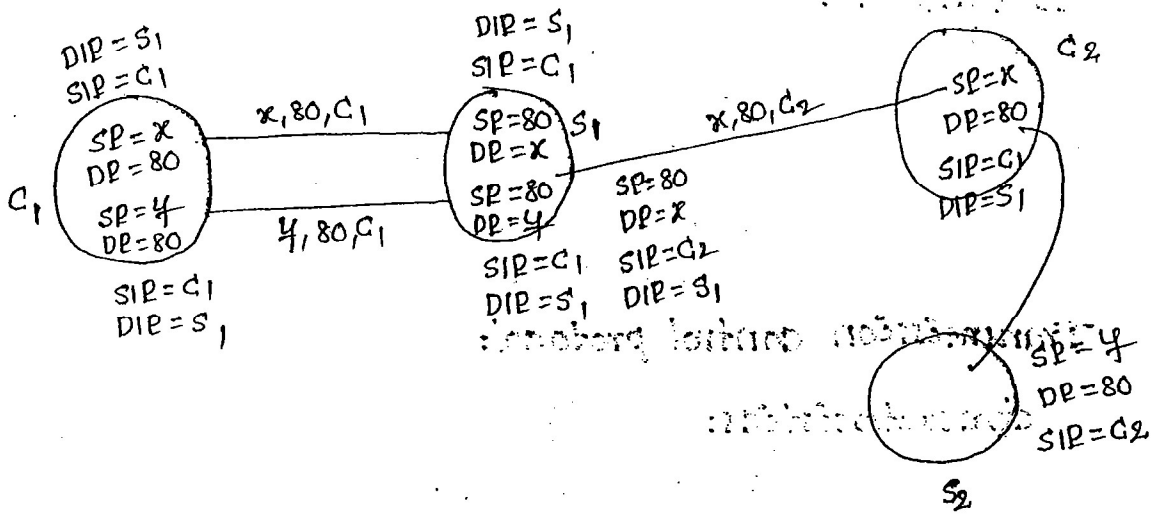
why we call network as TCP/IP networks

(or)

Relationship between TCP & IP.



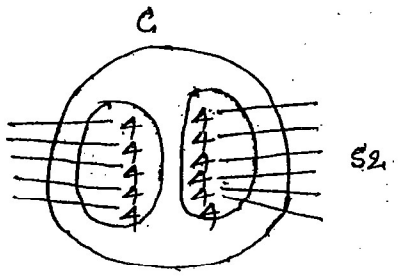
* Ctrl + N => another connection
* while establishing a new connection source port & destination Port is addressed.
* server can handle more no.of clients, so introducing a new client involves both source port and destination port.



* In order to differentiate the destination IP address either for server1 (or) server2, the DIP addressed. where "80" represent the http and SIP represents the user's own address.
* The source port & Destination ports are handled by TCP Protocol in the Transport layer => 2 ports parameters.
* The source IP & Destination IP are handled by IP Protocol in the

4 parameters { source port, Destination port, source IP, Destination IP.

Socket:



* socket is a logical component which groups a set of parameters for communication.

$5 \times 4 = 20$ $\quad$ $6 \times 4 = 24$

$5 \times 2 = 10 + 2 = 12$ $\quad$ $6 \times 2 = 12 + 2 = 14$

Advantages:

* Resource utilization
* Maintainance & Administration

→ Allows certain num-of connections for socket.

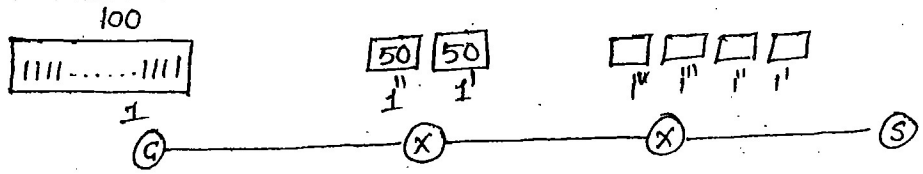
Transmission control protocol:

characteristics:

* it is reliable, byte oriented, point-to-point, transport layer protocol.

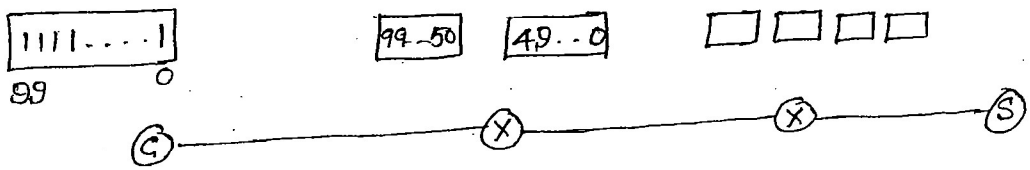
connection-oriented.

* Message Oriented (UDP)
* Packet Oriented (SWP)
* Byte oriented (TCP)
* Bit-Oriented (HDLC).



Burden on intermediate routers.

* The bits are splitted by half to router, as it can handle only certain set, so it have a problem to exactly split, (if not, there is a chance of missing a packet)



* The above problem can be overcome by not splitting the bits but just only dividing the ~~station~~ stream of bits and transfer so it has no burden on the intermediate routers.

* since the bits are transferred as byte (or) a stream, the TCP is considered as "stream oriented."

* TCP uses "cumulative acknowledgment".

* The connections are "full duplex". Therefore it is having two half-duplex connections.

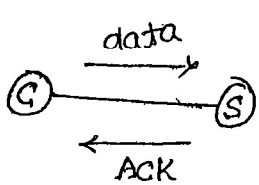


fig: full duplex

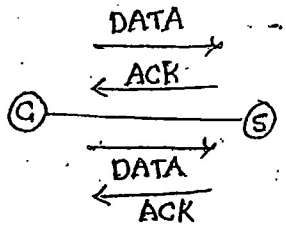
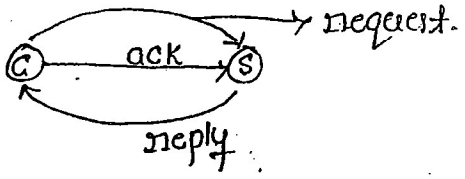


fig: Half duplex.

* TCP, uses, sliding window protocol (SWP) for its flow control. Therefore each TCP connection has 4 windows.

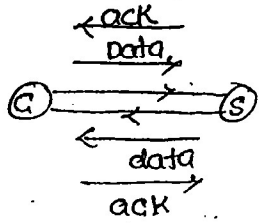
* TCP connection are having 3 phases.

1. connection establishment phase : /(negotiation phase)

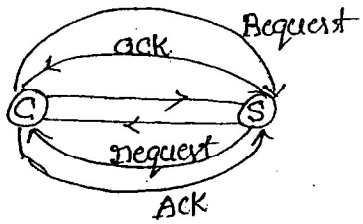


* it is a single step process.

2. Data transmission phase



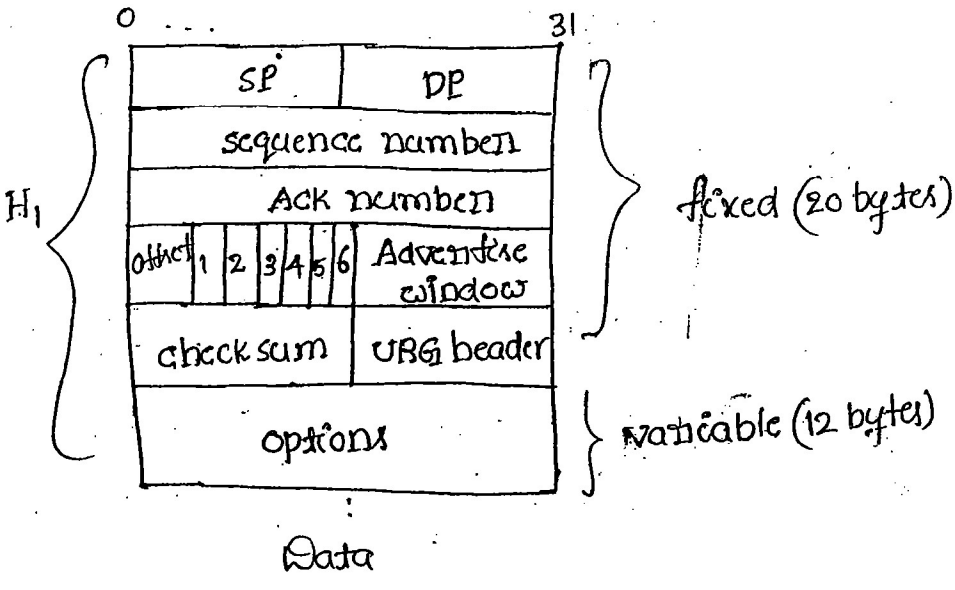
3. connection termination phase.



* it is not single step process eventhough it requested for connection termination. The termination must done in both the systems (not only one)

TCP operations:

TCP header:

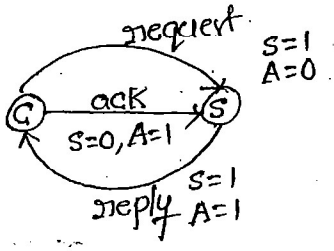


Flags:

1. SYN & ACK
2. FIN
3. RST
4. PSH
5. URG

1. SYN & ACK: SYN & ACK flags are used in connection establ

Phase of different request and reply packets



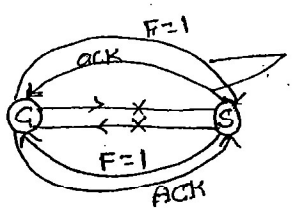
S=1 A=0 => Request

S=1 A=1 => Reply

S=0 A=1 => ACK

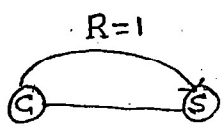
S=0 A=0 => Data

2. FIN : it is used in connection termination phase.



if received ack closed

3. RST Flag: it is used to reset the connections.

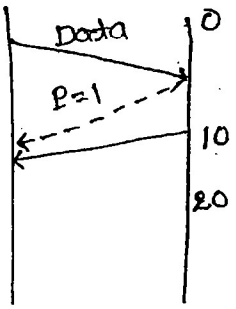


* New connection is establish
just refreshing the windo

* while transfering the data if any problem arises, then the complete connection is cancelled and a new connection is made, so it is not suitable for every time to have new connection. so we use RST. to reset.

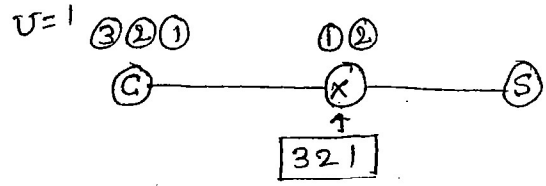
4. PSH flag:

* it is used for high priority packets to push the packet to upper layer without waiting for time interval.



if the data is transferred and of high priority it needs a fast ack from sender. so by using PSH flag sender sends the ack fastly as soon as data reaches without waiting for the time interval it have.

5. URG: it is used take care of "out of band rate"



sequential order

Inband: 1,2,3

outband: ③ 2 1

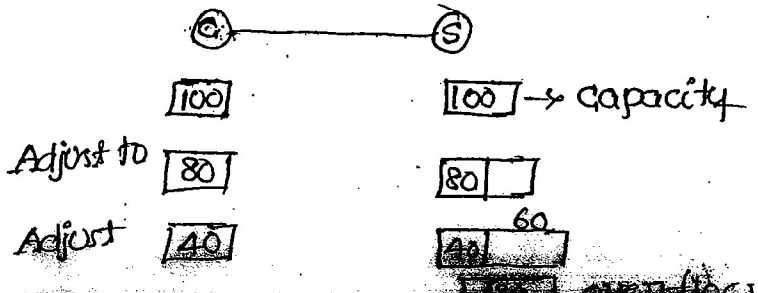
⇓

using URG Pkt.

* URGENT Pointer indicates the amount of the data that is important in the packet.
* it is valid only if URG=1
* if the packets ①②&③ sent to sender by representing URG=1 to ③ Packet then instead of 1 and 2, ③ packet is reached firstly, it represents an URGENT Packet.

Advertise window:

* it is used to implement flow control



* if the server have no empty space, it can't take further data. so it must represents that it has no empty space, but it is difficult to send to each and every client. so we can solve this problem as follows.

* At t=0, client sends the packet and sender have no empty space so simply discard it. it checks for all the time intervals.

* if suppose server have empty space, at particular time interval, then it can accept the data packet.

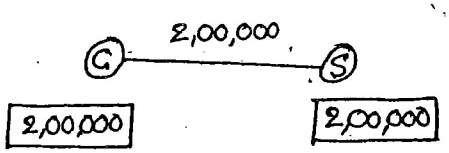
Silly window syndrome (SWS):

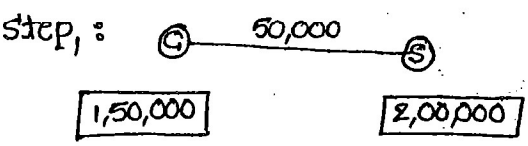
* when SWS occurs efficiency is "0".

* There are three reasons for silly window syndrome.
  - → when server announces it's empty space is "0"
  - → when client is able to generate only one byte at a time
  - → when server consumes only one byte at a time.

* Always ensure to transfer only one byte among all the bytes of data inorder to reduce virus.

* Always server needs to consume only one byte transfer. Possible to reduce the SWS value, so that efficiency increase





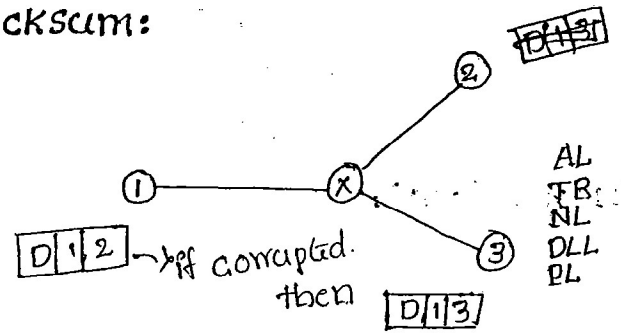
* There is a chance of sending all the 1,50,000 bits, at both the sides but we cannot transfer them because only 2^{16} bits

* if empty space in the server is more than 2 then use scale factor in the - "option" field.

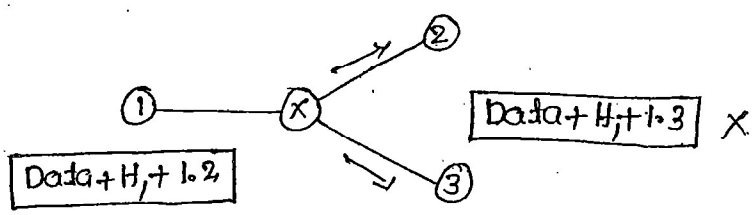
eg: if empty space is 1,50,000 advertise windows = 50,000
& scale factor = 3.

eg: if empty space = 1,00,000 & advertise window = 50,000
then scale factor = 2

Checksum:



* The transport layer transfers the corrupted data to the application layer. Then application layer recognizes as a corrupted bit.
* so inorder not to have burden on application layer, before a packet reaches application layer, it gets corrected at the Transport layer by including the concept of "checksum".
* checksum includes "data + H_1 + pseudo-header" in its calculation.



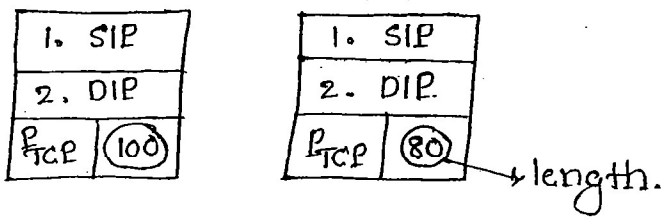
* calculate the checksum $\bar{c}(\text{Data} + H_1 + (1.2))$ at the sender side and send the checksum. at the receiver the check sum is again calculated. $[\text{Data} + H_1 + (1.3)]$ → corrupted, so discard it by transport layer.

* Pseudo-header is used to check whether data packet has been receive by the correct destination or not.

* it is prepared at the source and included in checksum calculations.

* once packet is received by the destination again it is prepared by the destination with destination values.

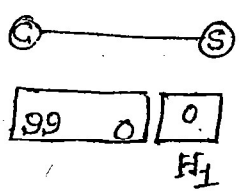
* if incoming checksum is similar to calculated checksum, then packet is consumed, else it is discarded.



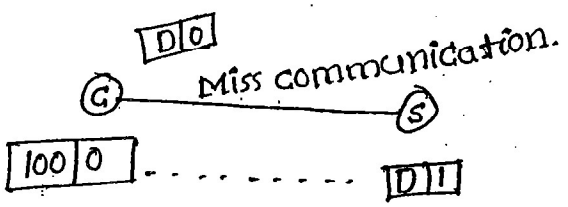
* sequence no. for the packet is first databyte sequence num. in the packet.

characteristics for Sequence num & Ack:

1. sequence no. for the packet is first databyte sequence num. in the packet.



2.

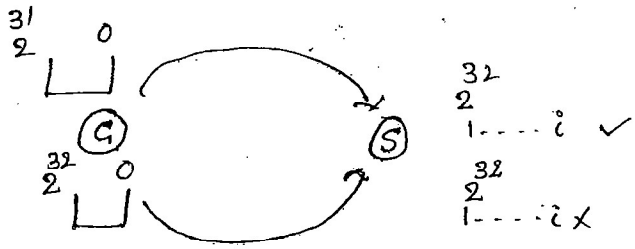


TCP uses random initial sequence number.

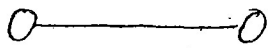
* The sequence number always lies start with 0 it selects a random number among the set of $0 - 2^{32}$ and then sends the random number. if corrupted random num generates simply discard it.

3. Get randomly sequence numbers:

if two different packets of the same sequence num are generated simultaneously then the server thinks that one is the duplication of another packet and just discard one of the packet, which leads to a loss of packet which is different from the other packet.



* In order to overcome this problem, increase the value from 2^{32} to 2^{64}.

Eg: 

1 sec = 1.536×10^6 bit ⇒ 1st pkt

1 sec = $\frac{1.536 \times 10^6}{8}$ bytes ⇒ 2nd packet

1 sec = $\frac{1.536 \times 10^6}{8}$ sequence no of 2nd packet

1.536 Mbps – 64 bits

10 Mbps – 57 min

100 Mbps – 6 min

:

1.2 Gbps – 28 sec.

Eg: consider bandwidth of link = 100 mbps

sequence no. field = 24 bits

Find the "wrap around" of sequence numbers

↳ 1 sec = 100×10^6 bits

= $\frac{100 \times 10^6}{8}$ bytes

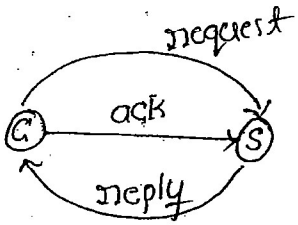
1 sec = $\frac{100 \times 10^6}{8}$ sequence num's

? 2^{24}

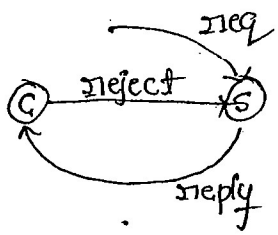
⇒ $\frac{2^{24} \times 8}{100 \times 10^6}$ = 1.3 sec.

Applications:

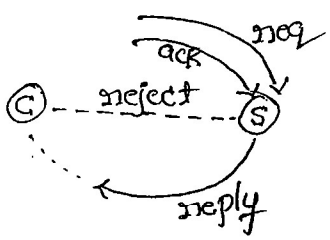
A.



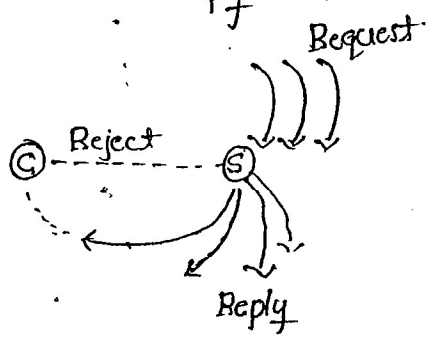
B.



C:



D.

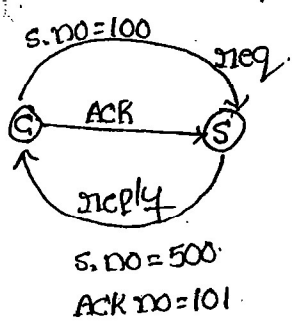


Denial of service attack:

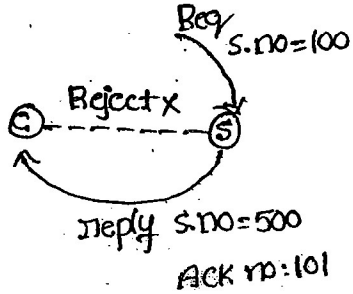
* server denies the client to server for a certain time inter

To overcome above problem we use "sequence number"

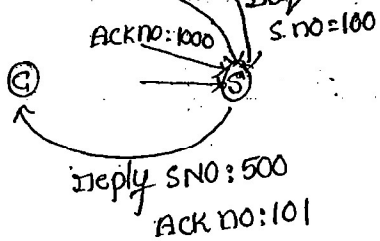
A.



B.



c.

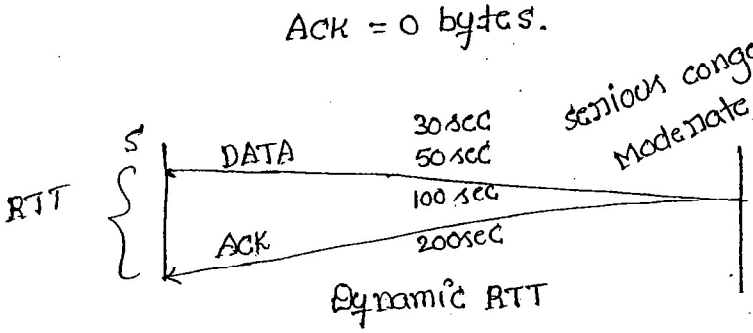


1.

Hacker can't know the random num generated by server so he guess simply rejected. if guess = exact no then the server cannot generate huge ack so it discard to send ack

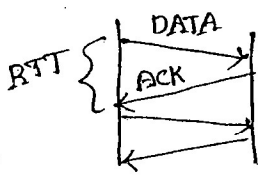
syn } 1 byte
FIN }

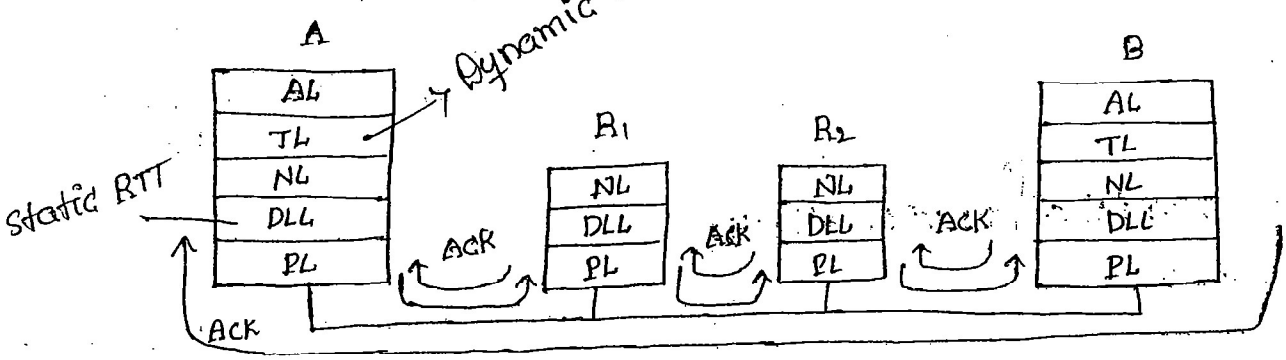
Ack = 0 bytes.

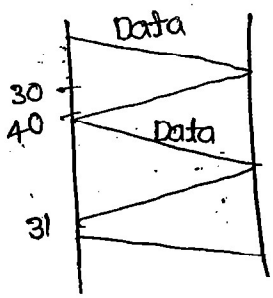


(3)

Unpredictable Timing
⇓
End-to-end
Link-to-link
⇓
Predictable Timing.







2. I

For the calculation of RTT we have certain Algorithms.

1. Basic Algorithm:

(IRTT) Initial Round Trip Time = 30 sec

(NRTT) New Round Trip Time = 40 sec

$\alpha = 0.9$

2. Estimated RTT = α IRTT + $(1-\alpha)$ NRTT

$= 0.9 \times 30 + (1-0.9)\,40$

$= 31$ sec

Time out (TO) = $2 \times$ ERTT

$= 2 \times 31$

$= 62$ sec

③ IRTT = 31 sec

NRTT = 50 sec

ERTT = α IRTT + $(1-\alpha)$ NRTT

$= 0.9 \times 31 + (1-0.9)\,50$

$= 32.9$ sec

Time out = $2 \times$ ERTT

$= 2 \times 32.9$

$= 65.8$ sec

④ IRTT = 32.9 sec

NRTT = 45 sec

ERTT = α IRTT + $(1-\alpha)$ NRTT

$= 0.9 \times 32.9 + (1-0.9)\,45$

$= 34.11$

Time out = $2 \times$ ERTT

$= 2 \times 34.11$

$= 68.22$ sec.

2. Jacobson's Algorithm:

IRTT = 30 sec

NRTT = 40 sec

$\alpha = 0.9$

initial deviation (D_i) = 5

② New Deviation $(D_N) = |IRTT - NRTT|$

$= |30-40|$

$= 10$

Estimate deviation $(D_E) = \alpha D_i + (1-\alpha) D_N$

$= 0.9 \times 5 + (1-0.9) \times 10$

$= 5.5.$

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

$= 0.9 \times 30 + (1-0.9) \times 40$

$= 31$

Time out $= 4 * D_E + ERTT$

$= 4 \times 5.5 + 31$

$= 53.$

③ $IRTT = 31$

$NRTT = 50$

$\alpha = 0.9$

$D_i = 5.5$

$D_{New} = |31-50|$

$= 19$

$D_E = 0.9 \times 5.5 + (1-0.9) \times 19$

$= 6.2$

$ERTT = 0.9 \times 31 + 0.1 \times 50$

$= 32.9$

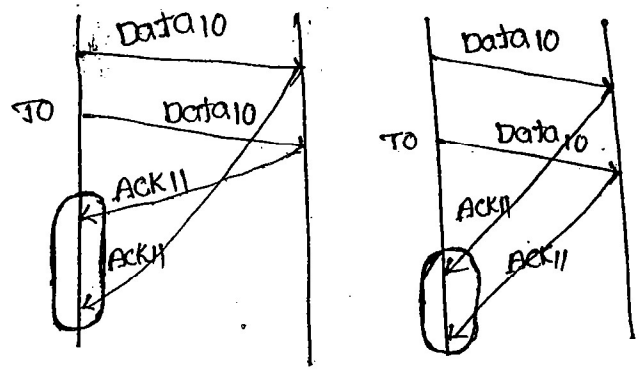
$TO = 4 \times D_E + ERTT$

$= 4 \times 6.2 + 32.9$

$= 57.$

* "Time out" is high under Basic algorithm then the Jacobson Algorithm.

Karn's Algorithm:



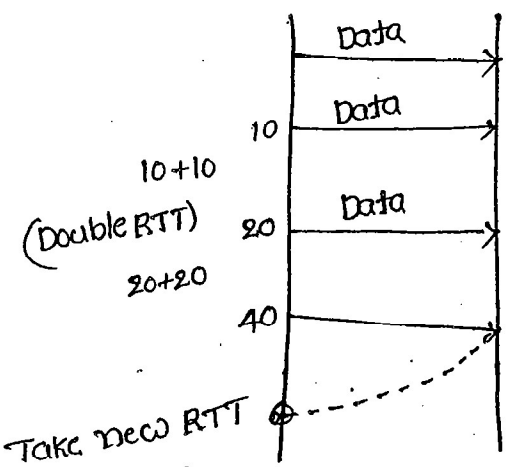
* if there is a time out, there is a possibility to receive 2 data packets
  1. From original packet
  2. From re-transmitted packet.

* Then there is an ambiguity that which ACK must be considered for next calculation.

∴ Therefore, Karn's has resolved this ambiguity by proposing the follows.

=> For every timeout double the timeout for the next transmision and continue this till to get a proper Ack.

=> Then we will go back to the jacobson's algorithm.



State Transition Diagram:

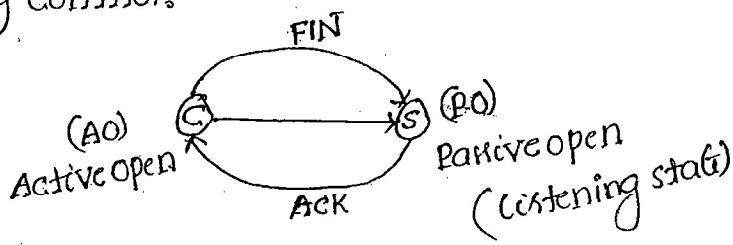
Need for state Transition dig:

* To evaluate any protocol, we use one of the following 2 methods.

1. Get the specification of the protocol develop a software for it, integrate with network operating system and evaluate it's features. it is time consuming process and costly.

2. Get the specification of protocol develop state transition dig for it and then evaluate it's features.
   it is a shortcut method but not so powerful.

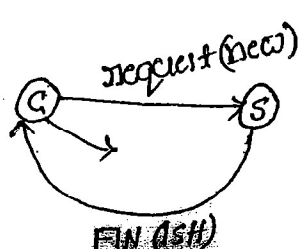
Dialog control:

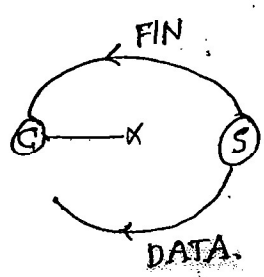


Purpose of Time wait state:

* As soon as Acknowledgement is generated client will go to time wait state instead of closed state by suspecting problem with acknowledgment. Even if ACK is lost and "FIN" segment is re-transmitted.

* it is treated for some connection as client is maintaining the connection in timed-wait state. if there is no such state, then retransmitted FIN is treated for new connection.





* Server wants to terminate the connection by using FIN/ But it denies, then before the time exceeds client wanted to establish a new connection, so it requests a new connection to server. After the request reaches server, client gets the FIN.
* Then client thinks, that it is the termination of newly established connection which is not True.
* In order to avoid such confusion, "Sequence numbers" are given to FIN at the same time, some time it is also allotted for "FIN" which is called as "Time wait state".

Limitations of State Transition dig:

* Error control procedures are not depicted in the dig
* Re-transmission are not shown in the dig.

Nagle's Algorithm: (Used in wide Area Network):

* checking the server that it works connection not, using remote system for checking it send characters.
* one character is sent at a time which cause silly window syndrome efficiency reduces drastically, for checking 100 characters, one at a time for which RTT is more for character typing.
* At such conditions, add the headers for all the bits & send it which improves performance.
* But it is not applicable in LAN technologies, but supports WAN tech because of RTT and typing speed capabilities
* Round trip time is less but input speed is high.

① Maximum data = 64 KB

= 65,535 ⇒ | 65515 | 20 |

Data Header.

② RTT = 30 msec

$\alpha = 0.9$

NRTT = 26

Basic algorithm = α (RTT) + (1−α) (NRTT) ⑤.

= 0.9 × 30 + (1−0.9) (26)

= 29.6 msec

T.O = 2 * 29.6 = 59.2 msec

D_{new} = |30 − 26| = 4

DE = 0.9 * 4 + (0.1) * 4

= 4

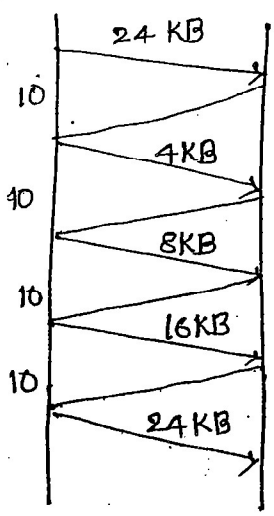
T.O = 4 * D.E + ERTT

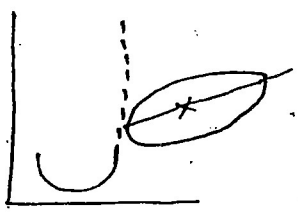
= 4 * 4 + 29.6

= 45.6 msec. ⑥.

③

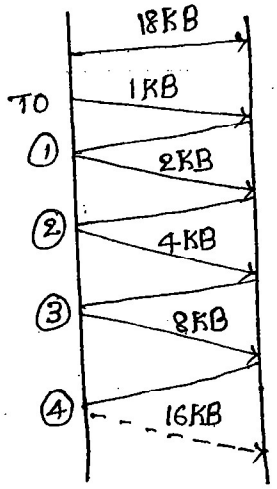
After 40 ms a full window is transmitted





congestion.

④



After time out occurs go back to

window size = 1

Ans: 16KB.

⑤. TCP USES SWP

$$\text{Throughput} = \frac{1 \text{ window}}{RTT}$$

$$= \frac{65,535 \times 8}{20 \text{ msec}}$$

$$= 26.5 \text{ mbps}$$

$$\eta = \frac{26.5 \text{ Mbps}}{1 \text{GBPS}}$$

$$= 2.6\%$$

⑥. Transfort data unit

Total numbers available $= 2^8 = 256$ and they should be consumed in 30sec.

$$\text{Data rate per connection} = \frac{128 \times 8 \times 256}{30}$$

=

(9) Probability of

total no $= 2^{32}$

$= \frac{10^6}{2^{32}} = 2.3 \times 10^{-4}$.

=

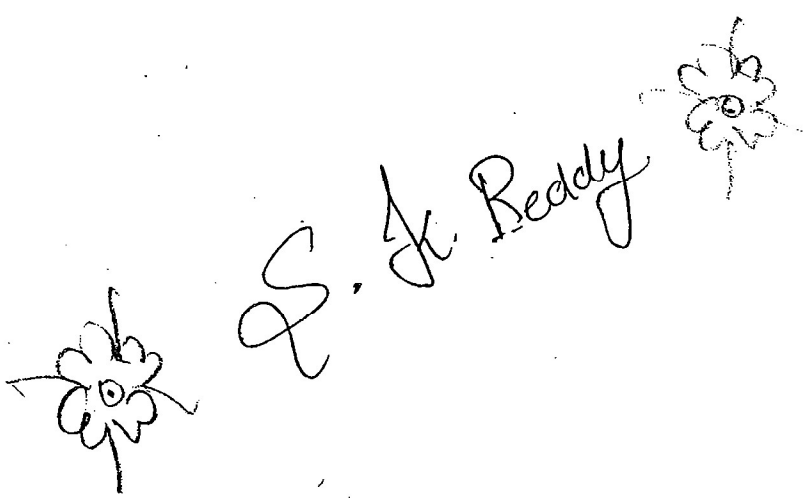
(11) A typist can type 600 characters for minute i,e to type a character takes 100 msec.

case:1: since RTT is very much less than typing speed. this algorithm cannot be implemented.

case:2: Since RTT is exactly equivalent to typing speed this algorithm cannot be implemented.

if RTT is more than 200ms then we able to implement this algorithm Nagle's algorithm.

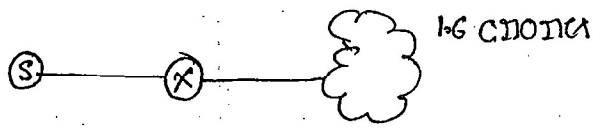
=



User Datagram Protocol (UDP):

Need for UDP:

For multicasting and broad casting applications, TCP can't be used Hence we need TCP.



Server communicates to 1.6 crore systems in TCP, we require 1.6 crore connections which cannot support a huge connection.

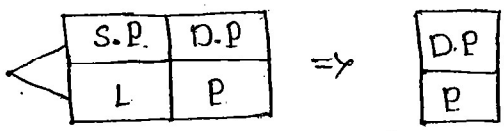
* Applications that requires constant data flow, cannot use TCP, Hence UDP is being used

  Eg: Rocket launching.

* Applications that requires bulk data transfer cannot use TCP Regulated flow in TCP, fluctuated flow in UDP.

* Applications that requires fastness than reliability cannot use UDP.

Since UDP is a connectionless, many fields in TCP are not needed in it.

optimal 

TCP

* connection-oriented
* slow
* Reliable
* overhead is high
* HTTP, FTP, SMTP, Telnet are used

Applications

* web applications
* Mail, RSA applications

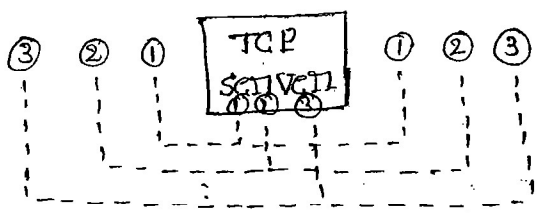
UDP

* connection less
* fast
* un-reliable
* overhead low.
* DNS, TFTP, NFS, SNMP, multimedia & Realtime.

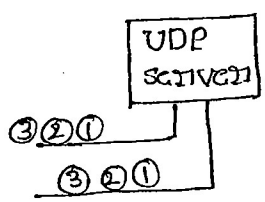
Applications.

* Name transfer Application
* Nlw management applications
* Multimedia & Realtime appl's

concurrent process:



iterative process:



* TCP and UDP Port numbers are different.

Domain Name System (DNS):

* it is using UDP, it's purpose is to keep track computers and services in a network environment.

it has four applications

→ Name Translation
→ Host Aliasing
→ Mail Aliasing
→ Lode balancing.

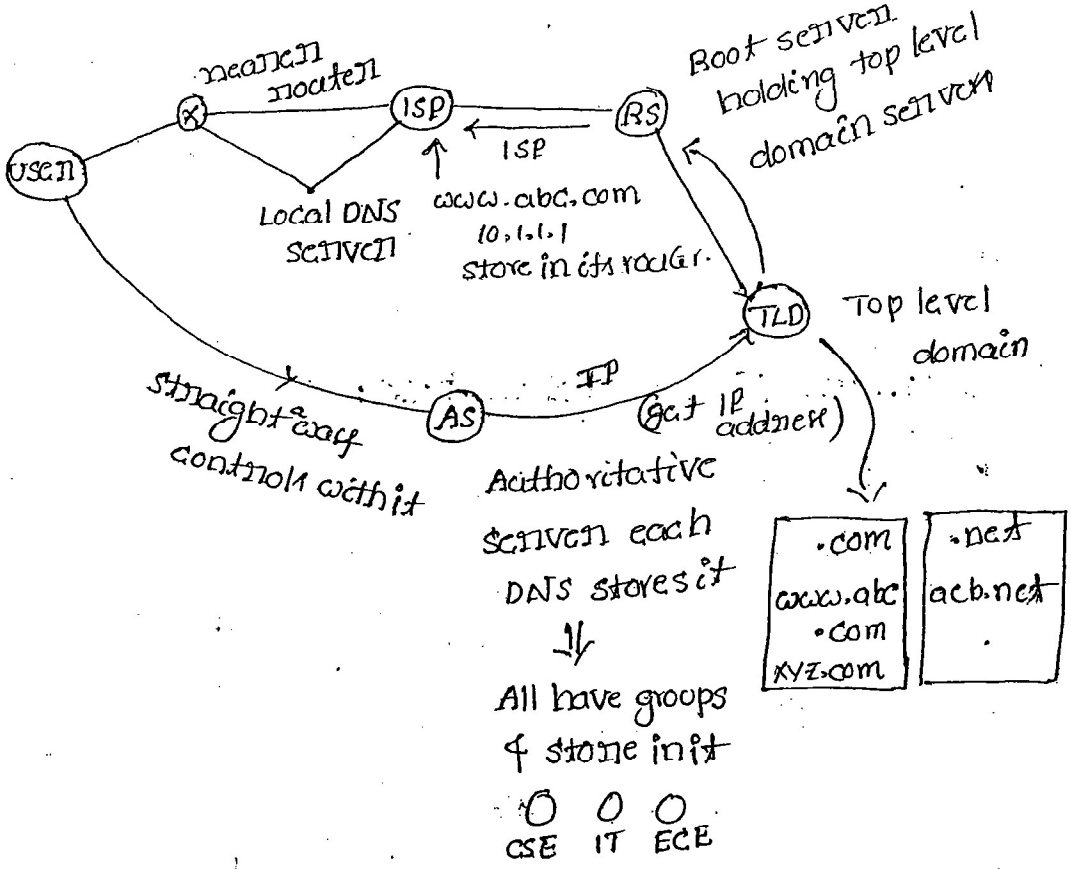
→ Root name server
→ Top-level domain server
→ Authoritative server
→ Local DNS server.

* it uses distributed database to perform its applications.

information about computers and services are stored in these servers in terms of resource records.

* Each resource records consists 5 attributes.

→ Name
→ TTL
→ class
→ Type
→ value.



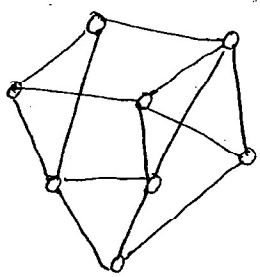
CSE IT ECE

Fault gateway: IP routers specifies the default address, suggesting that it can't know particular address.

* once a request for a website is sent, it gets stored in ISP and the local routers, whenever again a request of same website is sent then there is no need to visit all the servers, and there is only subsequently request simply it shows the website without visiting root server.

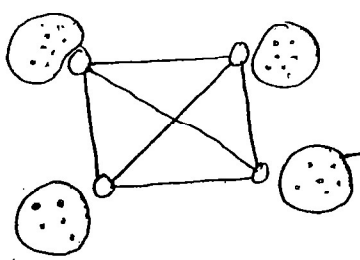
* Mainly 70% of internet services are provided by ISP & local routers.

* when Top level Domain servers maintained in terms of clusters. These are connected with MESH topology. it helps to improve efficiency and reliability.



Mesh connection

Rs maintains many routers and get connected not only single router.



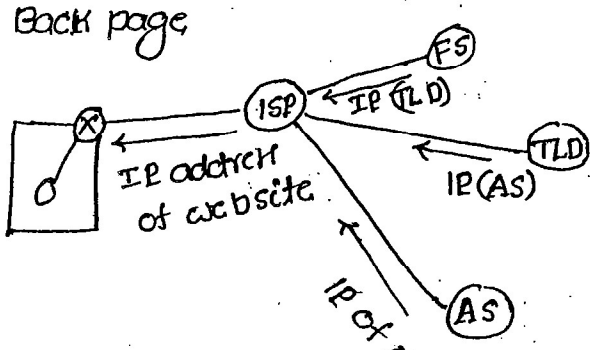
servers in the form of clusters and distributed data base.

* it will help for fast searching (effective usage)

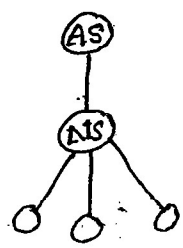
=> Getting IP address can be done in two ways.

1. Back page

2.

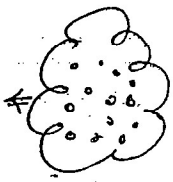


NS: Name Server



* Rs asks ISP to get connected

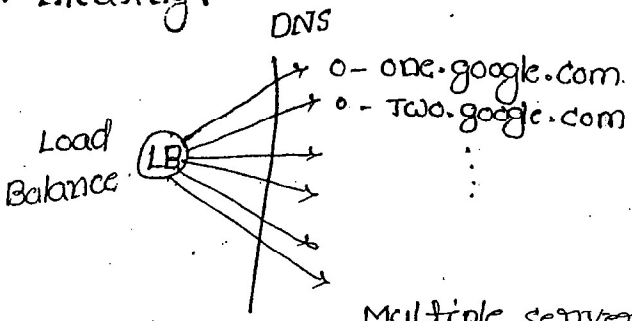
⇒ Giving names to systems in a network (hostnaming)



L1.one.abc.com ⇒ by seeing name, one can understand position of the system.

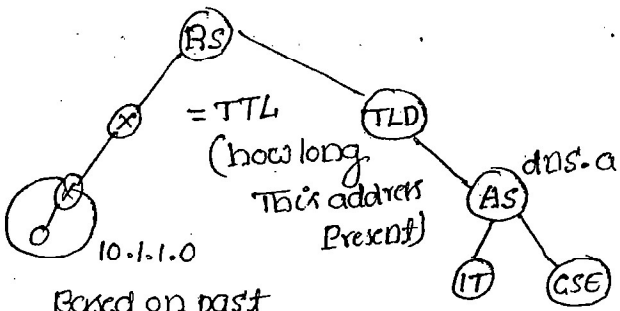
In "abc" college building, "one" lab, "one" first system

Mail Aleasing:



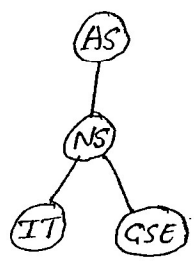
Multiple servers placed in different places.

Distributive Database:



Based on past data of a server

dns.abc.com 24 hrs in 10.1.1.2 changed



IT.abc.com 24 hrs in 10.1.1.3, www.abc.com 24hrs NS, dns.abc.com.

www.abc.com 2hrs cname backup.abc.com.

canonical name ⇓ Alternative names for websites.

Consider a request of page for which the IP address is stored in router. if again the same page is requested with little time gap. it is restricted from nearer router.

Circular dependency ⇒ glue record.

Replication ⇒ sharing data.

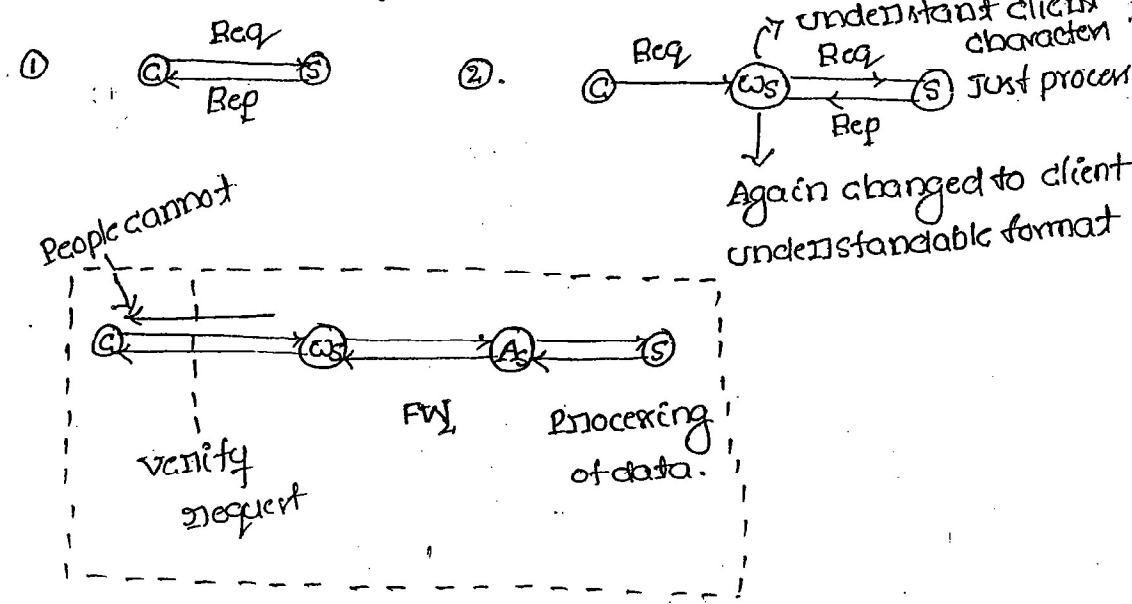
More replication ⇒ use TCP

Translation ⇒ use UDP.

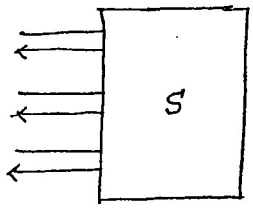
HTTP:

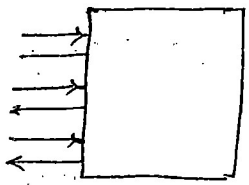
* it is a client server protocol using port 80 in TCP.
* it is stateless protocol.
* There are 2 types of HTTP protocols
  - → persistent
  - → Non-persistent.
* it have two types of messages → Request
  - → Response.
* HTTP will perform its operations by using 8 different methods
  - => Head: HTTP developed through ← web browser, version, os (all html pages stored in webserver)
  - => get Method.
  - => Put => creation (create an object in server)
  - => Post => changing (Modification)
  - => delete => defete
  - => Trace => debugging
  - => options => optimization.
  - => connect => Through the channel and users security Perform transactions.

HTTP uses 3 different types of status for its operations:



*Non-persistent connection in 1.1 (in 2.0 is changed to persiste)





Persistent

Every time new connection is established

using some time, limit connection is placed.

2- succesful
3- redirecting
4, 5 } Error Method.

* Each & every request expires head method (client).

Display status error (stateless protocol).

File Transfer protocol (FTP):

* it is a client server protocol, uses port numbers 20 & 21 on TCP

* it have two types of connections:

  → Data connection (using port-20)
  → control connection (using port-21)

* There are 3 modes of operations.

  → Active mode
  → passive mode
  → Extended passive mode.

* There are 2 flavours of FTP

  FTP => Authorized users

  TFTP => Anonymous users.

* To keep track data transmission, FTP uses wide varieties of status codes and also it is supported with many no.of commands.

* TFTP: never requires username, and password. All the users within the applications can access the data.

Eg: LIC Policy application

fig: Active mode

*

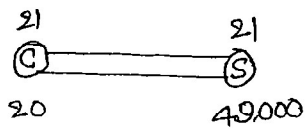


fig: Passive mode

* in passive mode, server generates a dynamic address and it is being get connected with in the client.

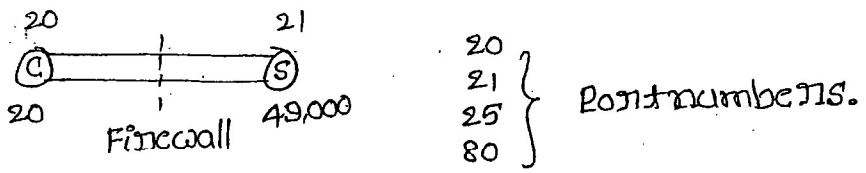


fig: Extended passive mode.

* in this mode, a firewall is being placed b/w client and server, where the time is assigned to packet. if packet is received after the time then firewall discards the packet.

FTP must be monitored constantly in there time, so for this are use a no.of commands.

* The monitoring of FTP constantly arauces the checkcking of status codes.

HTTP + SSL => HTTP

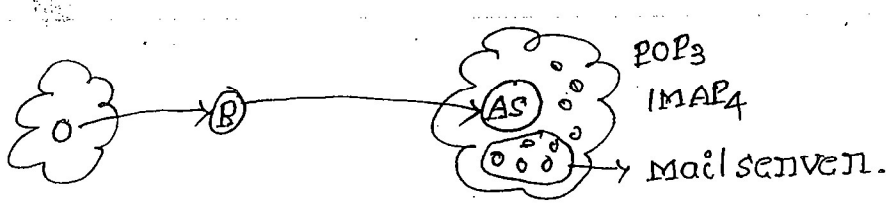
FTP + SSH => seccurity.

* By using SSH => a secured pipeline is established which is used in FTP.

SMTP:

* it uses port 25 at TCP.
* it is host-to-host transport protocol.
* it is text-based protocol, but enabled with multimedia with MIME extension.
* it is having two components.
  - => User agent
  - => Mail Transfer agent.
* it is a part of push-full mechanism in the mail commu
  ∴ SMTP is used to push the mail.
* POP-3 and IMAP4 are used for pulling the menage.
* it is an example for asynchronisation communication.
  ie client and servers are indirectly connected.
* if client and servers are directly communicated (connected) => synchronous communic

* it is connected to DNS server also.



* if a host is needed to send the data, then it gets connected to the router and then the router gets connected to the Authoritative server through SMTP Protocol. The AS identifies particular host in the mail server and "push" the data into that hosts. and if any other host requires the data within the network, then the data is being "pulled" by the server.

* Hence the mechanism is being considered as the "PUSH-PULL" mechanism, for this mechanisms POP_3 and $IMAP_4$ are used.

* $IMAP_4$ is more advantageous than pop_3 for this consider an example:

Before downloading a file, a msg is delivered to check it is SAFE (or) associated with any virus. if user is interested to continue (either it is SAFE or UNSAFE) then only it process and if he is not interested simply "CANCEL" it => it is

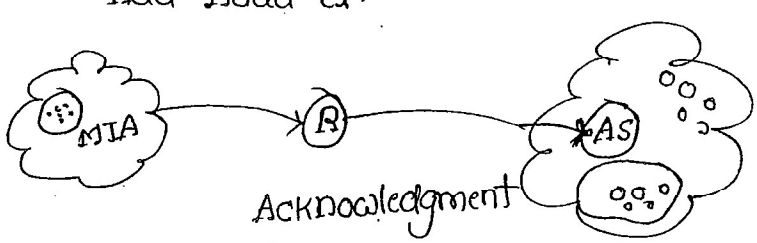
advantages of $IMAP_4$ over POP_3

* $IMAP_4$ Push some of the dangerous mails to junk mails where as POP_3 cannot do.

* A folder is created in the inbox and are configured with some mail address. so that there is no chance of mixing important information.

Two components:

* User agent
* Mail Transfer agent (Incharge to negatiate with TCP for connection SMTP)
* Forward message => attachments are available with the message
* Reply message => all the attachments are automatically dropped.
* Read Receipt component (if set) => Then user (who send a mail to other can able to know either the other user who received a mail) has received an read it (or) not.
* while user reads the mail an acknowledgment is sent to the sending user that the recipient had read it.



* User agent is used for handling the attachments and the disattachments and mail transfer agent is an incharge to have a connection with mails of SMTP through TCP.

Internet protocol.

Different special IP Addresses:

S.NO	Source IP Addresses	Destination IP Address.
1	✗	✗
2	✗	✓
3	✗	✓
4	✓	✗
5	✓	✓
6	✗	✓

* The above 6 IP addresses are used only for special purpose within the internet protocol.

* Some of the IP addresses are used to represent only source IP addressing and some are destination IP addresses. C, C it represents to have NID and some HID.

1. NID (✓ ✗) HID (✗ ✗) => filled by 0's

A: 10.0.0.0
B: 150.157.0.0
C: 192.168.1.0

Name: "This network address.

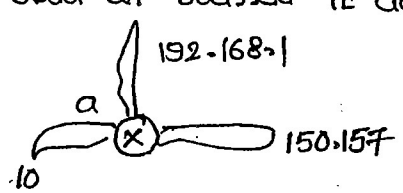
Purpose: used by Routers.

=> Represents different network

D	10.0.0.0	150.157.0.0

=> not valid

* They cannot be used as source IP address & Dest IP address.



10.0.0.0	a
150.157.0.0	b
192.168.1.0	c

* This type of IP addressing systems are used by routers to identify the network, for which it belongs to.

X ✓

A: 10.255.255.255

B: 150.157.255.255

C: 192.168.1.255

Name: Directed Broadcast system // Delivering packets to all system in some other network.

Purpose: To all in sender network

D	10.1.1.1	150.157.255.255	✓

D	150.157.255.255.	10.1.1.1	X

* Host ID is appended with all 1's and network ID can be any other value.

3. 1's 1's
NID HID

255.255.255.255.

Name: Limited Broadcast address. // Delivering packets to all systems in our own n/w's

Purpose: To all in the local network.

* Both the host ID and network ID's are being appended with 1's

4. 0's 0's
NID HID

0.0.0.0

Name: Dynamic IP address.

Purpose: Dynamic Host configuration protocol (DHCP).

* In dynamic more no.of system, so it can be limited IP addresses.

* I
E
By
a.
* Lo
n
ar
* S

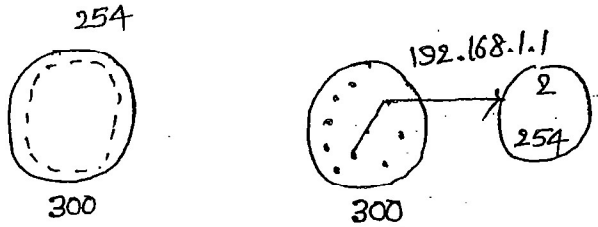
By

as

Ac
t
a

5.

* More IP addresses & less no. of systems



* For a particular duration of time an IP address is Permanently given to the user. After performing the "logout by any user on particular system, then the IP address is assigned to the other user.

* Likewise 254 IP addresses are managed by the servers (but not client) in the FCFS basis. If there are more request an "Queue" is maintained.

* so it is used only for the source but not destination

D	0.0.0.0	192.168.1.1

Dynamic:

The operating system on the basis of administration, assigned the IP address to the system.

Auto: The operating system directly assigned the IP addresses to all the systems without the intervention of administrator automatically.

5. 0 ✓
NID HID

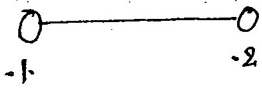
A: 0.1.1.1

B: 0.0.100.1

C: 0.0.0.100

Name: Host in this network

Purpose: local communication



192.168.1.1	192.168.1.2

(or)

0.0.0.1	0.0.0.2

IP

* The communication is done among the two host systems within the same network => local communication.
* The communication is not possible for host in one network with the other host in other network.
* Therefore, the network ID is fixed or unique, and only the host ID alternates for every communication.

6. 127... Any

127.1.1.1 or 127.100.0.255

Name : Loop back address

Purpose: inter process communication

* self checking or self connectivity with checks

127.0.0.0	X

127.255.255.255	X

* Both IP addresses are not valid other than these two IP address all the others are valid. which starts with:

ID = 127

=> which of the following IP addresses are used as only source IP addresses

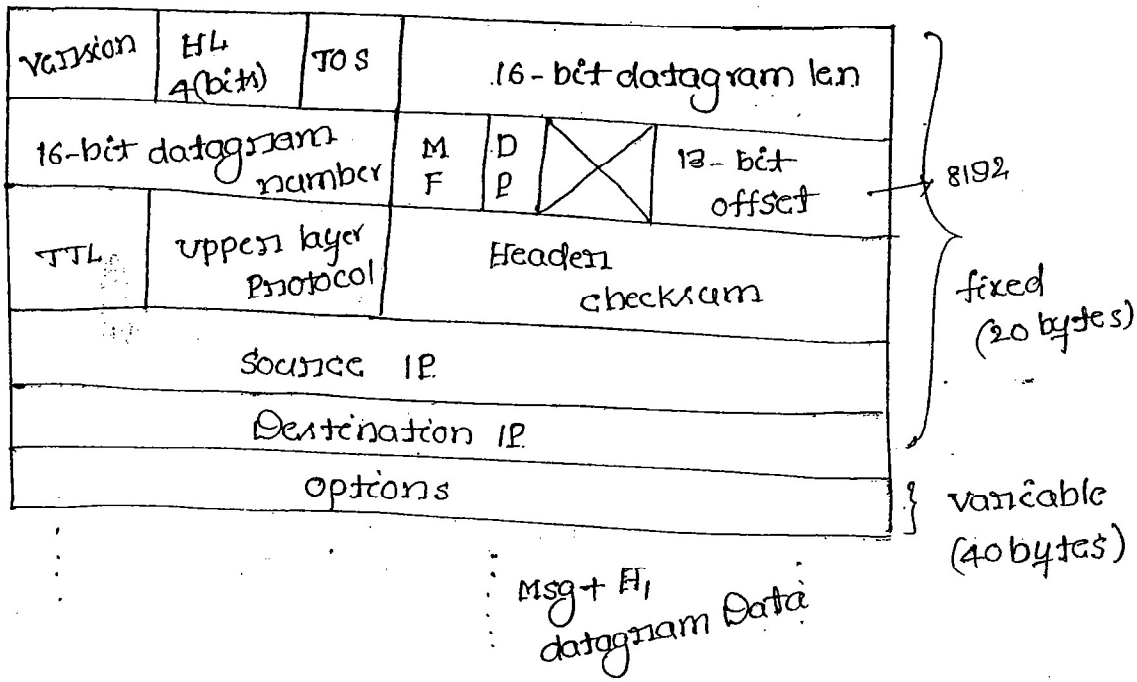
A. 10.1.1.1 (B.) 0.0.0.0 C. 0.0.0.1 D. 127.1.1.1

=> which one of the following IP addresses are used as both source and destination addresses.

61

A. 10.1.1.1 B. 255.255.255.255 C. 10.255.255.255 D. 0.0.1.1

IP operations:



* Version => to indicate either IP_4 or IP_6 Packet (4-bits)

IP_6 less complicated than IP_4.

* Header Length: Maximum size => 60 bytes (40+20)

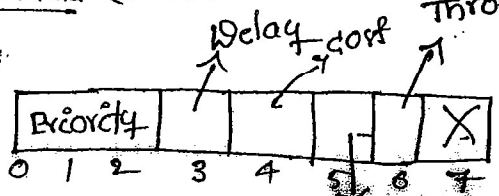
Minimum size => 20 bytes.

$2^4 = 0 \ldots\ldots 15$

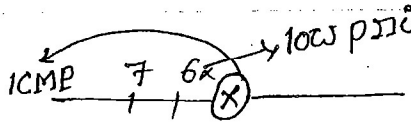
Constant scale factor = 4

$\frac{\text{Actual header length}}{8}$ = Available header length in PKT (4-bit)

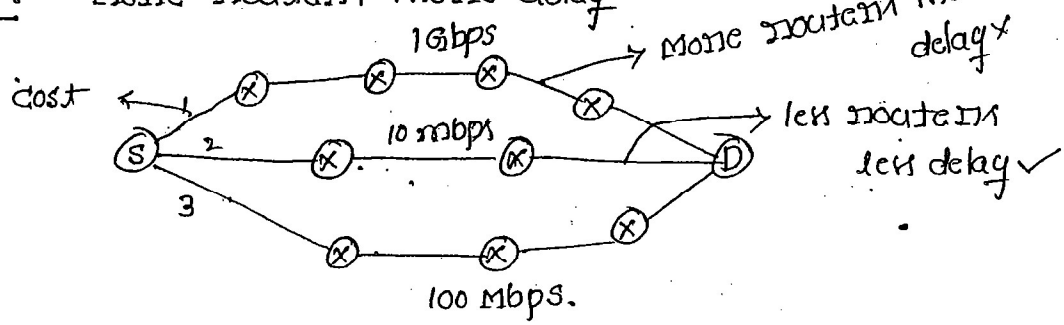
Types of service (TOS):



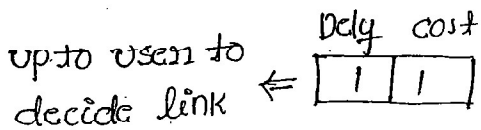
* Based on priority, packet is transmitted to intermediate routers.



Delay: More routers more delay



Cost: considering bandwidth, num. of routers, error rate, distance among the routers, the cost is being calculated.



Reliability: it represents the "error rate".

Throughput: it depends on bandwidth, if high bandwidth for a link => then allows

2^{16} = 64 KB => ie maximum size of the total Packet.

16-bit datagram numbers:

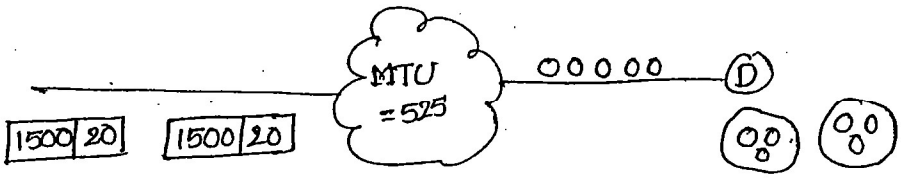
Every datagram associated with the sequence numbers starting with '0'.

More fragments (MF).

Maximum Transfer unit (MTU):

Fragmentation is applicable to only datagram packet but not for header.

* OFFSET indicates no. of data bytes ahead of this fragment in that particular packet.

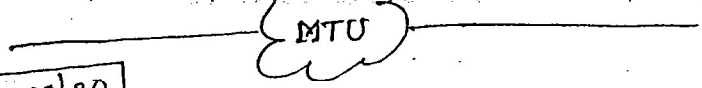


	①	②	③	④	⑤	⑥
504	505	504 505	492 490	505	505	490
	+ 20	+ 20	+ 20	+ 20	+ 20	+ 20
	5	5	5	6	6	6
MF	1	1	0 end	1	1	0
offset	0/8	505/8	1010/8	0/8	505/8	1010/8
		504	1003			

CSF = 8

Re-assembly Algorithm at destination:

* classify, fragments based on 16-bit datagram number.
* identify the fragment with offset=0 and designate it as a first fragment.
* identify the fragment with MF=0 and designate it as a last fragment.
* identify data in the first fragment and look for the fragment with same offset value and designate it as second fragments.
* Repeat previous step as many times as possible to cover all the fragment.

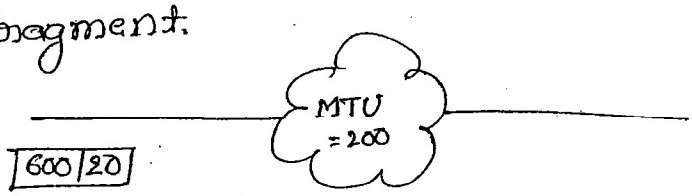


$\frac{64000}{8} = 8192.$

64000 | 20

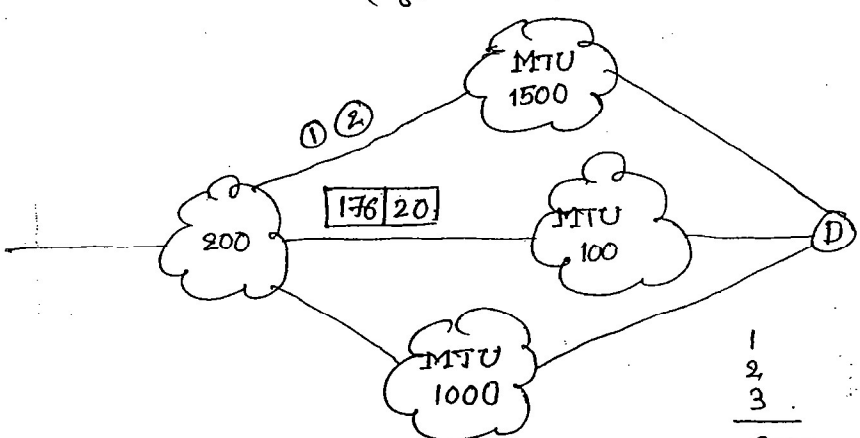
①	②	⑩	⑯	64
1000 + 20	1000 + 20	1,000 + 20	1000 + 20	

* Datagram data (or) fragment data must be divisible with '8' if not adjust it's number so that it is divisible with '8'.
* This rule is applicable for all the fragments except for last fragment.



600 | 20

176	176	176	72
180	180	180	80
+ 20	+ 20	+ 20	+ 20
MF 1	1	1	0
offset 0	$(\frac{176}{8})$ 22	44	66.



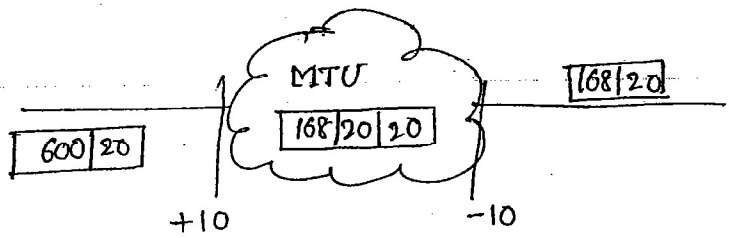
1
2
3

6x20 = 120
−20
100

* if offset = MF = 0 ⇒ Original packet.
* if any one of them is non-zero ⇒ fragment (intermediate fragment)

176 | 20

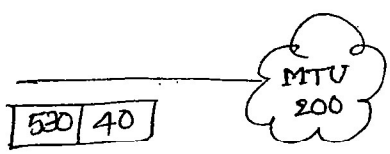
	80	80	16
	1 + 20	1 + 20	1 + 20
MF	1	1	1 ⊛ ⇒ Intermediate fragment.
offset	44	54	64

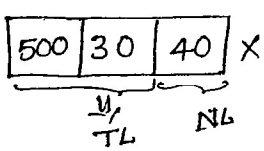


	168	168	168	96
	~~170~~	~~170~~	~~170~~	~~96~~
	+	+	+	+
	20	20	20	20
	+	+	+	+
	10	10	10	10
MF	1	1	1	0
offset	0	168/8	168+168/8	.....

Msg : 500
TCPH : 30
IPH : 40
MTU : 200

} Datagram data.

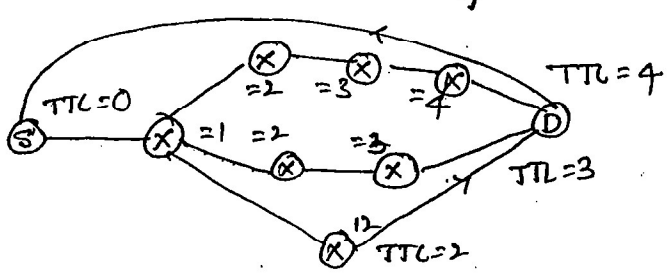




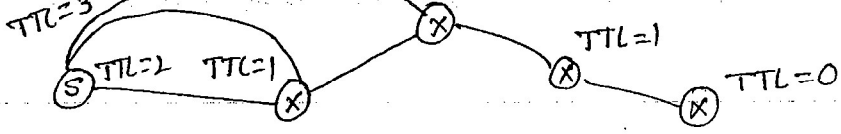
160	160	.....
+	+	
40	40	

Time To Live (TTL):

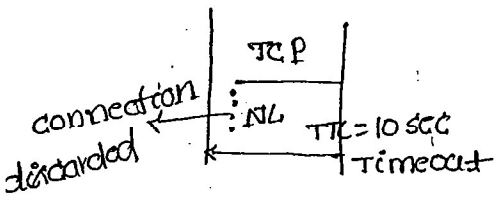
* it have four applications:-
  - -> To avoid infinite looping
  - -> To identify no. of routers between source & destina
  - -> To debog the network
  - -> To help upper layers in timer management

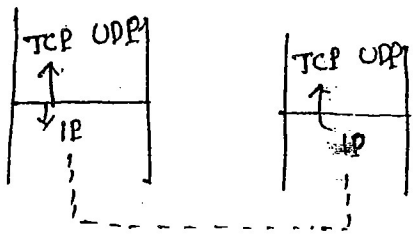


* It is possible to find no. of routers in a route by using

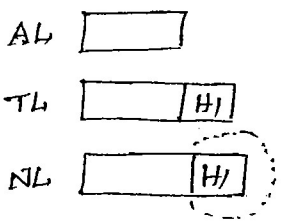


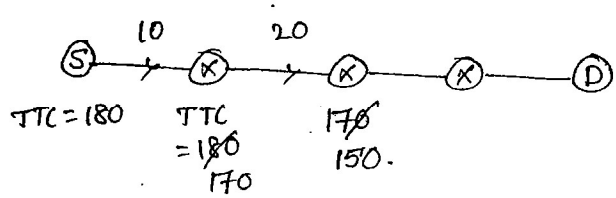
Timer Management:





* Header checksum is carried out at every router and only at Header.





→ TTL
→ MF
→ Offset
→ 16-bit datagram length
→ Options

} => Tend to change at every router (variable)

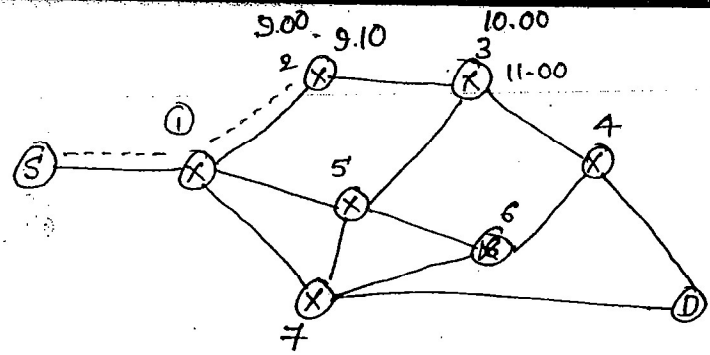
source
destination } => fixed.

Options:

* Strict source routing
* Loose source routing

} source will decide the route

* Record routing => Routers decides the route
* time stamp
* Security



Strict source routing: Each and every route is specified

(1, 2, 3, 4)

Loose source routing: only the important routers are specified and the route is generated based upon those routers => practical

1	2	3	4
1	5	3	4
1	7	6	4
	5	6	4

Record Routing:

(1, 7, 5, 6)

* Packet can be transferred as it wishes among all the routers

Time stamp: Arrival time and Departure time of each and every packet is stored.

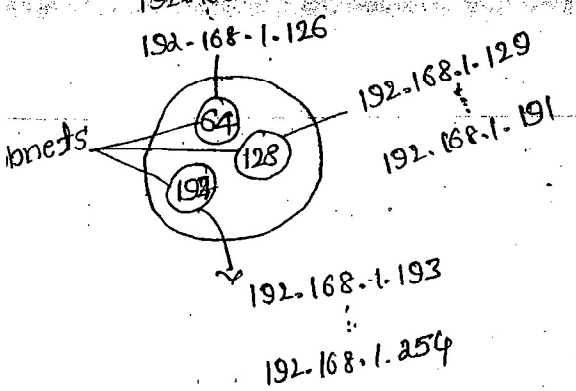
Security: Mails are sent along with certification which provides secured access for a page.

Q:

⇒ then which of the options are most available in all the fragments and which of them are necessary to present in any one of the fragments?

Ans: strict source routing, loose source routing } most available in all the fragments.

* Record routing
* Time stamp } must be necessary in any one of the following.



fcg: 192.168.1

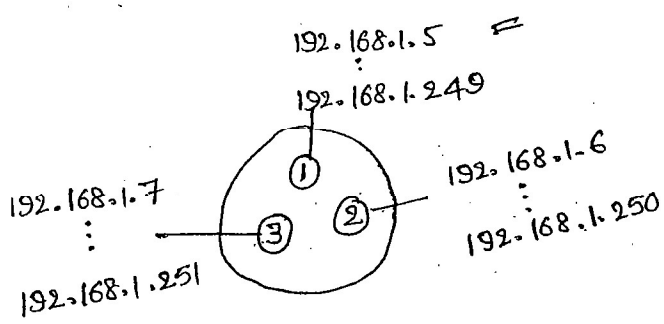
01 . . . . . .
01 000001 - 65
01 000010 - 66
⋮
01 111110 - 126

10 . . . . . .
10 000001 - 129
10 000010 - 130
⋮
10 111110 - 191

11 000001 - 193
⋮
11 000010 - 254

C: NID HID ⇒ borrow 2 bits
24 2,6
SID HID
$2^2 = 4$
$2^6 - 2 = 62$

128 64 32 16 8 4 2 1
0 0 = 0
0 1 = 64
1 0 = 128
1 1 = 192



⇒ 192.168.1

NID HID
24 6,2

128, 64, 16, 8 4, 2, 1
00 - 0
01 - 1
10 - 2
11 - 3

. 01
00000101 = 5
00001001 = 9
⋮
11111001 - 249

. 10
00000110 - 6
00001010 - 10
⋮
11111010 - 250

* Subnet id's and IP address of the network are changing

* when router receives a data packet, the following must be checked.

whether subnets are available (or) not

→ If not available then IP address is directly assigned to that.

→ If available then identify, the packet belongs to which subnet for this we use subnet mask (SM)

Subnet Masking:

* it is also a 32-bit system and it is used to indicate whether subnet are available (or) not in the network.

* if available, it will give the information about no. of bits borrowed from host id and their position based on the following 2 rules.

1. No. of 1's in the subnet mask, indicates n/w id plus subnet Id.
2. No. of 0's indicates Host Id part.

Eg: consider a class C IP 192.168.1.100 (192.168.1 | .100, SM) data point as previous problem.

subnet mask = 255.255.255.0

SM: 11111111.11111111.11111111.00000000

Rule (1): $\overset{24}{NID} + \overset{0}{SID} = 24$ (since, class c)

(2): HID = 8 (since, no subnets, nothing is borrowed from HID)

NID = 192.168.1

HID = 100.

Q. Consider a class network with SM: 255.255.255.192 identify no.of bits borrowed from HID and their position, possible subnets and their ID's possible no.of systems for subnet and range of IP addresses in each and every subnet.

Sol: SM: 255.255.255.192

.11000000

Rule(1): $\overset{24}{NID} + \overset{2}{SID} = 26$

HID = 6

No. of bits borrowed = 2

Their position is = 128th bit, 64th

Possible subnets = $2^2 = 4$

Their subnet ID's = 11000000

00 - 0
01 - 64
10 - 128
11 - 192.

No.of systems per subnet = $2^6 - 2$

Range of IP address = 62.

eg: Consider a class 'c' n/w with SM = 255.255.255.41. identify all data points as previous Question.

SM: 11111111. 11111111. 1111111.00101001

$\overset{24}{NID} + \overset{3}{SID} = 27$

HID = 5

No. of bits borrowed = 3

Their position is 32, 8, 1

Possible subnets = 2^3 (0, 1, 8, 32, 9, 40, 41, 33)

No.of systems per subnet = $2^5 - 2$

Their subnet ID's =

32 8 1		32 8 1
0 0 0 = 0	0 1 1 = 9	1 1 0 = 46
0 0 1 = 1	1 0 0 = 32	1 1 1 = 41

eg: Consider a class B n/w with SM = 255.255.255.0. identify all the data.

↳ subnets are available

Entire 3rd packet are borrowed for subnet IDs

∴ no. of subnets = 2^8

Their ID's = (0-255)

No. of systems = $2^8 - 2$

Their IDs = (1 to 254)

eg: Consider a class C network with SM = 255.255.255.15.

SM: 11111111.11111111.11111111.00001111

24 4

∴ NID + HID = 28

No. of bits borrowed = 4

Their position is = 1, 2, 4, 8

Possible subnets = $2^4 = 16$

∴ No. of systems per subnet = $2^4 - 2 = 14$

eg: Consider a class C n/w proposed an appropriate subnet mask to have 7 subnets each with 25 systems.

7 * 25 = 256 (since, it is class C)

3 bits must be borrowed (since, 7 subnets)

24 | 8

3 | 5

5 | 3

3-bits => $2^3 = 8$

255.255.255.224 ✓

255.255.255.7 ✓

255.255.255.41 ✓

255.255.255.67 ✓

} all are possible but left to right is appropriate.

eg: consider a class B n/w and propose an appropriate SM to have 150 subnets each with 200 systems.

$150 * 200 <= 64,000$

$150 \Rightarrow 8 \Rightarrow 2^8 - 2 = 254$

(required) 200

255.255.255.0
255.255.0.255
255.255.240.240
255.255.192.252

eg: consider a class C network. propose an appropriate SM to have 20 subnets each with 15 systems

$20 * 15 <= 256$

$300 \not<= 256$

not possible.

eg: consider a class C network, propose an appropriate SM of ~~60.6~~ 60.60.120

class C $\Rightarrow$ 8

2 6 1 7

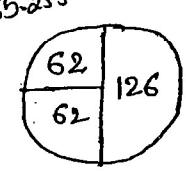
$2^2 = 4 \times 2^1 = 8$

$2^6 - 2 = 62$, $2^7 - 2 = 126$

(not possible) (not possible)

so in order to propose appropriate SM we use the concept of VLSM.

255.255.255.192



255.255.255.128

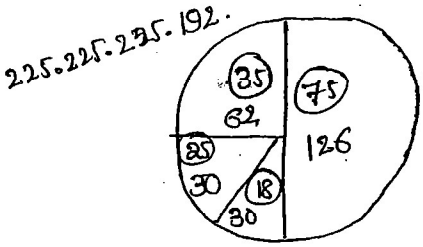
255.255.255.192

2^1, $2^6 - 2$
$= 2$ $= 62$

1 6

1 7.

Eg: consider a class 'c' network propose an appropriate SM to have 4 subnets of 75, 35, 25, 18.



2	6	1	7
$2^2 = 4$		$2^1 = 2$	
$2^6 - 2 = 62$		$2^7 - 2 = 126$.	

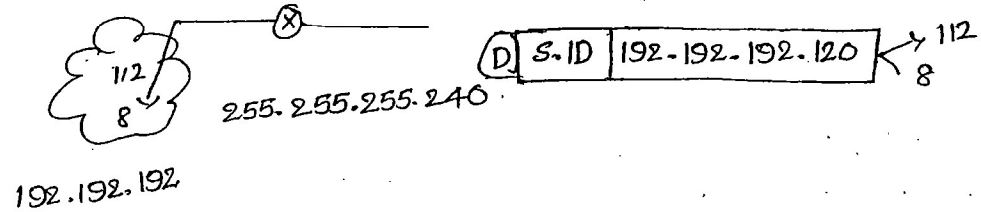
=

Eg: consider a class c network, propose an appropriate SM to have 6 subnets of 30, 25, 22, 20, 18, 15.

Class C = 8 $\quad 2^3 = 8$

2	6
$2^6 - 2 = 62$	

Eg:



Identify subnet ID, Host ID and directed broadcast address

IP: 11000000.11000000.11000000.01111000

NID SID HID

192.192.192

SM: 11111111.11111111.11111111.11110000

112 8

NID. SID NID

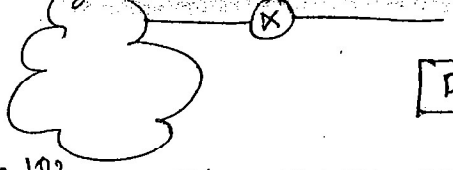
SID = 112
HID = 8
120

192.192.192.127

11000000.11000000.11000000.01111000 Broad

(76 to 32 10) Address

Eg:



D	S.IP	192.192.192.120

192.192.192

SM = 255.255.255.41

SID = 32+8+0 = 40

IP: 11000000 . 11000000 . 11000000 . 01111000

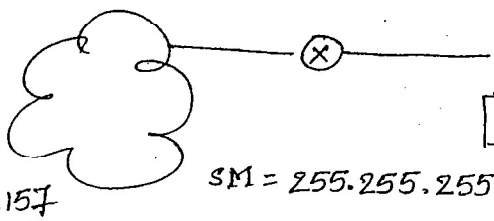
SM: 11111111 . 11111111 . 11111111 . 00101001

HID: 64+16+0+0 = 80

SID = 40
HID = 80

↳ 192.192.192.254 => Broadcast address.

Eg:



D	S.IP	150.157.100.70

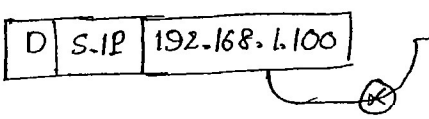
150.157

SM = 255.255.255.0

SID : 100.0
HID : 0.70

Directed Broadcast address = 150.157.100.255.

Eg:



D	S.IP	192.168.1.100

→ 192.168.1

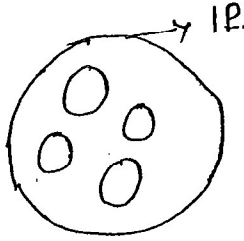
IP	SM	i/f
10.0.0.0	255.0.0.0	a
157.157.0.0	255.255.0.0	b
192.168.1.0	255.255.255.0	c
0.0.0.0	0.0.0.0	
0.0.0.0	0.0.0.0	

192.168.1.100	192.168.1.100	192.168.1.100
AND	AND	AND
255.0.0.0	255.255.0.0	255.255.255.0
192.0.0.0	192.168.0.0	192.168.1.0

* 0.0.0.0 => Default route subnet mask

* 255.255.255.255 => Host specific subnet mask

Supernet Mask:



192.192.0.0 : 11000000.11000000.00000000.00000000

192.192.1.0 : 11000000.11000000.00000001.00000000

192.192.2.0 : 11000000.11000000.00000010.00000000

192.192.3.0 : 11000000.11000000.00000011.00000000

8 8 6 2 8.

* It is a 32-bit system used to generate a single IP address of group of networks based on following 2 rules.

1. num.of one's in supernetmask indicates fixed part
2. num.of 0's indicates variable part.

11111111.11111111.11111100.00000000

SM = 255.255.252.0

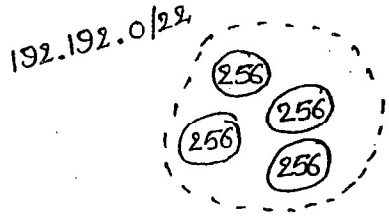
(1). 192.192.0/22 :

↳ Represents NID.

class C:

$24 - 22 = 2$

$2^2 = 4$



256 * 4 = 1024

/24 = class C

/16 = class B

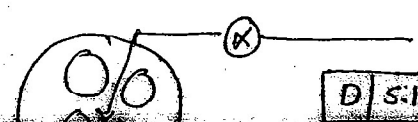
/8 = class A

32	
22	10
NID	HID

$2^{10} = 1024$

(2). 192.192.0

255.255.255.0



192
↓
11000000

Difference Between Subnet mask & Supernet mask

SUBNET MASK	SUPERNET MASK
* No.of is in the subnet mask is either equal to network id (or) more than network Id bits.	* No.of is in supernet mask is always less than NID bits
* Bits are borrowed from HID	* Bits are borrowed from NID.
* it is applicable to single network	* it is applicable for two or more network.

=> CIDR aggregation (classless Inter Domain Routing) => another name for supernet

	A	B	C
255.0.0.0	subnet	subnet	supernet
255.255.0.0	subnet	subnet	supernet
255.255.255.0	subnet	subnet	subnet.

192.192.0/22 => NID

/24 => class C
/16 => class B
/8 => class A

/22 - ? } 4
/24 - ? }
CIDR.

Routing Algorithms:

Routing is the procer of forwarding packets from one network to another. All the information needed for a router to forward packets to a hop can be found in the router's routing table.

⇒ Static Routing: it occurs when you manually add route in each router's routing table.

⇒ Dynamic Routing: it is the process of using protocols to find and update routing tables on routers and to maintain a loop-free, single path to each network.

⇒ common fields in a routing table:

* Mask
* Network address
* Next-hop address
* interface
* Flags
  - U: up
  - G: Gateway
  - H: host specific
  - D: added by redirection
  - M: Modified by redirection.

⇒ Delivery semantics:

unicast: delivers a msg to a single specified node.

Broadcast: " " " to all nodes in the network.

Multicast: Delivers a msg to a group of nodes that have expressed interest in receiving

NOTE: An autonomous system (AS) is a group of networks & routers under the authority of a single administration.

* Routing inside AS is called "intradomain routing"
* Routing between AS is called "interdomain routing".