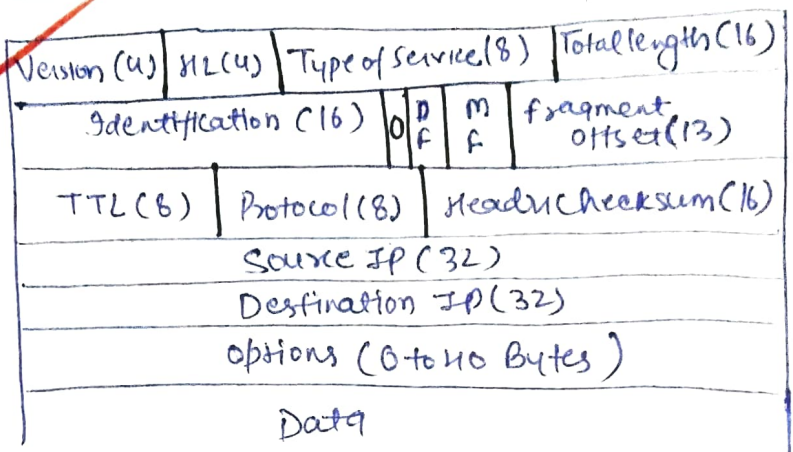
Internet Protocol

Introduction to IP Header (Day-6)

IPV4 Header



Version (4) → No. of bits HL → Header length

DF → Do not fragment (1 bit)
MF → More ——— (1 bit)

TTL → Time to live (8 bit)

Options → It is an optional feild. and written is Bytes.

1st row → 32 bits (4 Bytes)
2nd row, 3rd row, 4th row, 5th row → 32 bits
6th row → variable

(1-5) rows → 5×4 = 20 Bytes
first five rows should be definitely presented

Min. Header size = 20 Bytes.
Max. Header size = 20+40 = 60 Bytes.

Header length → 4 bits.
max. no possible = 1111 = 15

If each possible combination among (15) represent a single byte then for 60 Byte we need 60 combination but we have 15. so we do scaling

$\frac{60}{15} = (4)$ means calculate the header size and devide it by 4 and then put in HLo

Ex

HL size	HL value
20 Bytes	5
32 Bytes	8
40 Bytes	10

Range (20-60 B) (5-15) Range

You can differentiate both using ranges.

If not a multiple of 4 then,

Ex 30 add (more bits so that it become multiple of 4 as,

$(30+2) = 32\ \%\ 4 = 0$

These 2 bits are called dummy bits / padding.

At max we can add 3 dummy bits in worst case.

Version → Version 1 version 2 — version 6.

V1, V2, V3, V4, (V5), V6 → X

↓

Not using.

Each version have different packet structures and different parsing algorithms.

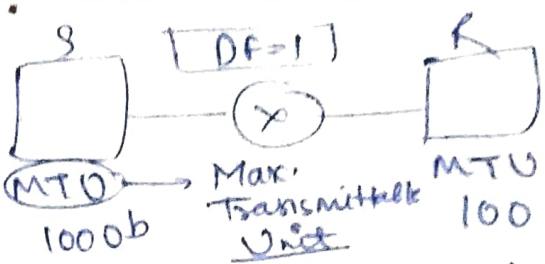
V4 → 0100
V6 → 0110

→ V6 is not in syllabus and V5 is not useful for various purposes, so we discard it.

Identification :- Use the number a datagram uniquely.

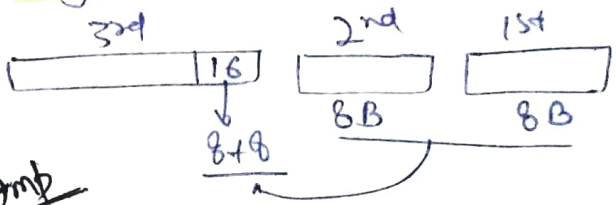
Df → Do not need to fragment datagram. (Df = 1)

Mf → More fragments are followings.



If DF=1 so we can't fragment and if data is of 1000b and receiver can accept only 100b at a time then router will discard the packet with an error msg, then sender will decrease the size of packet but DF=1 will be there bcoz sender doesn't allow anyone to fragment data bcoz of atomicity.

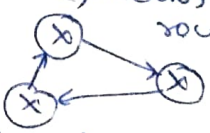
Fragment Offset → No. of data bytes ahead of this particular fragment in this datagram



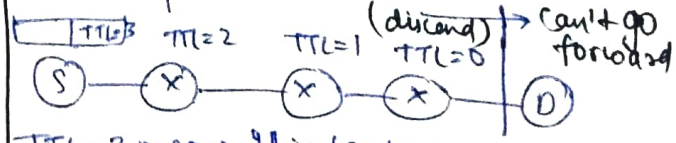
9mb

TTL : Routing table have 3 entries. If entry not present in table then default entry comes under consideration.

But the problem is if routher combination is like, means default routers is in cycle then

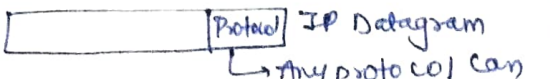


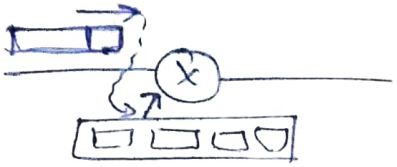
packet will continuously stuck in loop.



TTL=3 means this packet will receive under 3 Hop. No need to go more than 3 Hop

Protocol: TL → TCP/UDP; NL/DLL → IP, ICMP/IGMP





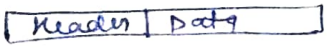
If queue is full then router check the protocol of packet is TCP then router discard one packet from queue (having other 3 protocols) and place the packet in queue.

Priority of Protocols

ICMP < IGMP < UDP < TCP

Header Checksum

→ Checksum calculated on header part of datagram.



At each router TTL ↓ by 1 and TTL is a part of header so, header is changing on each router. so checksum need to be calculated on each router.

If we include Data too in Checksum then it will become an Hdious task.

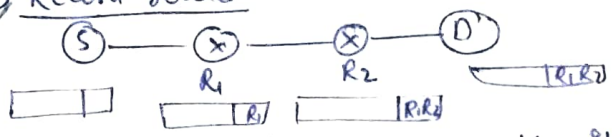
Other values might get change at router-
1) Fragment offset, MF, TL
2) Options
3) HL.

Source IP → Logical Add. for valid IP require valid (NID HID)

Can be used as (SIP, DIP)	NID	HID
x x	→ Valid	Valid → Valid IP
x	→ Valid	0's → NID
x	→ 1's	1's → DBA
x x	→ 1's	1's → Limited BA
Too host in same N/w	0's	0's → NM / subnet mask
x	→ 0's	valid → Host within a N/w
x	→ 127	0's → Default Entry.
		valid → Loop back Add.

Options:

i) Record route



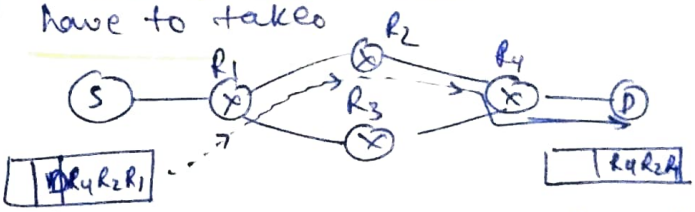
If record route is set then, it will add the IP Address of each router at header.

→ It can record at max '10' IP Add. and if you want to separate each IP Addres by some bits then best practical no. of of IP Add. record is '9'.

→ Only ISP can use this feild.

ii) Source routing

You can specify the route that packet have to take.



Each router will change the address to next HOP.

→ Above method is also called Strict Source routing

DR4R1 → loose source routing bcoz you can't worry about either choose R2 / R3.

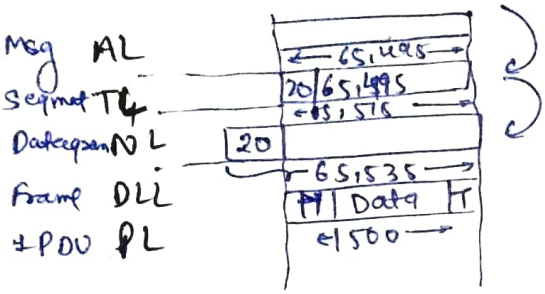
→ Only N/w administrator can use source routing.

Fragmentation

Difference between segmentation and fragmentation.

Total length (length of header + length of data)

Max size of IP datagram $= 2^{16} - 1$
$= 65,535$



TL can hold 65,495 Byte of data or payload but AL can send any amount of that then it is responsibility of TL to devide the ~~segment~~ data into segment, so that it can fix inside the space for payload (65,495).

→ Segmentation done at TL.

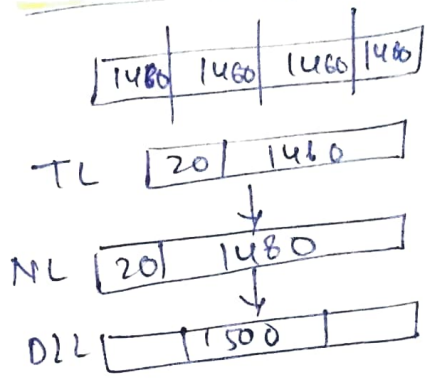
Similarly at DLL we choose LAN technologies. suppose we choose Ethernet then we can send at max 1500B at a time So, Network layer devide the data packet in packet of 1500 or less size packet.

→ fragmentation done at NL

→ framing done at DLL.

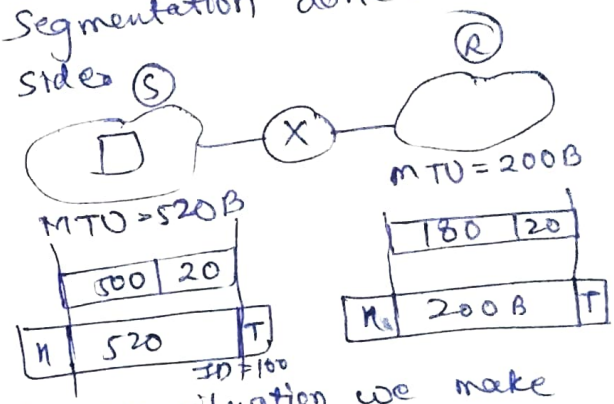
→ on any other LAN technology it might be possible that we can send the data without fragmentation.

→ TL devide the size of packet of data in a way that data can fix in the Data at DDL. So we can avoid fragmentation.



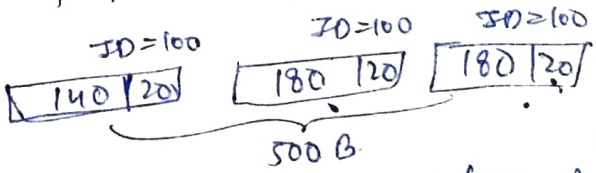
Ex fragmentation

Segmentation done at host side.



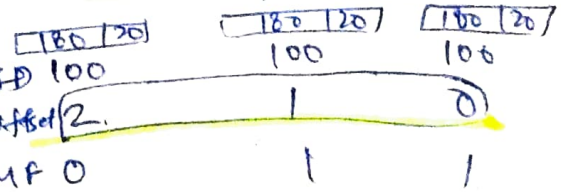
In this situation we make part of data so that it can fit at Receiver side.

So at receiver's end after fragmentation,



→ Both datagram as fragment present at NL and go through DLL framing.

→ Every packet must have a header in datagram



Q How receiver identify that these 3 fragments are belong to same datagram.

Ans → Using identification no. in header part.

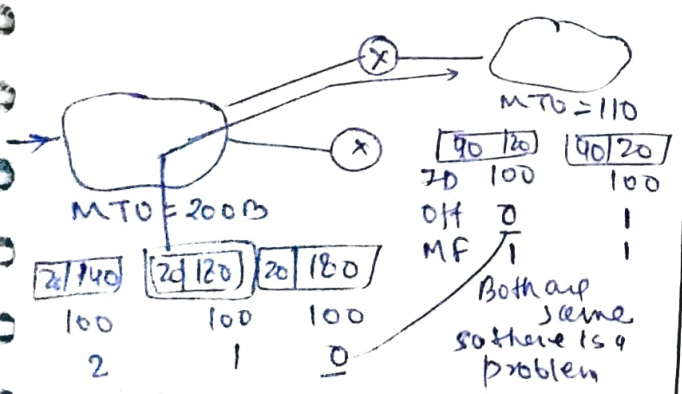
Q How will you identify the sequence of merging?

Ans → fragment offset feild use to solve above problem. It holds order of fragment

Q How to get or reach the end of all fragments

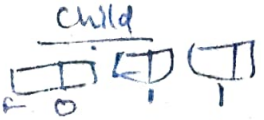
Ans → More fragment (MF) feild.

But writing fragment offset as 0, 1, 2 -- so on is not scalable bcoz, if any datagram can posses more fragmentation as,

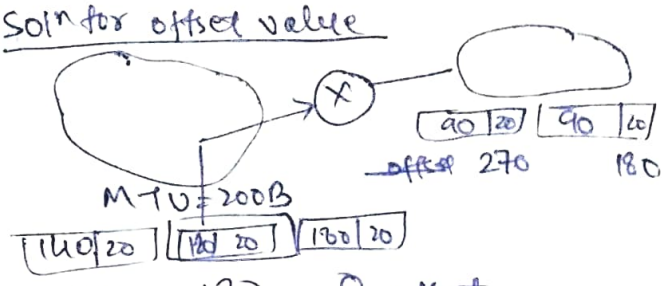


So, what should be the values of offset?

for MF, if parent have MF=1 then all the fragments of that parent will have MF=1 and if MF=0 (parent) then all the child parent will have MF=1 except last fragment



Soln for offset value



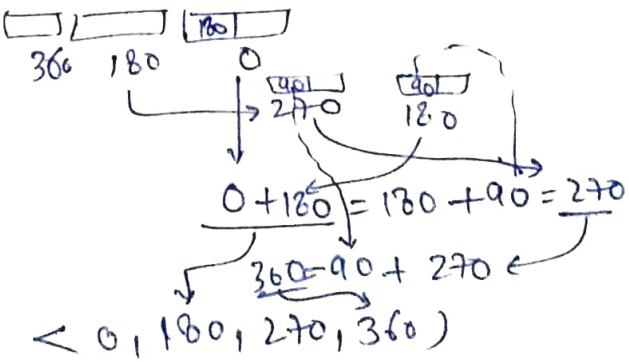
offset value is equals to no. of data bytes ahead of the fragments

→ So This way offset become scalable.

(0, 180, 270, 360) is way to represent offsets

→ Don't just sort the value of offsets and merge it bcoz in case if any offset is missing it will not detect it

So method use is take first offset and add it with size of next data fragment and verify that number should be in sequence



< 0, 180, 270, 360)

हर बार fragment का जो Data size है उसे offset में add करो और Check करो add करने के बाद जो sum आया है वो offset में हो।

To find the data size we use,

DS = (Total length − Header length)

Q. fragment offset is 13 bits so max no. can save is

$2^{13} - 1 = 8191$

But if data size is 65,515 and we do fragments in such a way that last fragment get only a 1 bit data as

1	20		140		120

↓

65,514 bits ahead of this but offset can have at max 8191.

So, we do scaling too here as header length feild.

→ For fragment offset we use scaling on the factor of 8 (2^3)

$\frac{2^{16}}{2^{13}} = 2^3 = (8)$

So

140	20		180	20		180	20

140/8 = 17.2 ; 180/8 = 22.5 ; 0/8 = 0

↳ But decimal is not allowed.

So, size must be divisible by 8. To make it divisible we can add extra bits but packet size is already full. so we can't add more bits.

Another approach is decrease the bits.

148	20		176	20		176	20

Off 144 ; 176/8 = 22 ; 0/8 = 0

Total length 168 ; 196 ; 196

Similarly for children

first fragment have same offset as parent

88	20		88	20

Off $22 + \frac{88}{8} = .33$; 22

If packet have data size of n bits (max) but it have only (m bits) so that $m < n$ then,

then we can add some extra bits to make it multiple of 8.

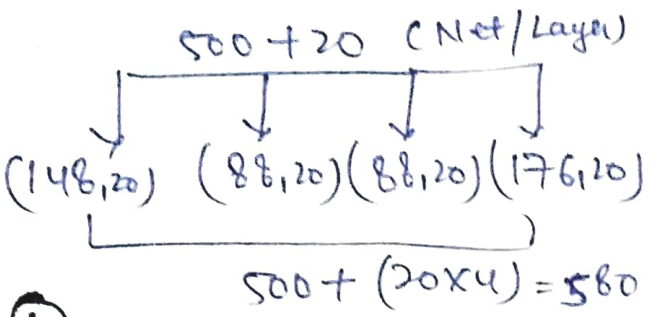
→ But if packet is last packet then it have no need to be multiple of 8 as (148 data size packet).

If parent data offset is multiple of 8 then its child will also have data size multiple of 8 if their parents are followed by another packets.

176	20	followed by
148	20	

→ fragmentation done by the routers not the source.

→ we send at senders end 520 Bytes as,



① So it has 60 bytes of overhead at NL.

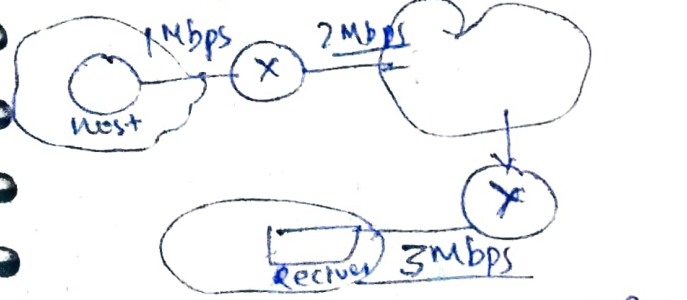
② We are sending 3 more fragments, and total 4 fragments.

③ Efficiency = Useful bytes / total bytes

$= \frac{500}{500+(4\times20)} = \frac{500}{580}$

④ Bandwith Utilization / Throughput

$= \eta \times Bw$



Each line have different Bw

→ In $\eta \times Bw$ you choose the min. Bandwidth (here 1Mbps).

So, Throughput

$= \frac{580}{580} \times 1$ Mbps

→ we choose 1 Mbps bcoz it can accept by every link but if we choose 2 Mbps then it is not accepted by 1st link and similarly for 3Mbps.

→ 1 Mbps is also called as bottom neck bandwidth.

Theory about fragmentation (9mb)

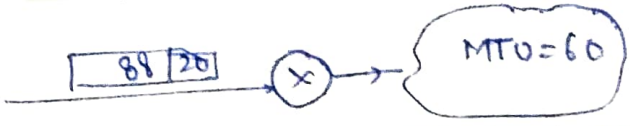
	20 | 176	20 | 88	20 | 88	20 | 148
offset	0	22	33	44
MF	1	1	1	0
HLF	5	5	5	5
TL	196	108	108	168
ID	100	100	100	100

→ fragmentation done at routers, not at the source.

→ All fragments combines to form a datagram (reassembled) at destination not on the router on the way. bcoz it might possible that all the packets follow different paths in (datagram service).

→ If all are following same path then it might be possible that we have need to do fragmentation again on successive routers.

→ Refragmentation



	8 | 20	48 | 20	40 | 20
Id	100	100	100
Offset	32	27	22
HL	5	5	5
TLF	1	1	1

→ Offset of first fragment going to offset of parent

→ MF of last fragment is MF of parent

Reassembly Algorithm

In order to identify that datagram is fragmented, we need both MF and offset.

→ first fragment identified as (MF = 1 and offset = 0)

→ Intermediate fragment as (MF = 1 and offset ≠ 0)

→ final fragment can be identified as (MF = 0, offset ≠ 0)

→ (MF = 0, Offset = 0) means no fragmentation.

Algo.

1) Destination should identify that datagram is fragmented using (MF and offset).

2) Collect the fragments belong to same datagram using Identification no. (done by Destination).

3) Identify first fragment (offset = 0)

4) Identify subsequent fragment (using $\frac{\text{data bits}}{8}$ + offset)

Data bits = Total leng. − header len.

5) Repeat 4) until MF = 0

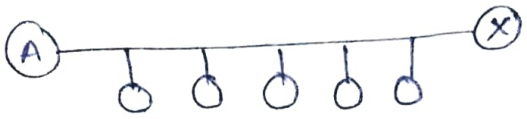
Protocols and Concept of N/w Layer (Day-7)

① Broadcasting

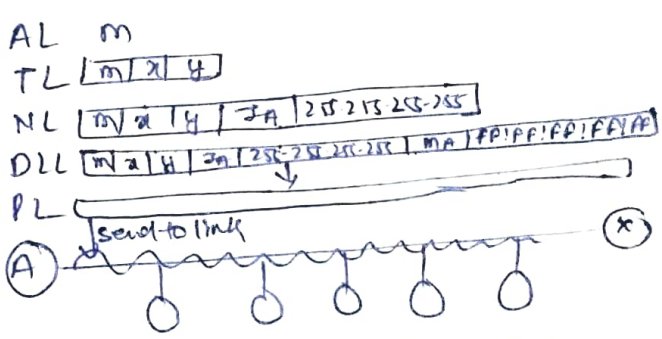
→ 255.255.255.255 → Limited Broadcast Add. (IP)

→ FF:FF:FF:FF:FF:FF → (Mac Address)

→ One more is directed broadcast add.



A wants to send packet to everyone in same N/w (Limited Broadcast)



→ Every network is limited to routers

→ Router will see the broadcast packet, accept it and not allow to go forward.

→ If you want to do DBroadcast instead of LBroadcast, change the IP of Destination to add. of destination router.

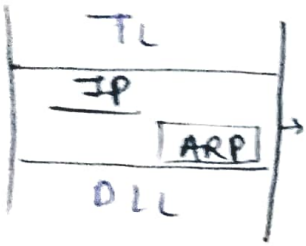
→ Broadcasting is not a general concept of Internet. It is a concept of Data Link Layer.

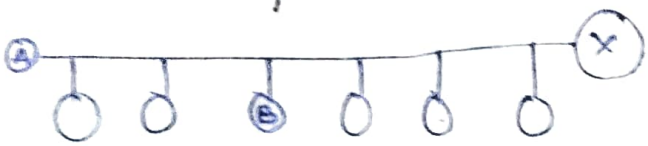
② ARP (Address Resolution Protocol)

~~gmp~~

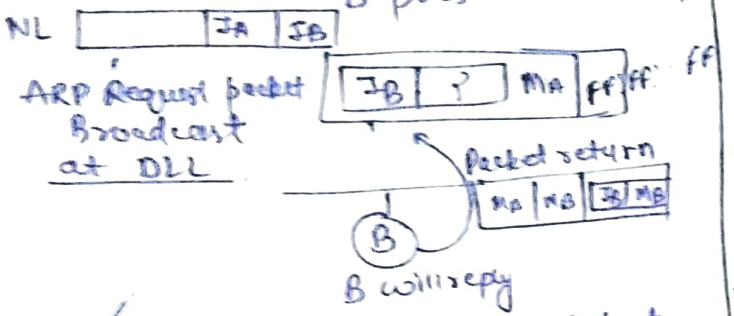
Given IP add. converted to Mac Address.

→ If packet in same N/w to transmit put MAC of destination, if packet need to send in diff. N/w put Mac. of Router.





Account to send msg to B present in same n/w



→ ARP request is broadcast but B's reply is Unicast

i) ARP reply is Unicast.

ii) Host → Host : Case 1
Host → Router : Case 2
Router → Router : Case 3
Router → Host : Case 4

ARP can apply in these cases.

Special Address: (127.—.—.—)

If you send a packet to B and you get no response then there may be following reason,

→ ① Either router not working
② Either NIC of anyone from A and B is fail.
③ Destination is not taking request.
④ Link is failed.

→ To check internet is working or not you have to check it from your end.

① Test self connectivity by sending a packet to yourself, to check the NIC of source.

→ If destination IP is start with 127 then packet come again via DLL to same host.

Command

ping 127.—.—.—

Don't use these two numbers. [0 0 0 / 255 255 255]

→ Using ping we can check the working of NIC and all layers.

Ex: ping 127.0.0.1

RTT = 15ms → NIC is working

if you get TO (Time out) means something is failed.

AL
TL
NL
DL
PL

↑ Loop back Add (127...)

If server and client are running in same host then we use (127....) because it's just like a interprocess comm^n.

Interprocess comm^n is a concept of OS not of CN.

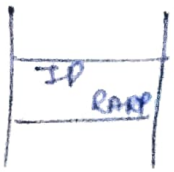
RARP → (Reverse ARP)

Mac ⟶ IP

(Network file Server) → (Central server (NFS))

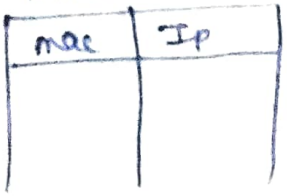
MAC Present in ROM of NIC. So it is permanent.

→ Computer knows its mac add. not IP address bcoz IP address is not permanently saved anywhere.

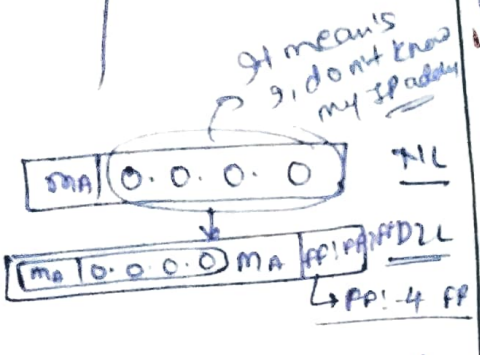


→ IP and RARP have no interaction with each other

RARP Server

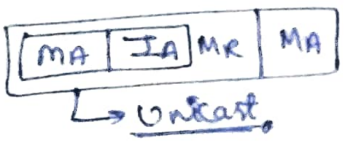


RARP req



This frame will get reply from only RARP servers

RARP server reply



→ RARP works at N/w Layer.

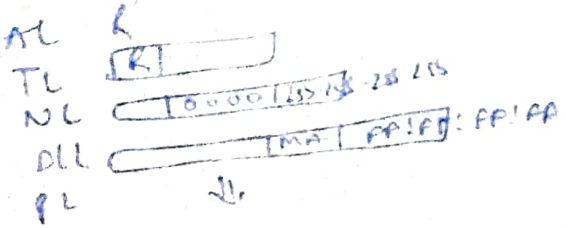
If RARP is not in same N/w then we get no reply means router work as boundary for RARP request

Disadvantages

→ In each n/w we need a RARP server. (Distributed)

→ Mapping table is static means not updating its data. (IP ⇒ Host)

Bootp → Similar to RARP but its run on Application Servers.



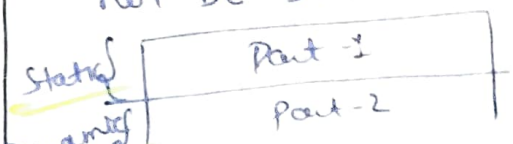
Bootp have standard mapping table contain IP and mac addresses.

Only one bootp server is required → Advantage

Mapping table is static → Disadvantage.

DHCP → Dynamic Host Configuration Protocol

→ Mapping table need not be static.



Part-1

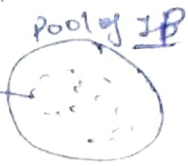
Public IP Add. →

mac	IP
M1	I1
M2	I2

Part-2

Private IP →

mac	IP	lease time
MA	IA	20 sec



If anyone ask for a ~~public~~ IP address, it will allot an IP from pool to a particular mac and give some time for which IP is available in use.

→ After lease time station needs to send a renew request, to hold and IP.

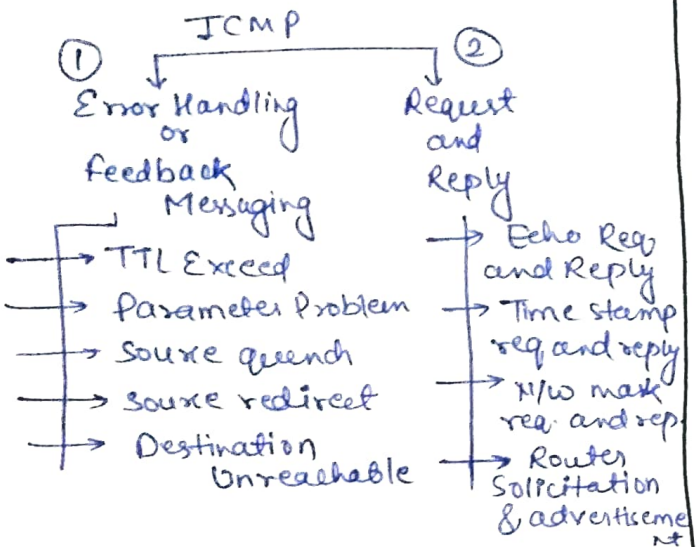
Bootp	RARP	DHCP
• Less Popular • Easy to manage	• Not in use now.	• very popular • Backward Compatible with Bootp.

Introduction to ICMP v. 9mp

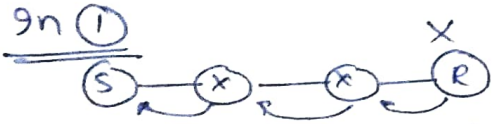
→ Works on Network layer.

→ Every ICMP aggregate with IP

→ Internet Control Message Protocol.



9n (1)

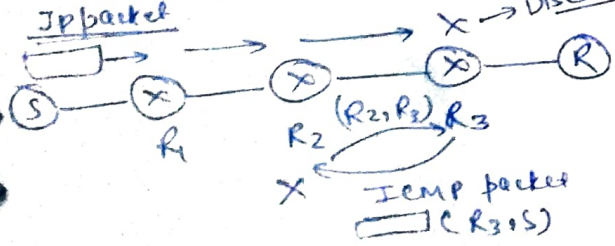


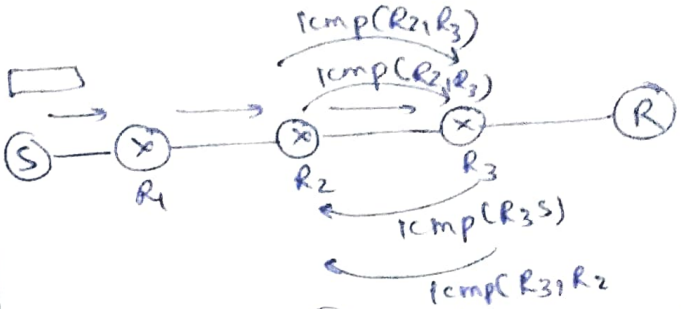
if packet discarded at any end (means either at router or destination) it will transmit a one way ICMP message.

9n (2)

Here you send a request packet and receive a reply packet.

IP packet is like a datagram, if it get lost then its assistent ICMP replies an error message.





Assume that (S) sends a IP packet for R but at R_3 due to congestion the IP packet discard and it sends a ICMP message to S but at R_2 due to congestion that ICMP message get discard and R_2 sends a Icmp to R_3 this process repeated in a way that both R_2 and R_3 get discarded each others ICMP packet. This way we can stuck into an infinite loop.

To solve above problem we come with the solution that ICMP can't be generate on discarding any ICMP packet.

→ IP packet discard → generate ICMP

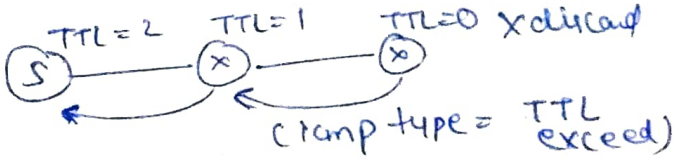
→ ICMP discard → Noting to do

So,

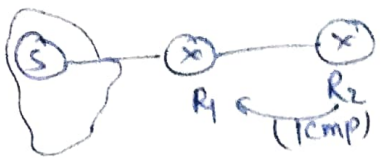
IP + ICMP → Unreliable

ICMP feedback messaging :-

(1) TTL Exceed : If packet doesn't receive before TTL = 0 it will get discarded.



Source Quench

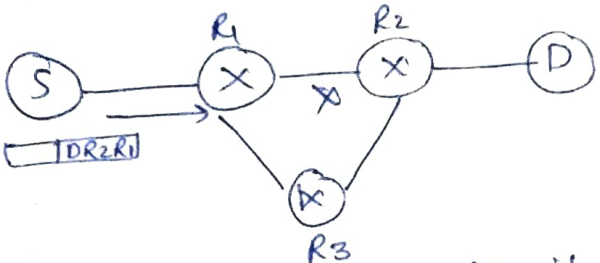


If R_1 is sending lot of packet at a time so that,

R_2 can't handle more packet then it send icmp ($R_2 \rightarrow R_1$) with type = source quench.

(quench means stop) ~~means~~ then source quench means router is saying to source to quench

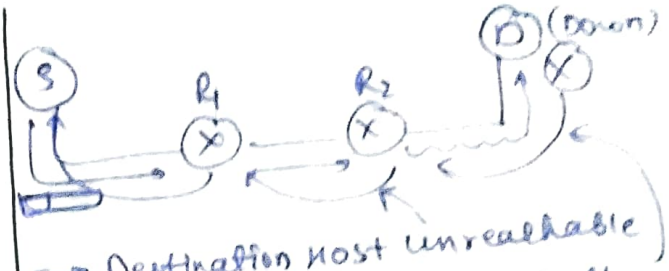
Parameter Problem: [Using strict source routing]



In case of strict routing if packet needs to follow the path $S \rightarrow R_1 \rightarrow R_2 \rightarrow D$ but link b/w $R_1 \rightarrow R_2$ is down then it have and another router for $R_1 \rightarrow R_3$ but bcoz of strict routing it can't go through R_3 then R_2 sends a message to (S) as (icmp type = parameter problem)

→ If the bits get corrupted then we get different checksum and get parameter problem bcoz checksum verifies at each router.

Destination Unreachable

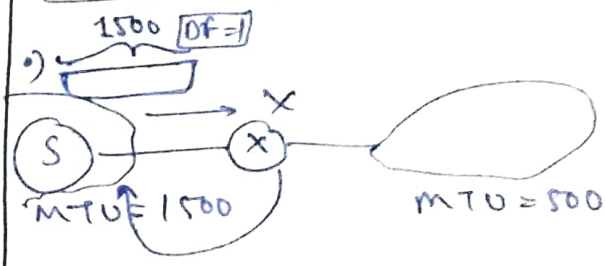


→ Destination Host unreachable
→ Destination port unreachable

•) Assum that packet have destination IP address of (D) then, when packet reaches R_2, R_2 sends ARP request to D that I have your IP address send me you mac add. and if the destination is down then R_2 sends an ICMP packet as (Destination Host Unreachable)

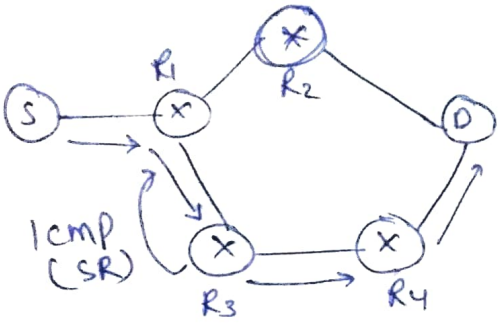
•) And if packet is accepted by the destination host and process for which the packet has come (port no) is not present in host then destination host sends a ICMP message as (Destination port unreachable)

•)



If DF = 1 then packet can't get fragmented at Router but the MTU of destination is 500 so router can't send packet without fragmentation. then, router send ICMP as (Destination Host Unreachable) along with (FR, DF=1, MTU=500) FR → fragments required).

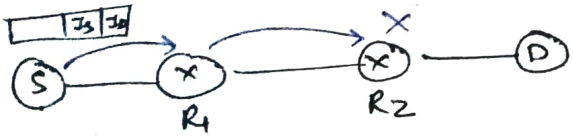
Source redirect



→ In all other cases ICMP is an error msg means message discard but in source redirect ICMP is a warning.

→ If S → D via R_2 is a better path then via R_3 then R_3 will send a ICMP message to R_1 that ICMP (source redirect) that you have can opt a better path via R_2, but R_3 will not discard the packet. R_3 will transmit that packet and R_1 will send next packet via R_2.

Requirements of sender

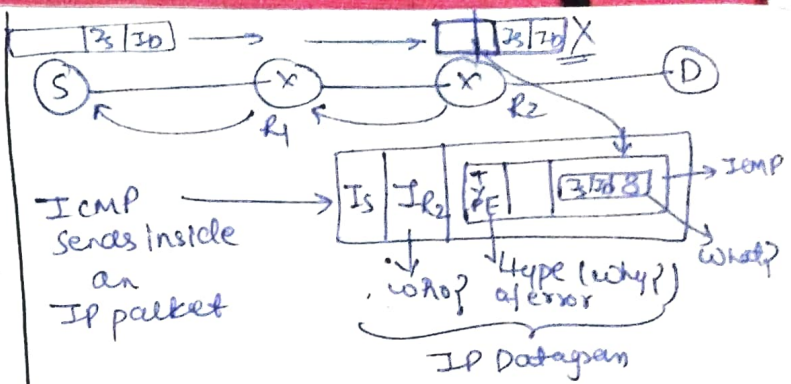


If packet discard at R_2 then by source IP add. at packet R_2 get to know about the source.

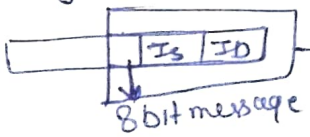
then what are the necessary info. R_2 sends to sender?

1) who? 3) what?

2) why?



To get info about what?



include as what? packet

what? packet contains → Is, Id and 8 bit message.

This way we send all the necessary info. to source.

→ To discard an IP packet we create another IP packet with ICMP.

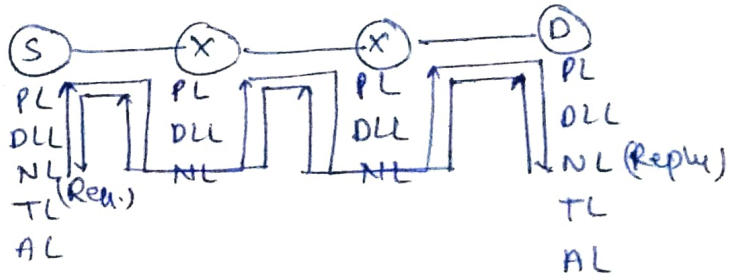
→ ICMP can never generate for ICMP packets.

→ ICMP can generate for both TCP and UDP.

→ In case of fragmentation ICMP will generate only for first fragment (offset = 0), means sender will only inform if first fragment is lost or discarded.

ICMP (Request & Reply messaging)

Echo req. and reply :- (ICMP packet)



This method is used to verify that N/w layer of destination & router is working or not.

To check that destination is working or not just send Echo ICMP packets

→ Ex→ ping command.

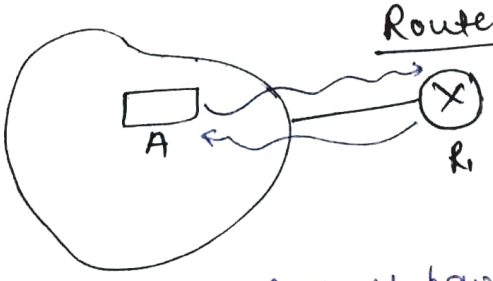
↓

Packet Internet Gofering

→ Ping is not a client-server application.

→ If we send huge / burst of ping request to single server then that server get busy in replying echo req. and you can make it down.

Router solicitation

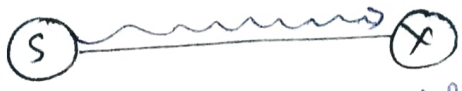


If source doesn't have info about nearby router then it sends or broadcast an ICMP message that I need the info. about nearby router and its nearby routers replies back called router solicitation.

→ Source set that router as default routers.

Router Advertising

If new router added to N/W then router broadcast a message that I am available for use.



After identify the default router you send a req. to router for subnet mask called Network mask req. and reply.

Time-stamp req. & reply

→ Use to synchronize the device in different time zones.

→ Earlier use, not useful in nowadays.

→ Today we use network protocol.

→ It provides synchronization

→ It is highly unreliable.

Traceroute application of ICMP.

tracert / traceroute → command

traceroute google.com → Linux

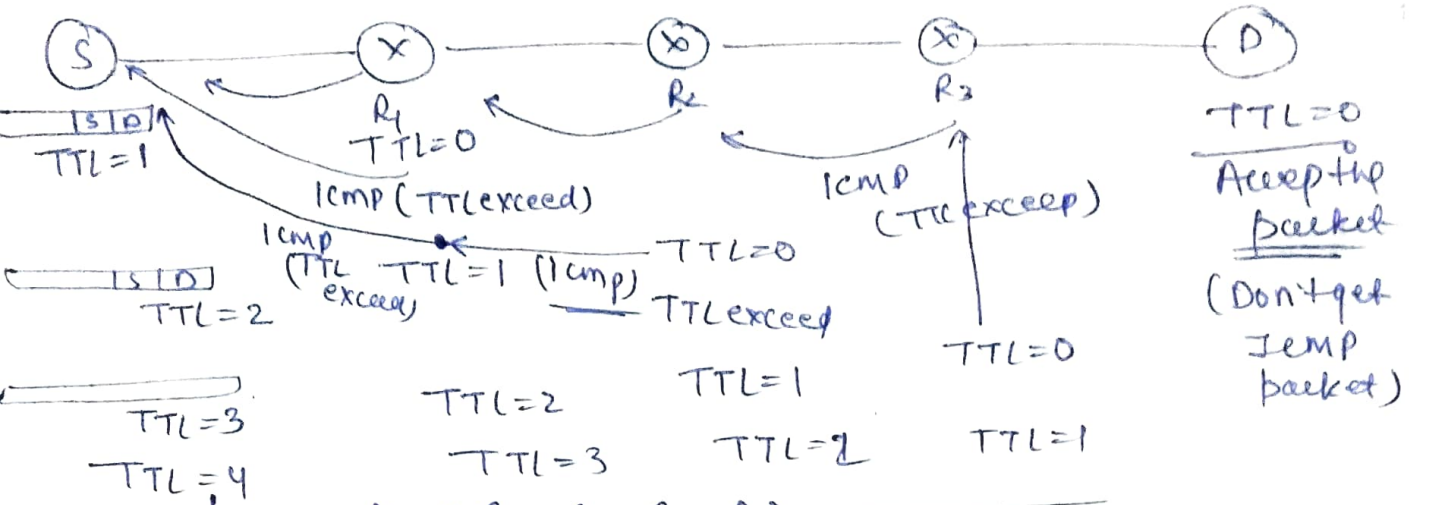
tracert google.com → windows

→ Trace the route between source and destination.

→ Record route is different from traceroute as in record route destination get the info. that packet comes from which routers.

→ In traceroute source identify that which is the route available to destination.

→ In record route we always get a correct path.



traceroute (R1, R2, R3, D) We increase TTL successively

→ Routers generally doesn't execute the unreliable command. So we cheat with them, we first send a packet with TTL = 1. It will reach upto R_1 and then if more routers are ahead R_1 will discard the packet and send an ICMP packet and we get the IP of R_1 in ICMP. Again we send same packet with 1 increment in TTL and same process will repeat. Using this method we get the info of IP about all the routers in between.

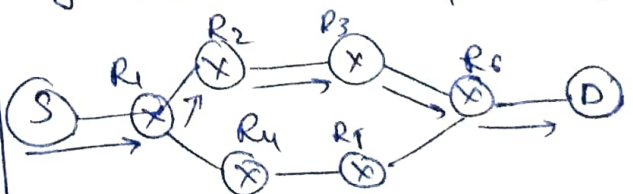
→ As the packet reaches destination it will not send ICMP message but if ICMP message is not received to sender it doesn't always means that packet reached destination, there is also a possibility of packet lost.

Packet reaches destination

→ If packet reaches destination then it will not send any ICMP, but to get Destination IP we have need to get an ICMP from destination. for this we attach the dummy port no (port no. that doesn't exist) in message. when that packet reaches destination, found that process (portno) doesn't found. then destination send ICMP with (destination port Unreachable)

ICMP packet lost

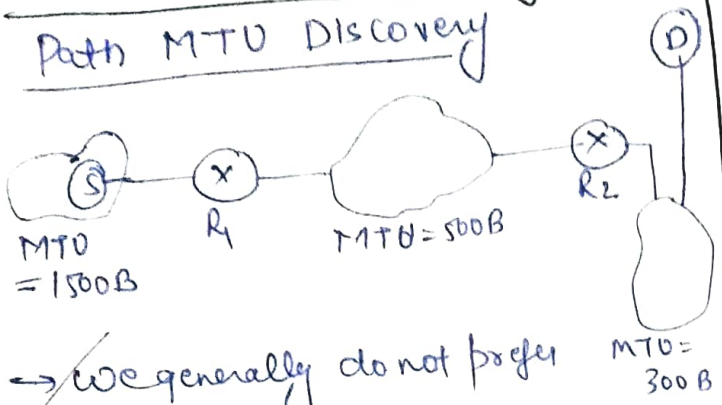
On running traceroute command if you get timeout in sequence means ICMP packet lost and you need to send packet again.



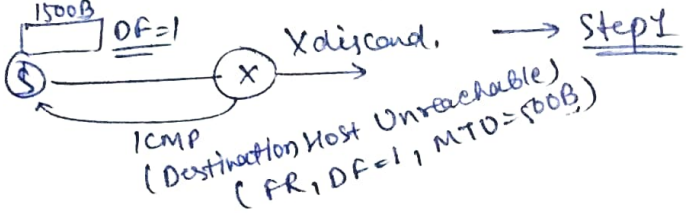
(R1, R2, R5, R6, D) If wrong route found packet will retransmit automatically

PMTUD – Application of ICMP

Path MTU Discovery

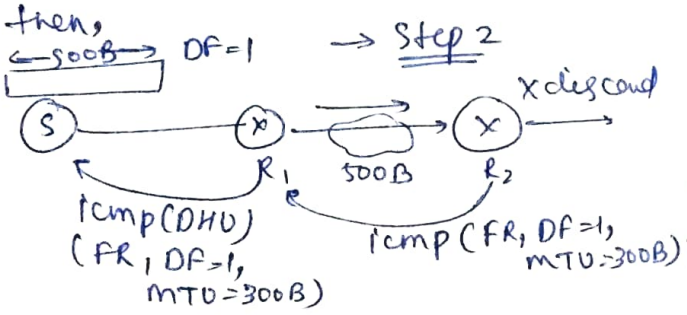


→ We generally do not prefer fragmentation at router, so to avoid this we can do that discover the min. MTU at path and make datagram packet at source according to this.



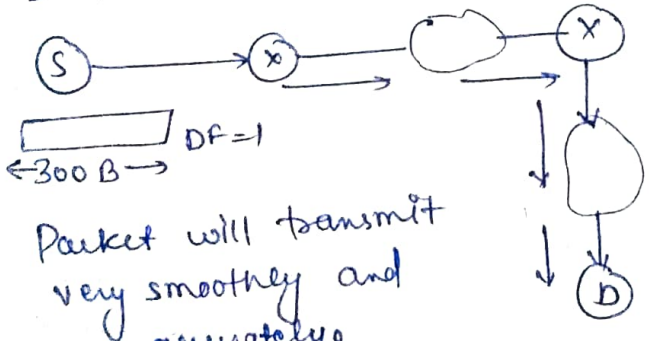
→ Step 1

then,



→ Step 2

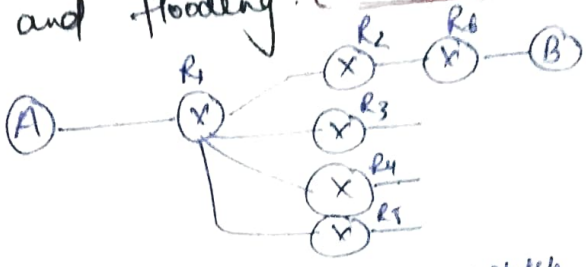
Step 3



Packet will transmit very smoothly and accurately.

So for more routers this process will repeat by OS continously

Difference between routing and flooding (Day-8)



How does router known which path is suitable for reaching B?

Ans– Routing table

Process of preparing routing table to make switching easier called routing.

→ If we dont have routing table then we do flooding means send packet to all the ways (direction) and packet will definitely reaches to destination.

Adv. of flooding

→ No routing is required.

→ Shortest path is guaranteed. (bcoz we take every path).

→ Highly reliable bcoz if any one path is down then atleast a path will definitely reach to destination.

Disadv. of routing

→ Routing table is required. due to this at each router some processing time wasted.

→ If shortest path fails then packet doesn't reach means 'not much reliable'

Disadv of flooding

→ Lots of packet arrive at destination (Duplication)

→ Single packet transmit on routes, so traffic is very high.

Adv. of routing

→ No duplicate packets

→ Not much traffic as compare to flooding

→ Military services uses flooding but normal Internet uses routing.

Routing Algo.

→ It can be of various types but basic two types are-

① Static

→ Manually prepared and upload offline

→ It can't be done in general bcoz any link or router can down so updation is tough.

→ Depend on traffic and topology they don't change (Disadv.)

② Dynamic

→ We generally go for it.

→ Done by router automatically

→ Routing table changes dynamically with change in traffic or topology.

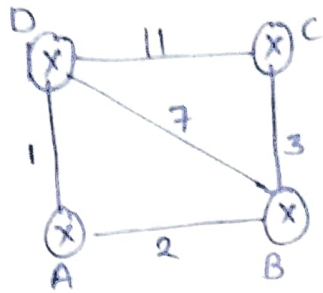
→ Ex →
DVR
LSR
} gmp for GATE

DVR - Distance Vector Routing
LSR - Link State Routing

Distance Vector Routing (DVR)

M.gmp

Shortest path from one router to another.



edge weight can be due to either distance, traffic, queing delay etc.

Every routing table have 3 colmn.

Destination	Distance	Next Node

Routing table at B: (local knowledge)

Dest.	Dis.	Next
A	2	A
B	0	B
C	3	C
D	7	D

Dist. can be node direatly attact to node, else ∞.

Routing table at A

Dest	Dist	Next
A	0	A
B	2	B
C	∞	—
D	1	D

At router C

Dest.	Dist	Next
A	∞	—
B	3	B
C	0	C
D	11	D

At router D

Dest	Dist	Next
A	1	A
B	7	B
C	11	C
D	0	D

So, we prepare local routing table in round-1

In second round we go for distance vector:

In computer science distance vector means array of distances.

Each router will exchange distance vector with all its neighbours.

At A : DV received from B, D

At B : DV from A, C, D

At C : DV from B & D

At D : DV from A, B & C

→ Due to full duplex DV can transmit in both direction parallely.

So, what happens at A,

from B: 2, 0, 3, 7

from D: 1, 7, 11, 0

] Using this A will compute new routing table

At A

A	0	A
B	2	B
C	5	B
D	1	D

→ Completely based on info. from neighbours not from prev. routing table

for A → B

$$= \min\begin{cases} A \xrightarrow[(2)]{} B + B \xrightarrow[(0)]{} B = (2) \\ A \xrightarrow[(1)]{} D \xrightarrow[(7)]{} B = (8) \end{cases}$$

$\min(2, 8) = 2$

for A → C

$$\min\begin{cases} A \xrightarrow[(1)]{} D + D \rightsquigarrow[(11)] C = 12 \\ A \xrightarrow[(2)]{} B + B \rightsquigarrow[(3)] C = 5 \end{cases}$$

→ means routes are direct connected [Edge]

⇝ means routes for which table is design not in direct connected [Path]

To solve (⇝) this we use Distance vectors.

A → D

$$\min\begin{cases} A \xrightarrow[(11)]{} D + D \rightsquigarrow[(0)] D = 1 \\ A \xrightarrow[(2)]{} B + B \rightsquigarrow[(7)] D = 9 \end{cases}$$

So, new routing table is changed as previous.

In round 2 we go for path with two edges to each another routers.

Similarly we can make table for all routers.

→

Procedure to use in exam

At A

from B: A 2, B 0, C 3, D 7 → AB = 2

from D: A 1, B 7, C 11, D 0 → AD = 1

(AB + BB) = (2 + 0) = 2

(AD + DB) = (1 + 7) = 8 | min(2, 8) = 2

(AB + BC) = min(5, 12) = 5

(AD + BC)

$\begin{pmatrix} AB + BD \\ AD + DD \end{pmatrix} = \min(9, 1) = 1$

A	0	A
B	2	B
C	5	B
D	1	D

AB or AD ki values ant DV में Add करके min. find करो.

At B → DV is from prev. table

from A	from C	from D
0	∞	1
2	3	7
∞	0	11
1	11	0
AB = 2	CB = 3	BD = 7

$$\begin{bmatrix} AB+AA = 2 \\ BC+CA = \infty \\ BD+DA = 8 \end{bmatrix} \min(2,\infty,8) = 2$$

$$\begin{bmatrix} \infty \\ 3 \\ 14 \end{bmatrix} \min = 3$$

A	2	A
B	0	0
C	3	C
D	3	A

we don't take DV of A from new table because all the DV's send to each router before computation starts.

Similarly we can do it for all edges (routers).

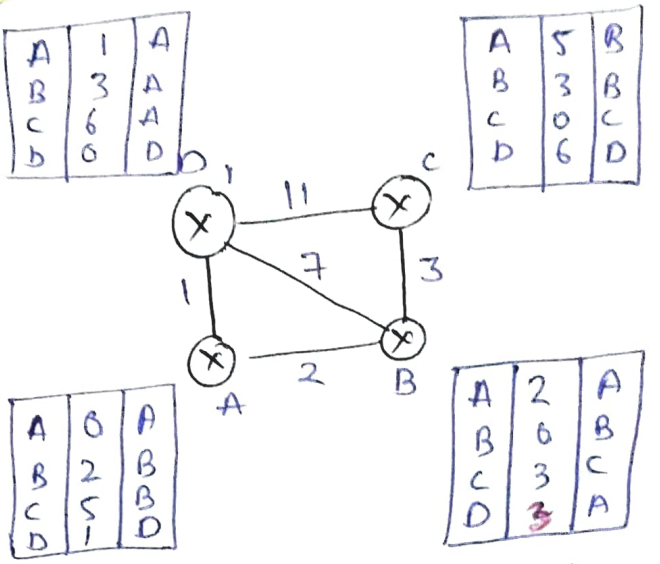
In round 2 we go for 2 edges. Since the graph we consider can have only at max 3 edges so we go upto round 3.

→ If graph have n-~~edges~~ vertices then go upto round (n-1)

In Gate exam Graph is small so do directly by seeing the graph.

→ No need to apply Algo.

Type of question asked



Q-1 Which route is never being used?

Ans D → C, D → B

Just see the next hop in each table and identify the path. After getting path from all the final DVR table you will get some path that have never used.

Q-2 Given, DV and values of AB, BD or anyone find out routing table

→ As we done previously in topic.

Q-3 Make the DVR final table for router.

Count to infinity [In case of doubt watch video again]

→ Drawback of DVR

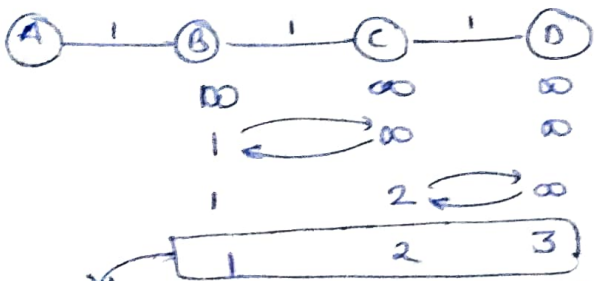
(Bad news spread slow, Good news spread fast)

Initial (A) is not connected so,

$d_{AB} = d_{AC} = d_{AD} = \infty$

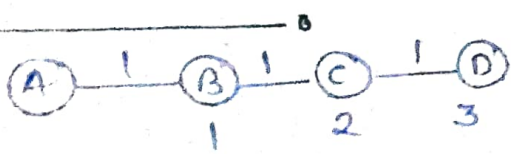
then suddenly after connecting (A) the image should be like,

after that (B) and (C) exchange info that (A) is in network and (B) says (A) is at distance 1 from me and (C) knows (B) at distance 1 from (C) then new updation should be like



this state will achieve when (C) and (D) exchange their info.

Since the updation process for good news ((A) is added to N/w) is spreading very fast



Bad News: (A) gets out of network

Now if we exchange info of (B) and (C) means (C) says to (B) that, I take you to (A) in just 2 hop and (B) knows it is 1 distance away from (C) So totally it is 3 distance away from (A) via any other path, then it updates as.

→ Here (B) doesn't knows that (C) doesn't have a path through (B) only

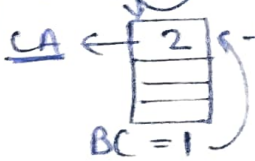
At C

A	2	B

At B

A	∞	–

when (C) sends DV to B then

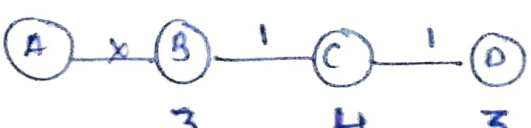


then in B

A	3	C

(BC + CA) = 1 + 2 = 3

B to A using C take 3



At C : DV from B and D

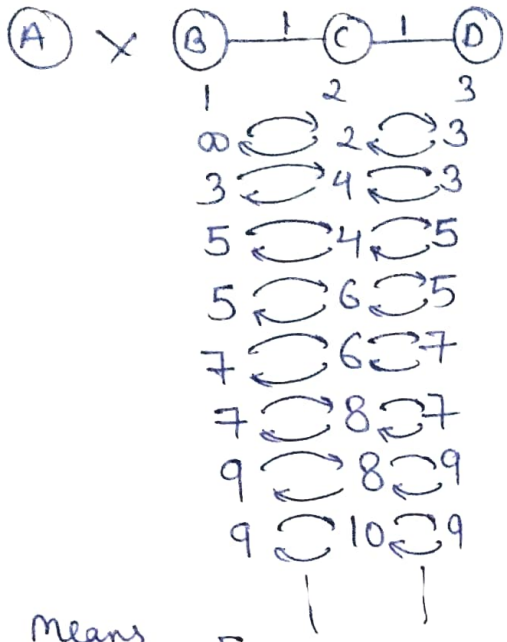
BA ∞ CA = 1

DA 3 CD = 1

$\min\begin{pmatrix} BA + CB \\ DA + CD \end{pmatrix} = 4$

At C

A	4	B



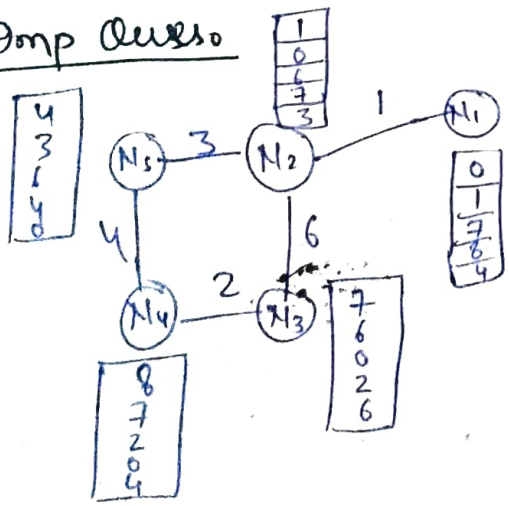
Means End no. in range [∞ ∞ ∞]

→ So Bad news spread slow.

→ Also called count-infinity problem.

→ This problem is due to reason that we send only DV in DVR and if we can send next hop too then this problem will not occur but it creats a burst of data.

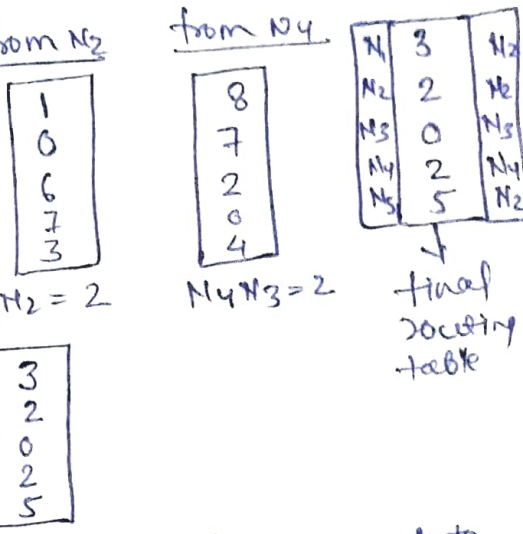
Imp Ques.



→ Given, DV are finally Converged after 4 rounds.

If distance (N_2-N_3) down to 2 from 6 then after 1 round of exchange what will be the DV at N_3.

At N_3 from N_2 from N_4



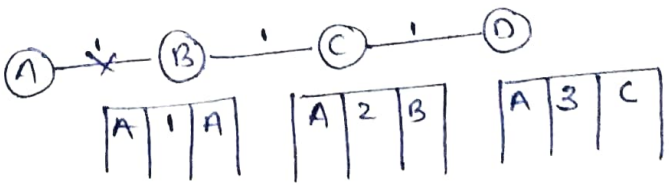
Ans

Note → Please do read the count to infinity problem from anywhere and do atleast 3 questions, other than Gate so, that you can get an idea.

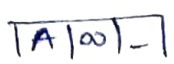
Split Horizon :- Solution to count to infinity problem.

→ Count to infinity problem might put the packet in infinite loop.

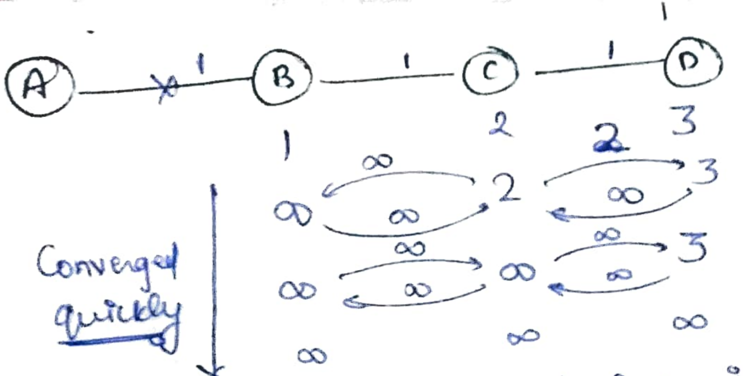
Using split horizon we can converge faster.



After down link,



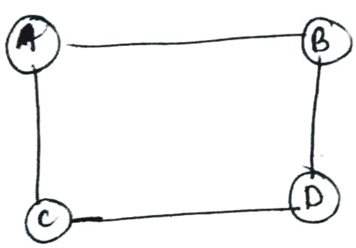
If link get down and C is saying that to B, I can take you to A via (B) then it will not consider, because (C) is already depend on B. So (C) will send (∞) to B instead of its actual value 2,



Converged quickly

∃ → C doesn't depend on D but D depends on C to reach A

→ By default split horizon is implemented means you have to apply count to infinity but if they given DVR implemented with split horizon then apply this.



A	1	A
B	0	B
C	3	A
D	4	D

If B sends its DV's to (A) and (D) then it will not send the actual value at with the corresponding node it depends on

Ex At A

∞
0
∞
4

B → A] via A
B → C]

means if next hop in table is that node to which you are sending DV then send ∞ instead of actual value.

Similarly at D

1
0
3
∞

→ This way split horizon work and it save us from loops.

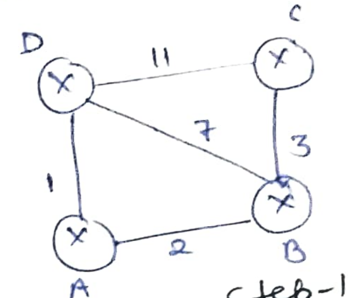
① Count to ∞
② Convergence is slow
③ There could be loops

Drawback of DVR.

→ Split horizon solve the problem of loops but the convergence is still slow

Link-state routing (LSR)

LSR is superior to DVR.



Step-1

Every state or router create link state packet

At B

Seq. no.	
TTL	
A	2
D	7
C	3

→ router links with (B)

→ Link state packet

At C

Seq. NO	
TTL	
D	11
B	3

At A

Seq. no	
TTL	
B	2
D	1

At D

Seq. no	
TTL	
C	11
B	7
A	1

Step-2 In LSR every node can flood the information to every routers.

→ Unlike DSR, LSR is not limited to connected routers only.

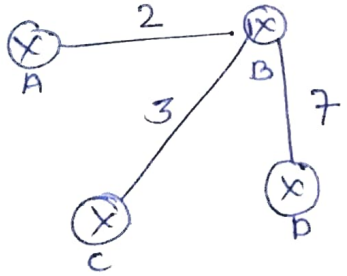
Each node will receive DVIs from all nodes.

→ So DVR based on local knowledge (neighbours) but LSR based on global knowledge (all routers)

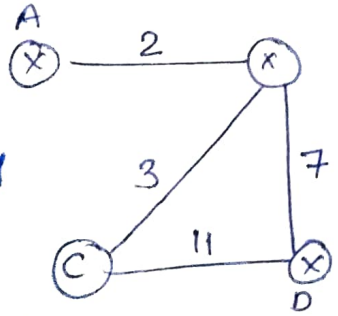
So,

At A: (A) will receive DVIs from all the other (3) nodes,

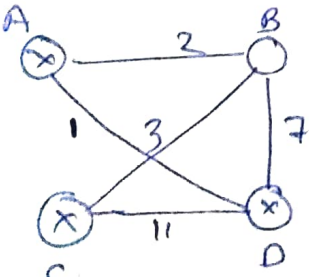
from B (A) will get



from C (A) will get including (B)



from D (A) will get including (C)



Above graph is similar to previous graph but seems to be change due to exchange in position of (C) and (D)

→ we consider undirected but it can be directed too.

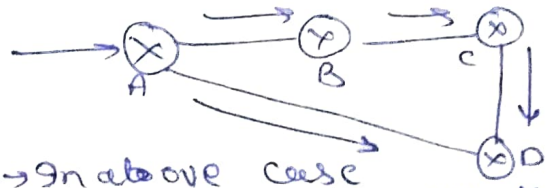
→ This way (A) will create a complete graph or tree in database.

→ (A) will apply single source shortest path algorithm. Using Dijkstra Algo. we make routing table at A as

A	0	A
B	2	B
C	5	B
D	1	D

→ Converges faster than DVR

Problems :- Heavy traffic due to flooding.



→ In above case it might get possible that A will get similar packets

→ Other problem can be that packet from A → D reaches faster but old packet and A → B → C → D packet reaches slowly but the packet is new one. To resolve this problem we have seq no. in routing table.

→ Old packet get discarded.

At router makes a database as

Route	latest
B	10
C	8

If packet to B needs to send with seq. no 18 or ≥ 10 then we send it and update value in table as 10 ↔ 18.

(latest seq. no ≤ need seq. no)

→ If packet with same seq. no came then it will get discarded.

→ TTL is going to solve the problem of infinite loop - if occurs.

Assume the case

Route	Latest
B	15
C	8

If new data packet comes with (B, 15)

↓

00001111 (15)

and in seq. no due to bit corruption it becomes

00011111 (32)

then routing table will get updated to (B, 31) and if new packet comes with (B, 16) it will get discarded due to error in seq. numbers

→ This might reject lot of valid packet. So to solve this problem.

We add lifetime in routing table

Route	Latest	lifetime
A	16	10ms

→ So if packet with seq. no of 15 comes it will get accepted after 10ms.

Transient Problem → occurs for small duration and get resolved itself.

(1) **Blackhole Problem** → If link is down then all packet received come to black holes (corrupted)

→ looping is also a transient problem.

→ But Looping is persistent problem for DVR.

DVR vs LSR (Imp).

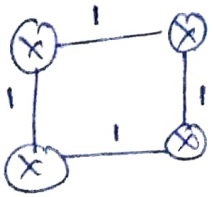
DVR	LSR
(1) 1980's	(1) 1990's
(2) Low Bandwidth.	(2) High BW because due to flooding.
(3) Based on local knowledge	(3) Global knowledge
(4) Bellman ford implementation	(4) Dijkstra
(5) Less traffic	(5) High traffic
(6) Periodic updates	(6) Periodic Updates
(7) Slow convergence	(7) Converges faster
(8) Count to infinity	(8) No count to infinity.
(9) Persistent looping	(9) Transient Looping
(10) Routing Information Protocol (RIP)	(10) OSPF: Open shortest Path first

Both are routing protocols.

DVR and LSR are general concept but RIP and OSPF are their implement version.

RIP:-

Metric → (Hop Count)



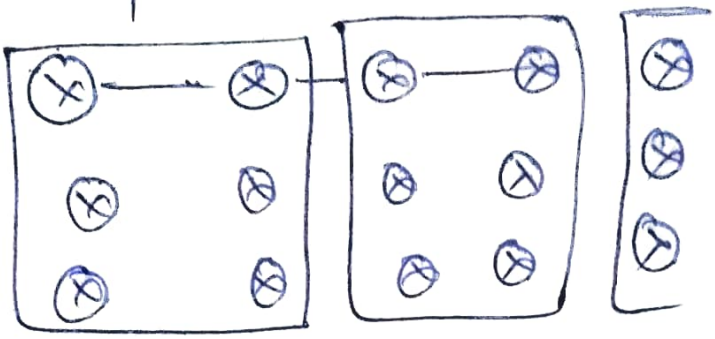
Every edge will consider as weight 1.

In DV we put ∞ in DVR but in RIP we use 16 instead of ∞.

→ DVR (simple computation)

OSPF :-

LSR have complex computation.



Devide routers in regions. then flooding get restricted to regions.

and each region have border gateway (BG) BG sends the flooded packet to (Area 0 or Backbone zone).

Area 0 used as intermediator

→ Hop count is not need to use in table.

→ EIGRP is a hybrid protocol combines both RIP and OSPF.