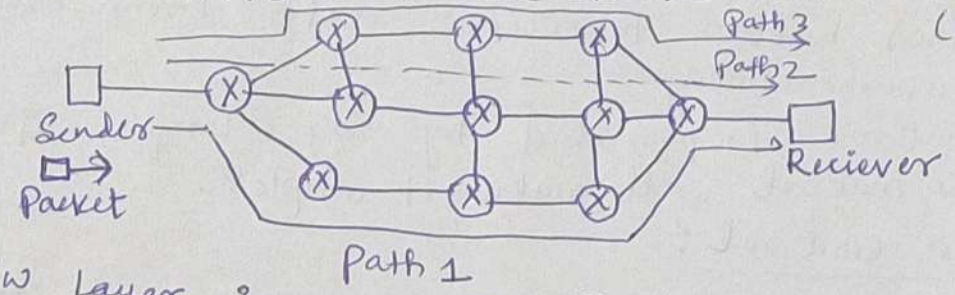
Chapter 1

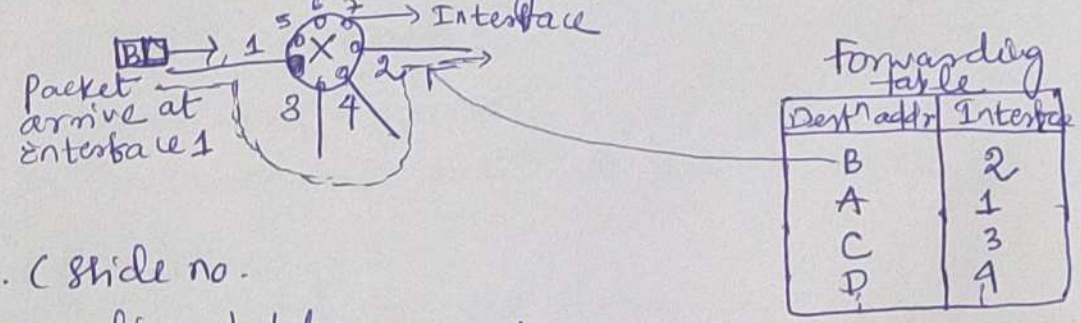
NETWORK LAYER

Network Layer Services:

1. **Packetizing:**
   - Encapsulating the payload in a n/w Layer packet at source and decapsulating the payload from n/w Layer packet at destinatn
   - Fig. 4.1 (Slide no.)
   - To carry a payload from source to destination without changing it or using it
   - Routers in the path are not allowed to decapsulate unless packet needs fragmentation.
2. **Routing:**
   - N/w Layer is responsible for routing the packet from source to destinatn.
   - There is more than one route from source to destinatn (Path 1, Path 2, Path 3)



   - N/w Layer is responsible for finding the best route among these.
   - Best route is defined by using routing protocols.
3. **Forwarding:**
   - It is defined as the action applied by each router when a packet arrive at one of its interface.
   - forwarding is done with help of forwarding table
   - Each router have forwarding table.



   - Fig. 4.2. (Slide no.
   - In forwarding table, each interface is provided with destinatn address.
   - In the packet destinatn address will be checked and then packet will be forwarded to corresponding interface.

4. Error Control:

- No error control is done in n/w Layer because
  * Packet in n/w layer may be fragmented at router if the packet is large size. Hence no need for error correction.
- Only a checksum field is added to control error of header but not whole datagram
- ICMP protocol provide error control if datagram is discarded.
  (ICMP - Internet control message protocol)

5. Flow Control:

- N/w Layer does not provide any flow control because
  i. No error control. Hence the reciever will rarely be overwhelmed
  ii. Upper layer uses buffer to recieve data from n/w Layer. So no overwhelming.
  iii. Flow control is provided by upper Layer. Another level of flow control will make it complex.

6. Congestion Control:

- Some congestion control schemes are there in n/w Layer although it is not implemented in internet.
- Congestion control is provided by upper Layer because IP protocol of n/w layer is unreliable.

7. Quality of Service

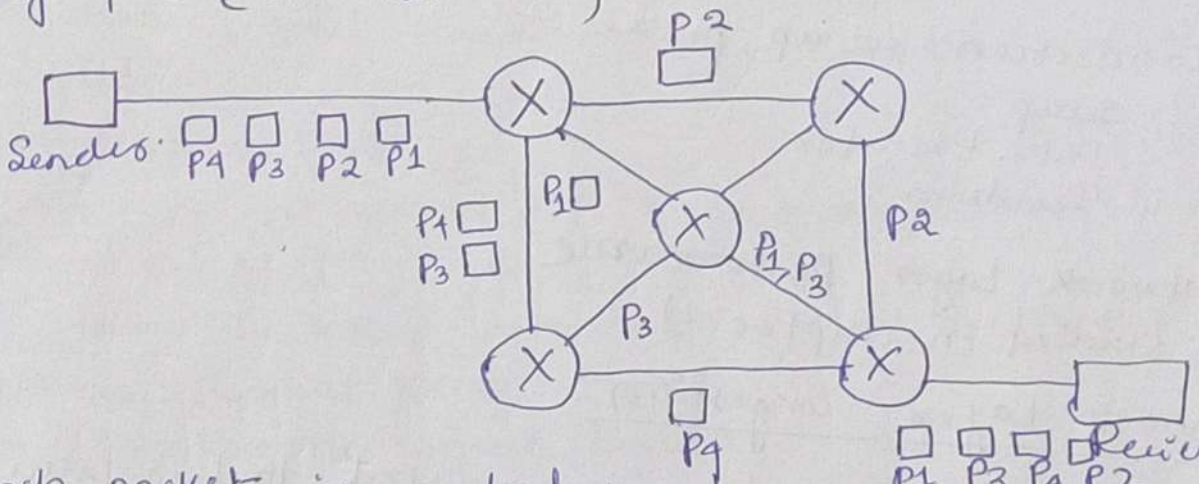
8. Security.

Packet Switching :-

- A router is a switch that create connection bet^n i/p port and o/p port.
- Switching two types
  i. Circuit Switch
  ii. Packet "
- Only packet switch is used at n/w Layer.
- Packet switch n/w has two approaches -
  a. Datagram approach
  b. Virtual circuit approach.

a. Datagram approach :

- It is a connectionless servile.
- Packet in a message may or may not travel the same path to their destinat^n.
- Fig. 4.3 (Slide no.)

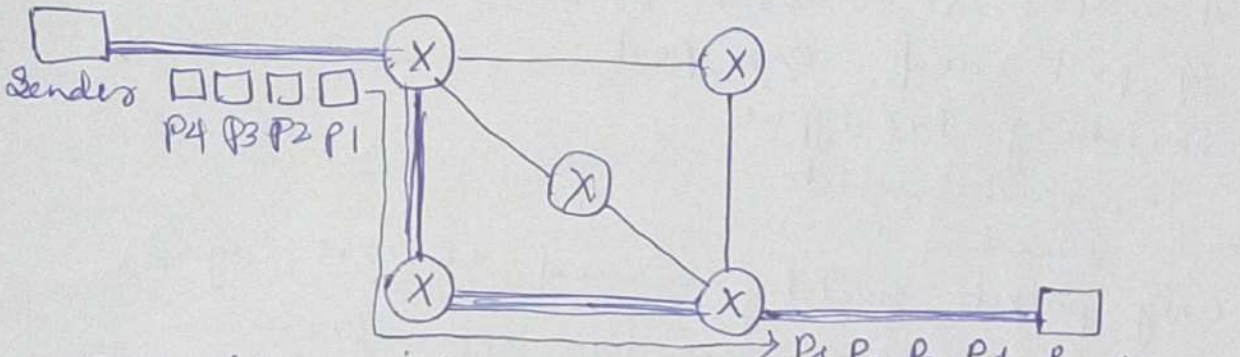


- Each packet is routed based on source and destinat^n address.
- Fig. 4.4 (Slide no)

b. Virtual circuit approach :

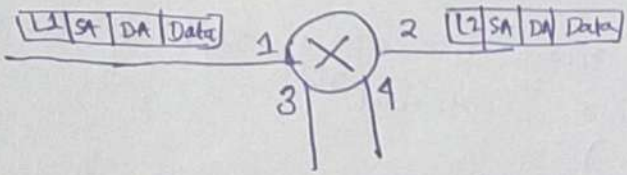
- Connection-oriented approach.
- There is relationship bet^n all packets belong to same message.
- Fig. 4.5 (Slide no
- After connection set-up the datagrams follow same path.
- The packet contains source ip, destinat^n ip, flow level, virtual circuit identifier.

- Virtual circuit identifier define the virtual path packet should follow.



Forwarding table

Incoming		Outgoing	
Port	Label	Port	Label
1	L1	2	L2
⋮	⋮	⋮	⋮



(Fig - 4.6) (slide no.)

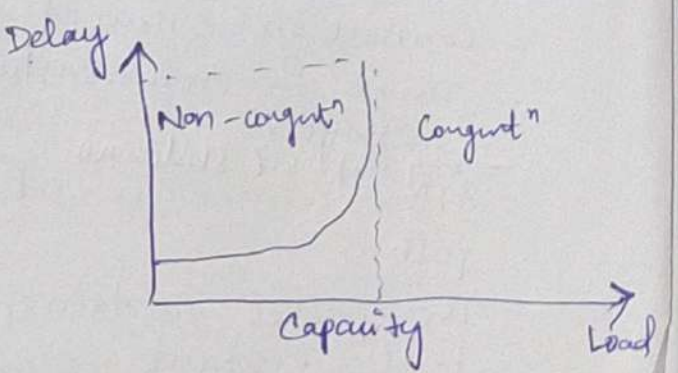
-

- Connection set up phase -
  i. Setup.
  ii. Data transfer
  iii. Teardown.

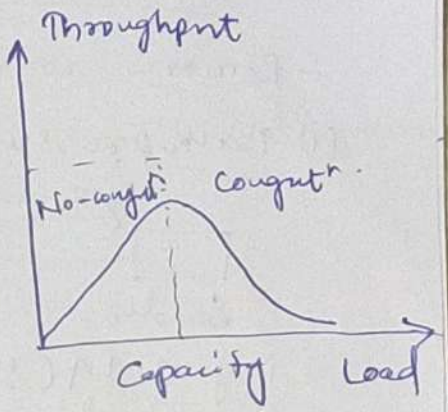
* Network Layer performance
  (Studied in chapter 1)

Network Layer Congestion:

- Congestion at the n/w layer is related to two issues
  - Throughput
  - Delay.
- Fig. 4-13 (Slide no)
- Delay.
  * when load is less than capacity, delay minimum
  * when load reach capacity, delay increase sharply
  * Load greater than capacity, delay infinite.



- Throughput
  * Load below capacity, throughput increases.
  * Load reaches capacity, throughput remain constant
  * Load exceeds capacity, throughput decreases.



(A) Congestion Control :-

- It refer to the technique and mechanism that can either prevent or remove congestion.
- Two mechanism
  1. Open loop congestion control (Prevention)
  2. close " " " (Removal)

1. Open Loop :

- It prevent congestion before it happen.
- Congestion control handled by either the source or destination.

(i) Retransmission policy :
- Packets are retransmitted if lost or corrupted.

(ii) Window policy :
- SR window is better than GBN windows for congn control

(iii) Acknowledgement Policy :
- Good ACK policy can prevent congn.

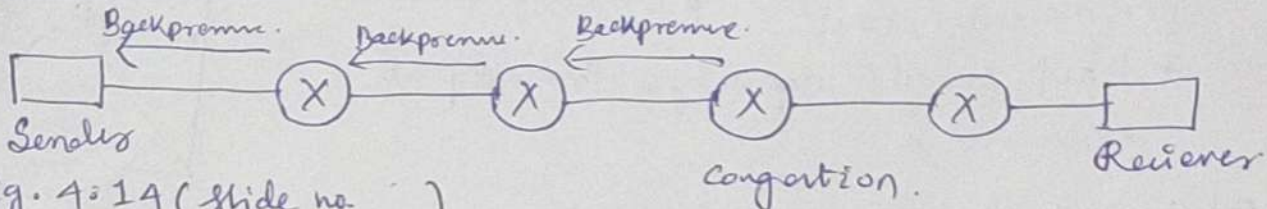
(iv) Discarding Policy :
- Router should implement good discarding policy.

(v) Admission policy :
- Switches in a flow first check the resource requirement of a flow before admitting it to n/w,

2. Closed Loop:

- Remove congestion after it happen.

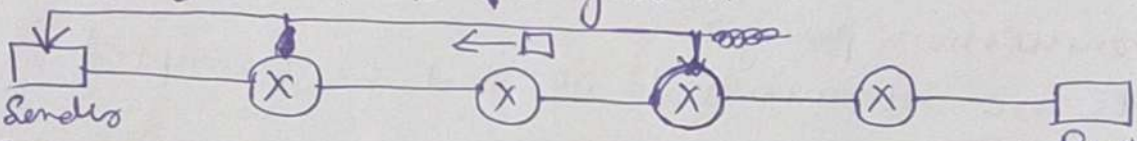
(i) Backpressure:



- Fig. 4.14 (Slide no.)
- Congested node stop recieving data from upstream node.
- May cause upstream node congested. So it reject data from its upstream node and so on.
- Backpressure is a node-to-node control that starts with a node and propagate in opposite direct^n of data flow.
- Applied to only virtual circuit n/w.

(ii) Choke Packet:

- It is a packet sent by a node to source to inform it to congestion.



- Fig. 4.15 (Slide no.)
- Router which in congested directly send a choke packet to source to say about the congestion.

(iii) Implicit Signaling:

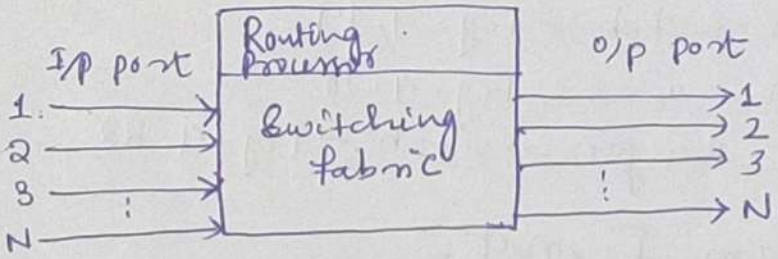
- No communicat^n bet^n congested node and source
- Source guess that there is congest^n in n/w from other symptoms such as no acks

(iv) Explicit Signaling:

- Node that experience congestion can explicitly send a signal to source or destination.

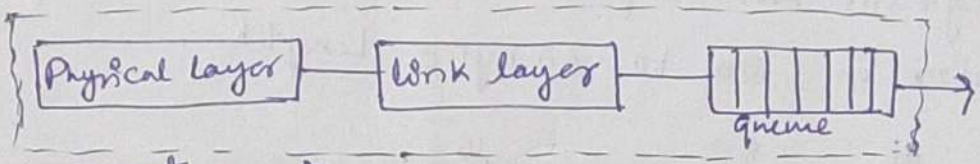
Structure of a Router :

- Router accept packet from one of input port, uses forwarding table to find the output port.
- Router has 4 components -
  * Input Port
  * Output "
  * Routing processor
  * Switching fabric
- Fig. 4.16 (Slide no.)



1. Input Port :

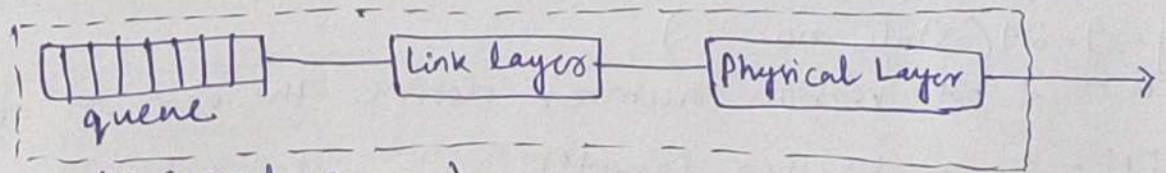
- It perform the physical layer & link layer funct"



- Fig. 4.17 (Slide no.)
- Bits are constructed from the signal
- Packet decapsulated from frame and checked for error.
- It has buffers to hold packets

2. Output Ports :

- Perform same function as i/p port but in reverse order.



- Fig. 4.18 (Slide no.)
- Outgoing packet are queued, each packet encapsulated in a frame and converted to bits and then signal is created to be sent

3. Routing Processor:

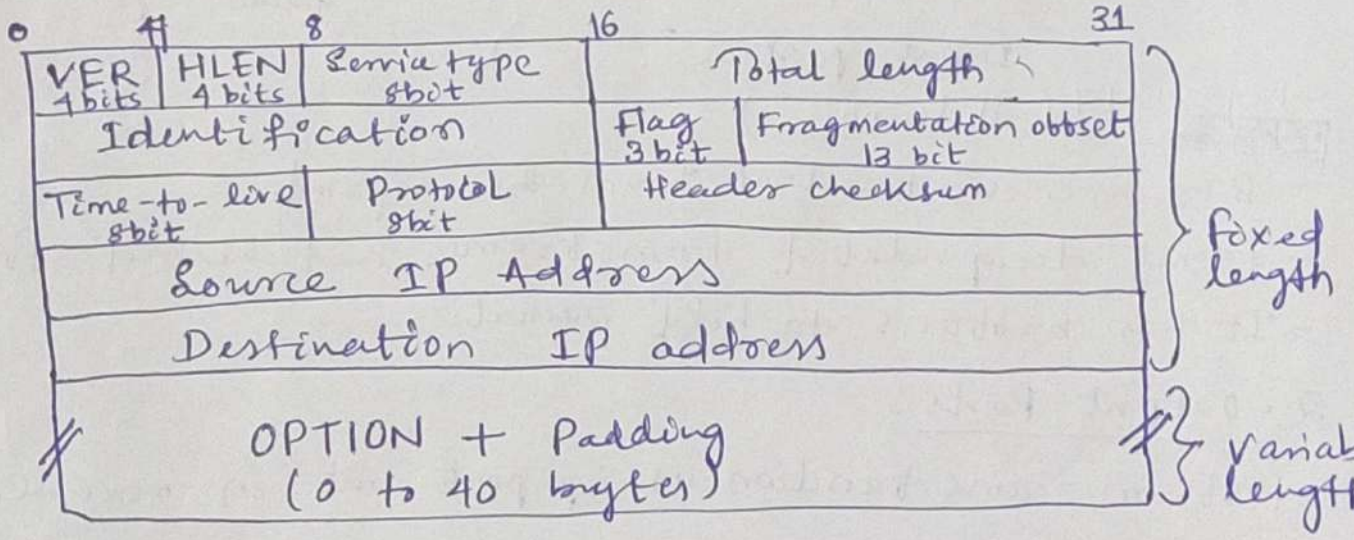
- It perform the function of network layer.
- Destinat° addren is used to find next hop and o/p port number. It is called as table look up.

4. Switching fabric:

- Moving the packet from i/p queue to output queue.
- This speed affects the size of i/p/o/p queue and overall delay in packet delivery.
- Switching fabric used in router
  - * Crossbar switch - Fig. 4.19
  - * Banyan " - Fig. 4.20
  - * Batcher Banyan switch - Fig. 4.22.

IPv4 Datagram Format:

- Packet used by IP are called datagram
- Datagram has 2 parts - Header & payload (Data)
- Header is 20 to 60 byte length.



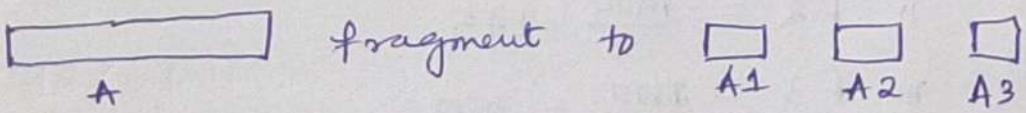
-- Fig. 4.24 (Slide no.)

- VER - 4 bit version number define the version of IP
- HLEN - " " header length " total length of header in 4 byte word.
- Service type - How the datagram should be handled.

- Total length - 16 bit field define total length of the datagram.
- Identification, Flag and offset - Related to fragmentation when a large datagram is divided. will be discuss there.
- Time-to-live - It is a 8 bit field used to control the maximum number of hops visited by the ~~router~~ datagram. (TTL)
  * Each router that panes the datagram decreaments this number by one.
  * When TTL become zero, datagram discarded.
- Protocol - It is a 8-bit field which define which protocol the payload should be delivered (UDP or TCP)
  • Fig. 4.25 (Slide no.) (17) (6)
- Header checksum - Calculated at each router.
- Source and destinatn IP addrss.
- Options - 0 to 40 bytes.

Fragmentation

- Maximum transfer Unit (MTU) - Fig. 4.26 (Slide no.)
- Size of datagram must be less than MTU
- So large datagrams are fragmented to make it possible for it to pass through network called as fragmentation.
- Reassembly of datagram occur at destinatn.
- When datagram fragmented, three fields are mentioned flag, identification and offset.
- Identification -
  * Suppose datagram A fragmented into A1 A2 A3



  * All the fragments have same identification
  * Suppose identificatn value is 10, It is same for all three fragments A1, A2 and A3.

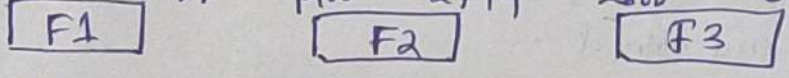
- Flags -
  * 3 bit flag is used (.
  * Left most bit is reserved or unused.
  * Second bit is called D bit or do not fragme bit.
    - If D = 1, datagram will not be fragmented
    - " D = 0, " " " "
  * Third bit is M bit or more fragment bit.
    - If M = 1, datagram is not Last fragment, more fragments are there
    - If M = 0, datagram is last fragment.
    - Among A1, A2 and A3 fragment

      A1, M = 1
      A2, M = 1
      A3, M = 0

- Fragmentation offset
  * It shows relative position of the fragment w.r.t. other fragments.
  * offset is calculated by deviding the first byte of fragment by 8.

Example

- Suppose size of datagram is 4000 Byte
- MTU = 1400
- Datagram will be fragmented to three parts



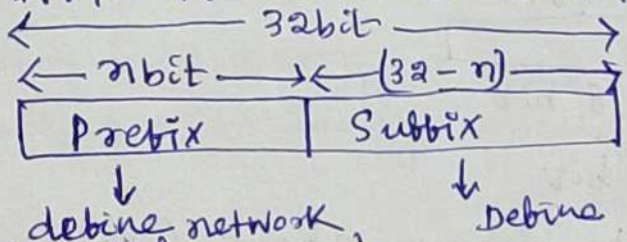
- Offset of F1 = $\frac{0}{8} = 0$
  " " F2 = $\frac{1400}{8} = 175$
  " " F3 = $\frac{2800}{8} = 350$

- Fig. 4.27 (Slide no.)

Q - Consider sending a 2400-byte datagram into a link that has an MTU of 700 bytes. Suppose the original datagram stamped with identification number 422. How many fragments are generated? What are the values in various fields in the IP datagram generated related to fragmentation?

IPv4 Addresses :-

- IP addresses are the identifiers used to identify the connection of each device to the internet.
- IPv4 means IP addren version 4.
- It is a 32 bit address.
- It is the addren of connection not host or router.
- Addren space is the total number of addren used by protocol.
- IPv4 uses 32 bit for addren, so total 2^{32} addrens are there.
- Three notations are used to represent IP addren -
  (i) Binary (10000000 00001011 00000011 00011111)
  (ii) Dotted decimal (128 . 11 . 3 . 31)
  (iii) Hexa decimal (800B031F)
- Fig. 4.29 (Slide no.)
- 32 bit IPv4 addren divided into two parts -



- Fig. 4.30 (Slide no.)
- Addreming two types -
  (i) Classful Addreming
  (ii) Classless "

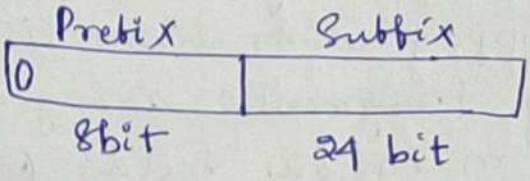
Classful Addreming :-

- Oldest version
- Prefix length was fixed.

- Five classes of classful address.
- Fig. 4.13 (Slide no.)

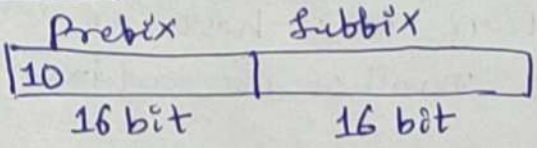
a. Class A:

*



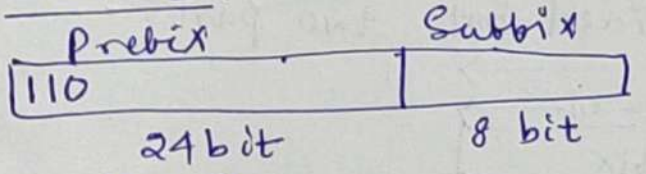
* N/w or prefix length 8 bit
* Among 8 bit the first bit is 0 which define the class.
* Rest 7 bit used for n/w identifier.
* So total 2^7 n/w are possible in the world.
* Total number of host in each n/w is 2^{24}.

b. Class B:



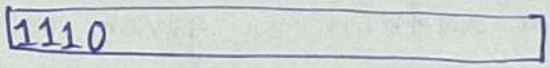
* N/w length is 16 bit
* Among them first two bits are reserved for class (10)
* So 14 bit are there for n/w identifier
* Total 2^{14} n/w are possible
* " 2^{16} host in each n/w.

c. Class C:



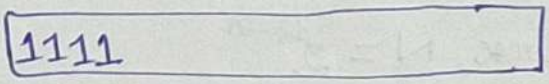
* N/w length is 24 bit
* 3 bits are used to define class (value is 110)
* So total 21 bit for n/w.
* Total 2^{21} n/w are possible
* Total 2^8 host in each n/w possible

d. Class D:



* Not divided into prefix & subbix.
* Used for multicasting.
* First 4 bit used to define class (value 1110)

e. Class E:-



* First 4 bits are used to define class (value 1111)
* Reserve for feature use.

- Drawbacks of classful addren -
  1. Class A addren has only 2^7 n/w addren. but 2^{24} host in each n/w. If the organizat^n does not have many host addrens will be wasted
  2. Class B is designed for midsize organizat^n but host addren will be unused.
  3. Class C has only 2^8 host, so can not be used for large organizat^n.

  There are called as addren depletion.
- Subnetting and supernating are sol^n to addren depletion.
- Subnetting - Dividing a large n/w to small n/ws.
- Supernetting - Combining small n/w to form a large network.
- Advantage of classful addren is that class of the addren can be easily found.

Classless Addressing:-

- With the growth of internet, Large addren space was needed
- Fig. 4.33 (Slide no.)
- Two sol^n - (i) length of addren can be increased (IPv6)
  (ii) Class less addren

- Classless addressing - No classes, addresses are given as blocks.
- SLash Notation:
  * The prefix length in an address is given with slash notation.

    a.b.c.d/n

    12.24.76.8/8
  * This is called as classless interdomain routing or CIDR.
- Fig. 4.33 (Slide no.)
- Number of address in a block, $N = 2^{32-n}$ where n = prefix length
- To find the first address, keep n leftmost bits and set (32-n) rightmost bit to 0.
- To find the last address in a block, keep n left most bits and set (32-n) right most bit to 1.
- Fig. 4.34 (Slide no.)
- A block of address assign to a n/w, 2 address are reserved. (one for n/w id and one for broadcast id)
- First address of a block is used as n/w id.
- Last " " " " " " broadcast id.

Example 4.1 (Slide no.

A classless address is given 167.199.170.82/27. Find the number of address, first address and last address.

* Address 167.199.170.82/27
* Prefix length 27
* Number of address in block, $N = 2^{32-n} = 2^{32-27} = 32$
* first address:

10100111 11000111 10101010 01010010 - Address given

Keep 27 bits as it is, set (32-27) bit to 0.

10100111 11000111 10101010 010 00000 → first address

27 bit 5 bit

- Last Address -

Keep 27 bit as it is, set (32-27) bit to 1

10100111 11000111 10101010 010 11111

27 bit 5 bit

Address Mask :

- It is a 32-bit number in which n leftmost bit are set to 1 and rest bit are (32-n) are set to 0.
- It can be used to compute N, first address and last address.
- Mask of a address .167.199.170.82/27 is

11111111 11111111 11111111 11100000

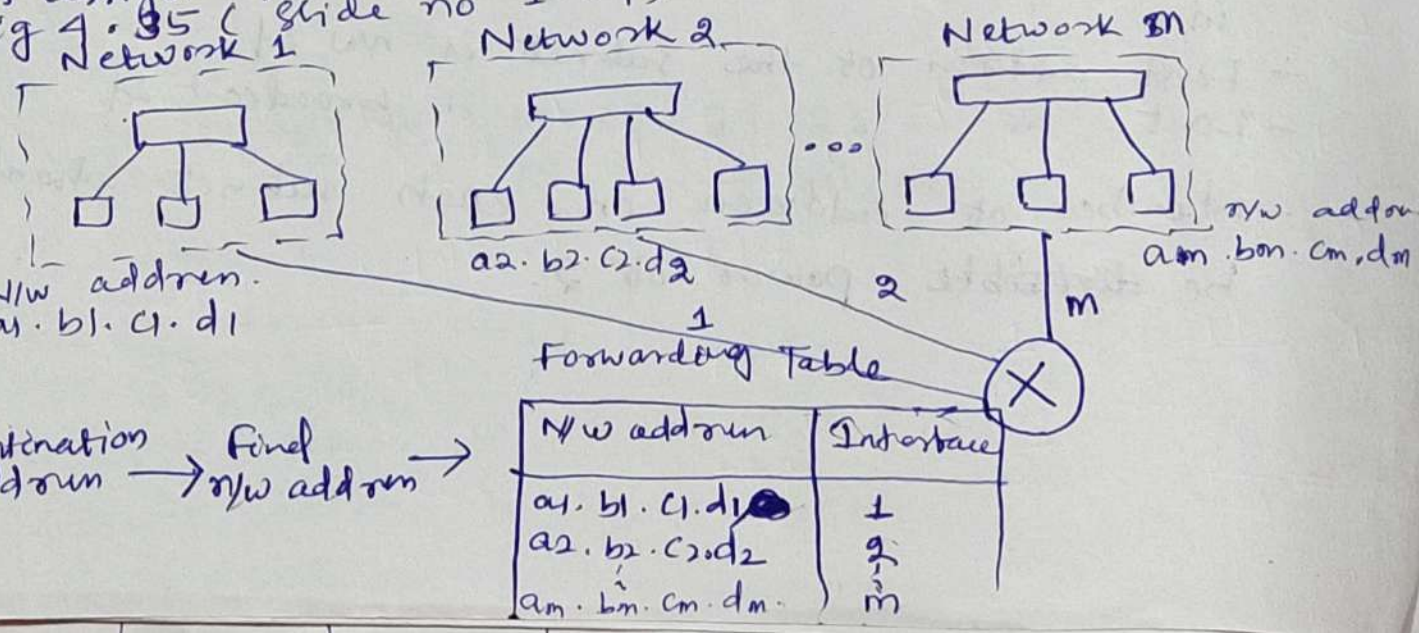
255 . 255 . 255 . 224.

- Number of address in a block, N = NOT (Mask) +1
- First address = (Any address in block) AND (Mask)
- Last " = (") OR [NOT (mask)]

Example - 4.2 - Book (Slide no.)

Network Address :

- The first address of a block of address is assign to n/w. (Slide no 1.59)
- Fig 4.35 (



- **Block Allocation:**
  - ICANN is responsible for assigning block of address to organizations.
  - It also assign block of address to ISP, and ISP assign to individual user.
  - Two restrictions during assigning block
    1. Number of requested address N should be in power of 2.
    2. Requested block needs to be allocated where contiguous number of available address are there. First address need to be divisible by number of address.
  - Example 4.4 (slide no.).

Subnetting:

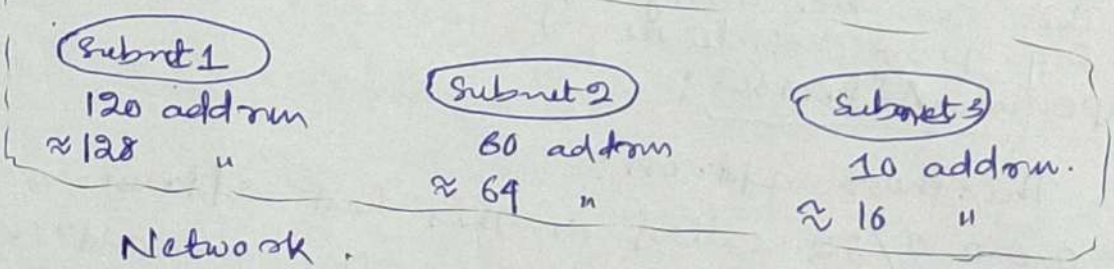
- In an organizat° level of heirarchy can be created using subnetting.
- Suppose total address assign to an organization is N. and prefix length is n.
- Total number of address assign to each subnet, N_{sub} and prefix length is n_{sub}.
- So $N_{sub} = 2^{32 - n_{sub}}$

  $\Rightarrow n_{sub} = 32 - \log_2 N_{sub}$
- Each subnet will have a n/w id and broadcast id.
- First address of the subnet is n/w id
- Last " " " " " broadcast id.
- Number of address in each subnet should be ~~divisible~~ power of 2.

Example - 4.5 (Slide no.)

Q - A organizat" granted a block of addrm which begin with 14.24.74.0/24. Organizat" need to ha 3 subnets - one with 10 addrm, one with 60 ad one with 120 addrm. Design the subnets.

Ans -



- Prefix length $n = 24$
- Total addrm in block, $N = 2^{32-24} = 256$
- First addrm/network id is 14.24.74.0/24
- Last " / broadcast id is 14.24.74.255/24.
- Three subnets are designed

a. Subnet 1

* 120 addrm required which is not power of 2. So 128 addrm will be assigned. $N_{sub1} = 128$
* Subnet mask, $n_{sub1} = 32 - \log_2 N_{sub1} = 32 - 7 = 25$
* First addrm of subnet, 14.24.74.0/25
* Last " " " , 14.24.74.127/25

b. Subnet 2

* 64 addrm = N_{sub2}
* $n_{sub2} = 26$
* First addrm, 14.24.74.128/26.
* Last " , 14.24.74.191/26

C. Subnet 3

* 16 addrm = N_{sub3}
* $n_{sub3} = 28$
* First addrm, 14.24.74.192/28
* Last addrm, 14.24.74.207/28.

4.36 (Slide no.)

Addren Aggregation:

- Addren summarization
- Block of addren are combind to create a lasger block.
- This can be assigned to ISP.
- Fig. 4.37 (slide no.)

Special Addrenes:

1. This host addren:
- 0.0.0.0/32 called as this host addren.
- Used when host do not know its own addren.

2. Limited broadcast addren:
- 255.255.255.255/32.
- Used when host or router needs to send datagram to all devices.

3. Loopback Addren:
- 127.0.0.0/8
- A packet never leaves the host if this addren is used.

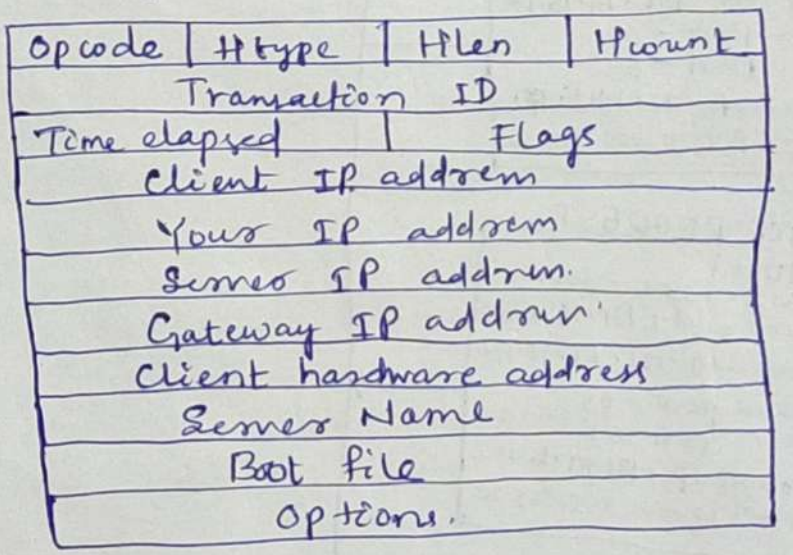
4. Private Addren
- four blocks used for private addren
  10.0.0.0/8
  127.16.0.0/12
  192.166.0.0/16
  169.250.0.0/16

5. Multicast Addrenes -
- 224.0.0.0/4 used for multicast.

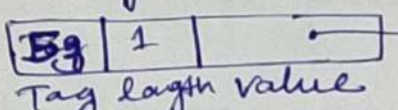
11)

Dynamic Host Configuration Protocol (DHCP)

- After an organization obtaining a block of IP addrsn, it anign the addrsn.
- Automatic anignment of IP addrsn can be done through DHCP.
- DHCP is called plug and play protocol.
- DHCP can anign IP addrsn permanently, temporary or on-demand.
- DHCP message format - Fig. 4.38

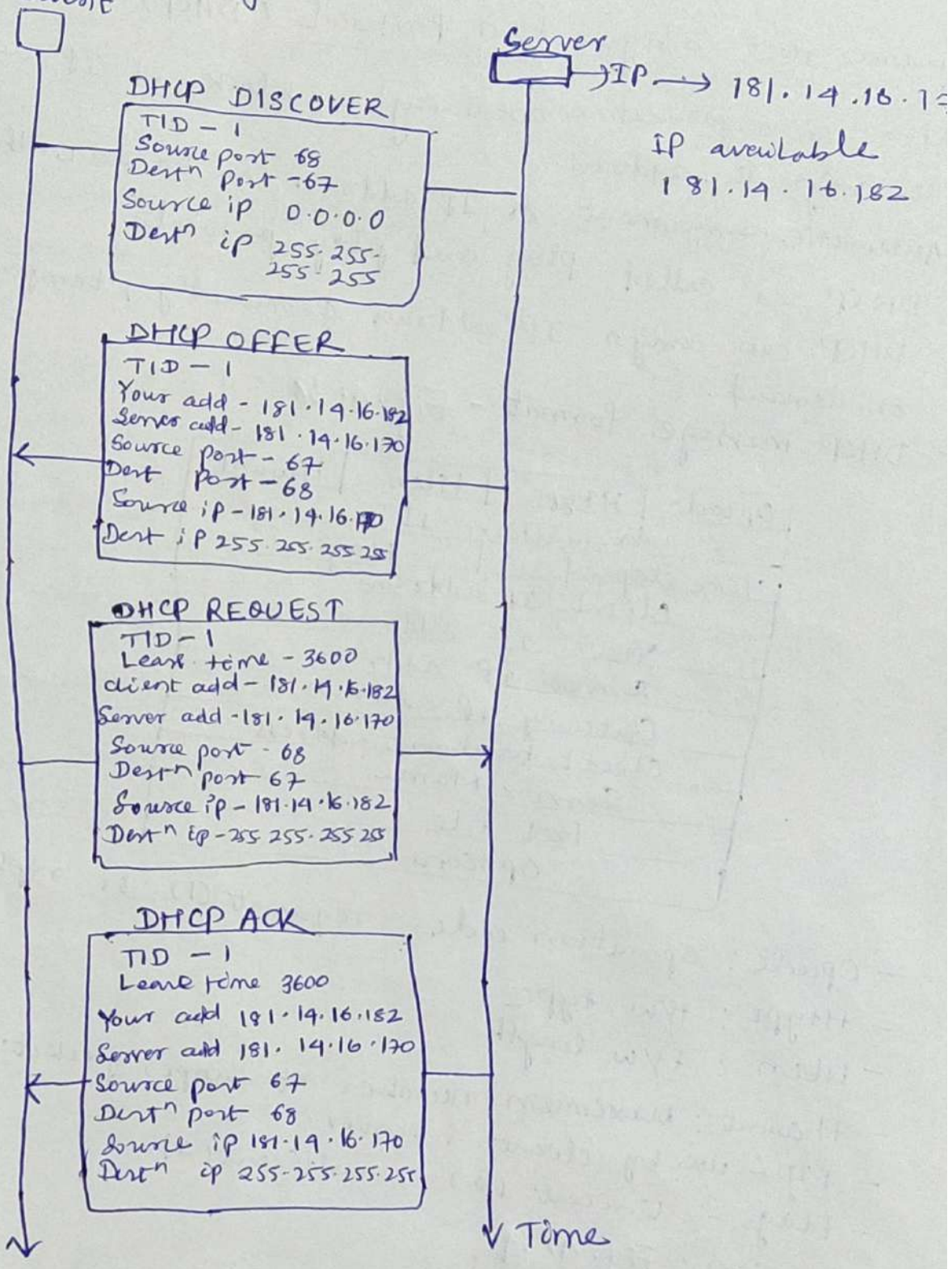


- Opcode : operation code, request (1) or reply (2)
- Htype : H/w type
- HLen : H/w length
- H count : Maximum number of hopes packet can travel
- TID - use by client & server.
- Flags - Unicast (0), multicast (1)
- Option : Fig. 4.39
  - * 64 byte field
  - * Composed of three field
    1. 1-byte tag
    2. 1-byte length
    3. Variable length value
  - * If tag value is 53, one of 8 type of menage



1. DHCP DISCOVER
2. " OFFER
3. " REQUEST
4. " DECLINE
5. DHCP ACK
6. " NACK
7. " RELEASE
8. " INFORM

DHCP operation : Fig. 4.40



- Two well known port number are used 67 & 68.
- DHCP discover:
  * joining host send DHCP discover message to DHCP server requesting an ip address.
  * Port no. of client is 68 and server is 67
  * Source ip is 0.0.0.0
  * Destinat ip is 255.255.255.255 (broadcast)

2. DHCP OFFER

* Server sent DHCP offer menage with an ip address.
* menage in also broadcasted.

3. DHCP REQUEST

* After recieving the ip, client request the ip so that it can use it.

4. DHCP ACK

* Server sent ACK that this ip can be used by the client.

- Two well known port numbers (67 & 68) are used because response to client is broadcast. If two client use same port number there will be problem.

Network Address Translation: (NAT)

- NAT provide mapping between private and universal address.

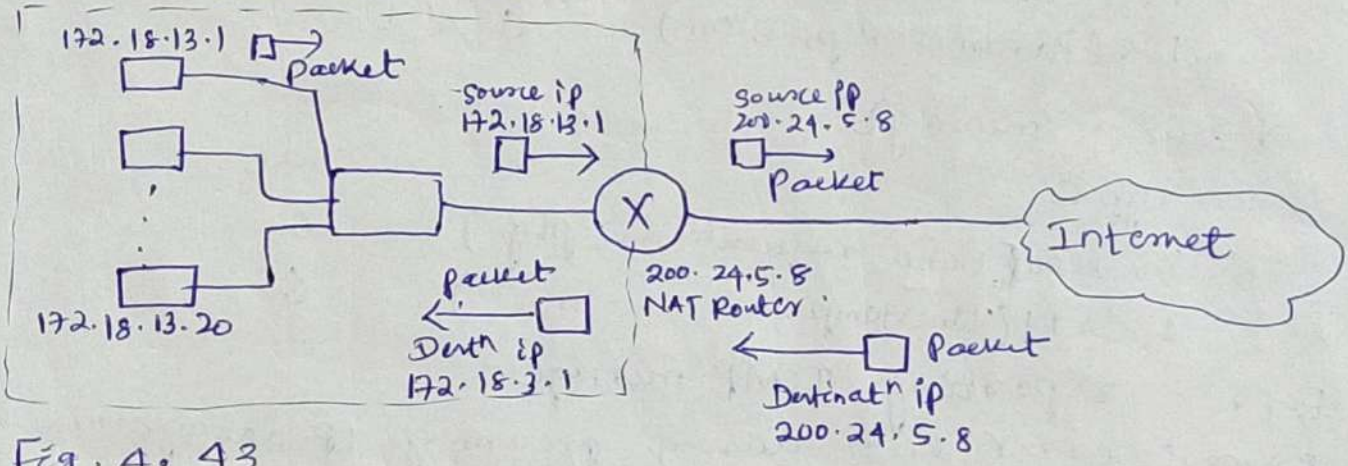


Fig. 4.43

All the outgoing packet replace the source address with NAT router address.

All incoming packet destinatn address with packet address.

Translation Table is used for this.

Private	Universal
172.18.13.1	25.8.2.18
.	.
.	.
.	.

* Multiple global address can be used
* Many-to-many relationship - port number used

Refer fig - 4.44

Table. 4:1.

Forwarding of IP packets:

- Forwarding means to place the packet in its route to its destination.
- Forwarding also means to deliver the packet to next hop.
- Forwarding is based on destination address of IP datagram.
- " " " " Label if virtual circuit is used.

(1) Forwarding based on destination address.

- Its a traditional approach.
- Forwarding table is required which gives the next hop to deliver the packet to.
- Table searched based on network address or first address of block.
- But packet contains destination address.
- To find n/w address from destinat^n address a mask is required. (/n).
- In classless addressing, forwarding table includes, mask, n/w address, interface number and IP address of next router.
- IP address used to find link layer address of next hop.
- Fig. 4.45 (slide no.
- After calculating n/w address, forwarding table is searched
- If it matches, corresponding informat^n are extracted,
- The last row of forwarding table has a default value in first column which indicate all destination address that did not match with other.
- Fig. 4.46 and Table 4.2
- Another approach used is to match the prefix bits.
- Table 4.3.
- Table shows first row has longest prefix and fourth row has shortest prefix.

- Longer prefix means smaller range of addresses and short prefix means long range of address.
- Longest Mask matching :- Forwarding table is sorted from longest mask to shortest mask.
- Hierarchical Routing - Fig. 4.49.
- Forwarding table search algorithm — Longest prefix match
- It takes time.

(2) forwarding based on Label.
- Packet forwarded based on labels.
- Fig. 4.51.
- An index is used for Label.

ICMP

- Internet Control Message Protocol
- Two types –
  (i) error reporting menage
  (ii) Query menage.
- Message formal :
  * It has 8 byte header
  * First field of header is Type. which debine the type of message.
  * Second field is code specifies the reason for menage
  * Third " checksum.
  * Fig. 4.54 (Slide no.)
  * Error reporting message

Type	Code
03 (Destinat^n unreachable)	0 to 15
04 (Source quench)	0
05 (Redirection)	0 to 3
11 (Time exceeded)	0 & 1
12 (Parameter problem)	0 & 1

  * Query menage

Type	Code
08 & 00 (Echo request & reply)	0
13 & 14 (Timestamp " ")	0

- Error reporting ICMP message -
  * Report error occur during proceing IP datagram.
  (i) Destinat^n Unreachable – Datagram did not reach it's destination.
  (ii) Source Quench – Inform source that n/w encounter cangestion and datagram droped.
  (iii) Redirection – When source use wrong router to send menage.
  (iv) Time exceeded – When all the fragment not arrive within time to the destinat^n.
  (v) Parameter problem – Problem in header.

- Query Message:
  * Use to probe or test the liveliness of host or router.
  * Two types.
    (i) Echo request & reply:
    Host or router send request message to check if it is alive or not. Host or router then reply.
    (ii) Timestamp request & reply:
    Use to find the round trip time between devices.

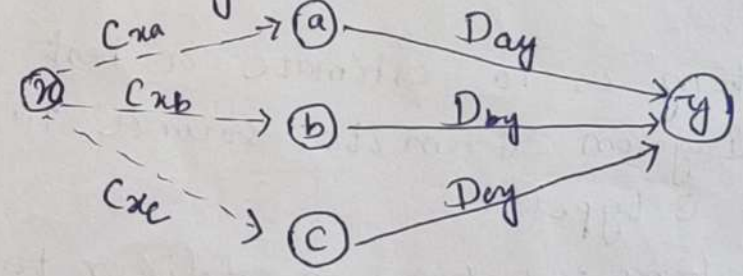
Routing:

- Objective of routing is to estimate a best route for delivery of datagram from its source to destination.
- Routing are of 3 types
  * Unicast Routing : Datagram deliver to only one destinat^n
  * Multicast " : Datagram deliver to several destinat^n
  * Broadcast " : Datagram deliver to all host.

- Routing Algorithms:
* It determines least cost path and find the ~~the~~ least cost tree for each node.
* Three routing algorithms.
  (i) Distance vector Routing
  (ii) Link state "
  (iii) Path vector "
* The internet can be considered as a graph with nodes.
* Routing algorithm help in finding best route in the graph.

Distance Vector Routing:

- Distance vector (DV) routing goal is to find best route.
- First each node create its own DV with information it have ~~with~~ from neighbours.
- Then it exchange the DV with its neighbour and update.
- The update of DV is done with the help of Bellman-ford equation.
- Suppose source node is x and destination is y. There are many intermediate nodes bet^n x & y.

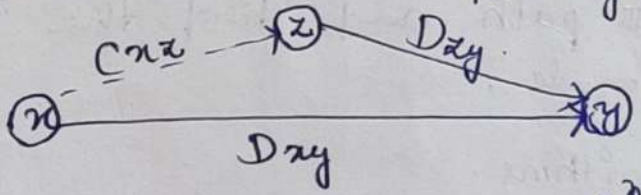


C_{xa}, C_{xb}, C_{xc} = Cost bet^n node x & a, b, C

D_{ay}, D_{by}, D_{cy} = Shortest distance bet^n a, b, C & y

$$D_{xy} = \min \{ (C_{xa} + D_{ay}), (C_{xb} + D_{by}), (C_{xc} + D_{cy}) \}$$

- In DV routing, existing least cost is updated with a least cost through intermediary node, z



So the updation equat^n become

$$D_{xy} = \min \{ D_{xy}, (C_{xz} + D_{zy}) \}$$

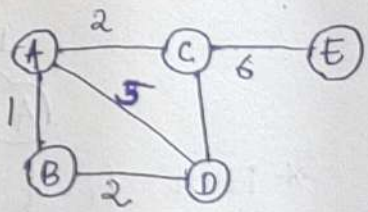
- D_{xy} ↓ new DV
- D_{xy} ↓ previous DV to any node
- $(C_{xz} + D_{zy})$ ↓ New DV after obtaining DV from neighbour using Bellman-for

- Distance vector (DV)

* A one dimentional array to represent the tree.

* Creation of Distance vector :

- Nodes are A, B, C, D, E
- Cost of each node to adjacent node are mentionted in graph.



- node A to B, cost 1.
- " " " C, " 2
- " " " D, " 5
- " " " E, no direct root, so cost ∞
- node A to A, cost 0

- So the DV of node A will be represented as

	A
A	0
B	1
C	2
D	5
E	∞

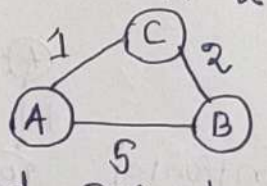
- Similarly DV of each node will be created in initial phase.
- After that in will be exchanged with the direct neighbour of a node.
- Then each node will update it's DV with Bellman-ford equat^n.

- Steps of DV routing :-

1. Initially the DV will be created with the information available
2. The DV will be exchanged with direct neighbour.
3. The DV of each node will be updated.
4. Then again it will be exchanged with neighbour.
5. This will continue untill a stable DV obtained.

- Example of DV Routing:

* Suppose there is a graph with 3 nodes A, B, C



* First find DV of each node

DV of A

A	
A	0
B	5
C	1

DV of B

B	
A	5
B	0
C	2

DV of C

C	
A	1
B	2
C	0

* Exchange the DV with direct neighbor and update the DV of each node

  - So node A will recieve the DV of B and C

* Update DV of A with B first with following eqn

$$A[\,] = \min\{A[\,], 5 + B[\,]\}$$

A	
A	0
B	5
C	1

new DV of A

$\min(0, 5+5) = 0$
$\min(5, 5+0) = 5$
$\min(1, 5+2) = 1$

old A

A	
A	0
B	5
C	1

old B

B	
A	5
B	0
C	2

* Update DV of A with C

$$A[\,] = \min\{A[\,], 1 + C[\,]\}$$

new DV of A

A	
A	0
B	3
C	1

$\min(0, 1+1) = 0$
$\min(5, 1+2) = 3$
$\min(1, 1+0) = 1$

OLD A

A	
A	0
B	5
C	1

old C

C	
A	1
B	2
C	0

* So final DV of A is

A	
A	0
B	3
C	1

- Update the DV of B with the DV obtained from A & C

  * Update with A using eqn

$$B[\,] = \min\{B[\,], 5 + A[\,]\}$$

new B

	B
A	5
B	0
C	2

$\min(5, 5+0) = 5$
$\min(0, 5+5) = 0$
$\min(2, 5+1) = 2$

old B

	B
A	5
B	0
C	2

old A

	A
A	0
B	5
C	1

  * Update with C using eqn

$$B[\,] = \min\{B[\,], 2 + C[\,]\}$$

new B

	B
A	3
B	0
C	2

$\min(5, 2+1) = 3$
$\min(0, 2+2) = 0$
$\min(2, 2+0) = 2$

old B

	B
A	5
B	0
C	2

old C

	C
A	1
B	2
C	0

  * Final DV for B in this phase is

	B
A	3
B	0
C	2

- Update the DV of C with A & B.

  * Update C with A using eqn

$$C[\,] = \min(C[\,], 1 + A[\,])$$

new C

	C
A	1
B	2
C	0

$\min(1, 1+0) = 1$
$\min(2, 1+5) = 2$
$\min(0, 1+1) = 0$

old C

	C
A	1
B	2
C	0

old A

	A
A	0
B	5
C	1

* Update C with B using

$C[\,] = \min(C[\,], 2 + B[\,])$

new C

	C
A	1
B	2
C	0

$\min(1, 2+5) = 1$

$\min(2, 2+0) = 2$

$\min(0, 2+2) = 0$

old C

A	1
B	2
C	0

old B

	B
A	5
B	0
C	2

final DV of C for this phase is

	C
A	1
B	2
C	0

* Exchange the updated DV of each node to directly connected neighbours.

	A
A	0
B	3
C	1

	B
A	3
B	0
C	2

	C
A	1
B	2
C	0

* Update the DV of each node as previous.
* In this second updation phase the DV of each node will not chang. It will be same as previous.

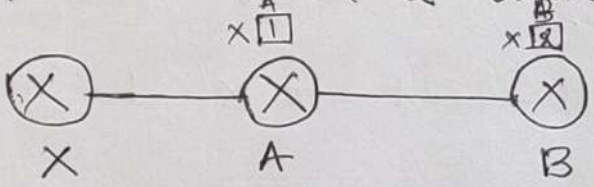
→ All A, B, C

* So the stabilized DV for each node is this.
* For n number of nodes, $(n-1)$ number of updatn is required. For stabilized DV.
* As in this example 3 nodes, 2 updation it take to get stabilised DV.

- Algorithm steps of DV routing (Table 4.9, page 305)
- Drawback of DV routing is count to infinity problem.

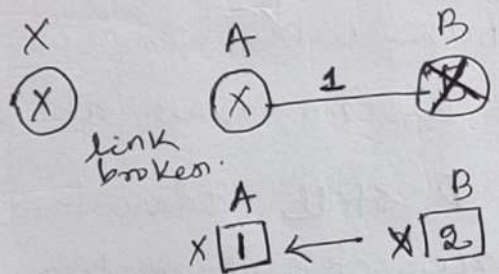
Count to infinity Problem :-

- Suppose there is a link and nodes.

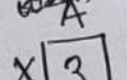


At beginning both A and B know how to reach X.

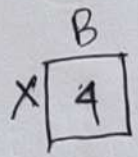
- But the link between X and A is broken at some point
- At this point, if node A update its DV, then no problem.
- But if node B sends its DV to A first, then A will thought B has informat° how to reach X. And B will thought A has informat° about X.



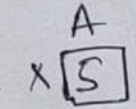
* After sometime when B's DV arrive at A



* After A's DV arrive at B



* After B's DV arrive at A



And it will increase and reach to ∞.

- This is called as count to ∞ problem

- Two solutions are provided for count to infinity problem
  a. Split Horizon
  b. Poison Reverse.

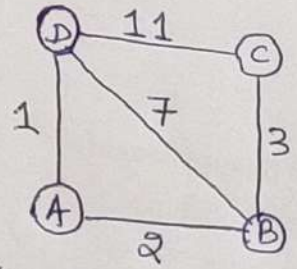
a. Split Horizon.

- Instead of sending whole DV to neighbours, only part of the DV is send.
- So A will not advertise the cost to X to node B.
- As B knows the information of X come from A, it will not send the cost of X to A.
- So A will know that there is no route to reach to X through B.
- So A will update its DV. the route to X is ∞.

b. Poison Reverse

- But the problem with split horizon is that when B will not send any information of X to A, A will be confuse wheather it is due to split horizon or B has no news about X recently.
- So in poison reverse, B still advertise the value of X, but if the source of information is A, it replace the distance with ∞.
- Meaning B says what it knows about X is come from A.

Q.



If this n/w uses DV algorithms, what is the DV of each node after stabilization?

Sol^{n} -

Step 1

Initialization of each node

A	
A	0
B	2
C	∞
D	1

B	
A	2
B	0
C	3
D	7

C	
A	∞
B	3
C	0
D	11

D	
A	1
B	7
C	11
D	0

Step 2

Exchange the DV with each node
Update the DV of each node.

(a) Updat^{n} of DV of node A → It will get DV of D and B

* with DV of B

A	
A	0
B	2
C	5
D	1

new[A] = Min{[A], 2 + [B]}

→ min(0, 2+2)
→ min(2, 2+0)
→ min(∞, 2+3)
→ min(1, 2+7)

* with DV of D

A	
A	0
B	2
C	5
D	1

new[A] = min{[A], 1 + [D]}

→ min(0, 1+1)
→ min(2, 1+7)
→ min(5, 1+11)
→ min(1, 1+0)

old A

A	
A	0
B	2
C	5
D	1

Final DV of A after updat^{n} is A

A	
A	0
B	2
C	5
D	1

(b) Updatn of DV at node B

B has DV of A, C, D

So it will do updatn with each node.

Final DV of node B after updatn is

B	
A	2
B	0
C	3
D	3

(c) Updatn of DV at node C

C has DV of B and D.

So it will update with DV of B & D

Final DV of node C after updatn is

C	
A	5
B	3
C	0
D	10

(d) Updatn of DV at node D

D has DV of A, B and C

So it will update with DV of A, B & C.

Final DV of node D after updatn is

D	
A	1
B	3
C	10
D	0

Step 3

Each node will exchange DV obtained at ~~step~~ 2 with directly connected neighbor.

Each node will update the DV

(a) Updatn of DV of node A

A has DV of B & D. So it will update.

Final DV of node A after updation

A	
A	0
B	2
C	5
D	1

(b) Updatn of DV of node B

B has DV of A, C, D. So it will update

Final DV of node B after updatn

B	
A	2
B	0
C	3
D	3

(c) Updat^n of DV at node C

C has DV of B & D. So it will update.

Final DV of node C after updation

C

A	5
B	3
C	0
D	6

(d) Updation of DV at node D

D has DV of node A, B & C. So it will be update.

Final DV of node D after updation

D

A	1
B	3
C	6
D	0

- As it is a 4 node graph, the DV will be stable after step 3.
- Stabilized DV of all the nodes are

A

A	0
B	3
C	5
D	1

B

A	3
B	0
C	3
D	3

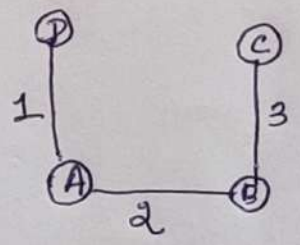
C

A	5
B	3
C	0
D	6

D

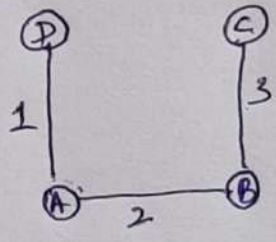
A	1
B	3
C	6
D	0

- Least cost tree from node A is

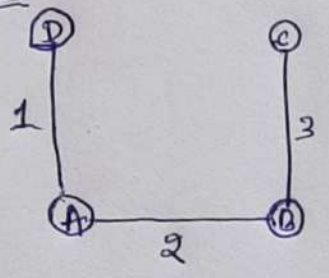


Least cost tree from node B, C and D

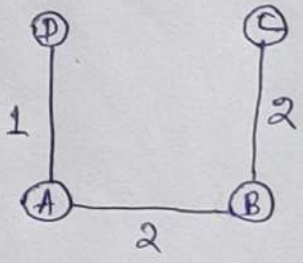
B



C

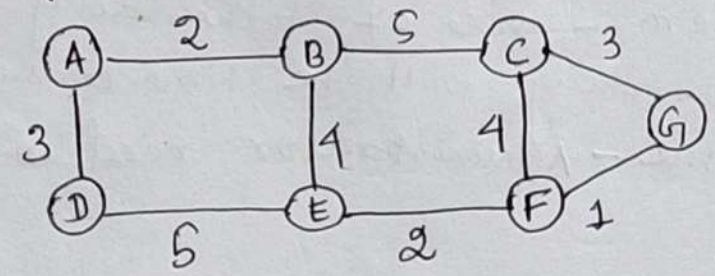


D



Link State Routing:

- Link state: Characteristic of a link (an edge) that represent the internet
- Cost associated with an edge define the state of the link.
- link with lower cost are preferred to link with higher cost.
- If cost is infinity, link does not exist.
- Collection of states for all links are called link state database. (LSDB)
- It can be represent with 2-D matrix (fig. 4.63)
- Each node sends information to immediate neighbors. using flooding.
- Two pieces of information is collected
  (i) identity of node (previous node)
  (ii) cost of ~~node~~ link
- Combination of these two information is called LS packet. (Fig. 4.64)
- Least cost trees are formed using Dijkstra's algo.
- Dijkstra's algo Table 4.5 (Page 309)
- Algorithm
  1. Node chose itself as root. Create total cost of each node based on LSDB.
  2. Root node ~~node~~ choses one node which is close and add to tree.
  3. Node repeat step 2 until all nodes are added.
- Example:



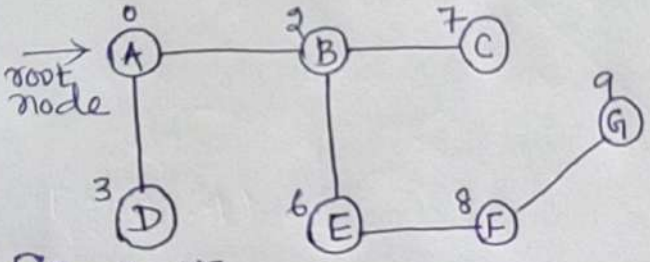
* Suppose root node is A
* Create a table with cost & previous node of each node.

(37)

D → Distance to node or cost to node
P → Previous node.

Step	Tree	D(B), P(B)	D(C), P(C)	D(D), P(D)	D(E), D(E)	D(F), D(F)	D(G), P(G)
1.	A	(2, A)	—	3, A	—	—	—
2.	AB		7, B	(3, A)	6, B	—	—
3.	ABD		7, B		(6, B)	—	—
4.	ABDE		(7, B)			8, E	—
5.	ABDEC					(8, E)	10, C
6.	ABDECF						(9, F)
7.	ABDECFG						

- Least cost tree



- Fig. 4.65

Path Vector Routing:

- Least cost tree is not important here.
- Here which path the packet is travelling is important.
- Source can control the path.
- It is designed to route a packet between ISPs.

Routing protocols

- Routing in internet cannot be done using single protocol because.
  (i) Scalability Problem - Size of forwarding table will be huge. Searching will be time consuming.
  (ii) Administrative issue - Administrator need control in its system.
- Hierarchical Routing:
  * It means considering each ISP as autonomous system (AS).

* Each AS run a routing protocol.
* Global internet run global protocol to glue all AS together.

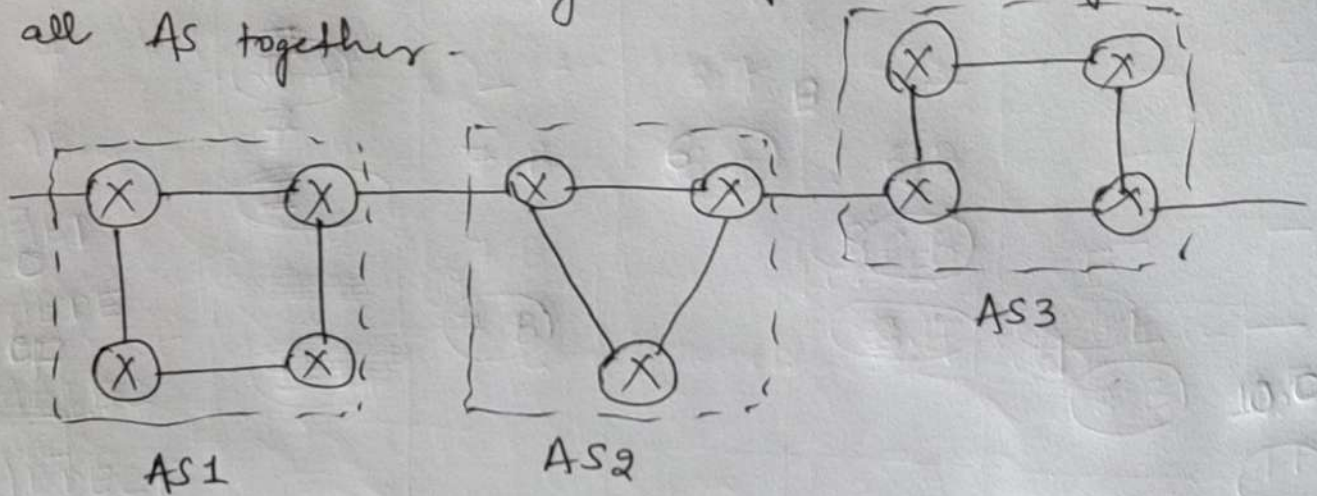


Fig. 4.79.

* Routing protocol run in each AS is called intra-AS routing protocol or intradomain routing protocol or interior gateway protocol (IGP)
* Global routing protocol is called inter-AS routing protocol or interdomain routing protocol or exterior gateway protocol (EGP)

- Three routing protocols are there
  a. Routing Information Protocol (RIP)
  b. Open source shortest Path first Protocol (OSPF)
  C. Border Gateway Protocol (BGP)
- RIP baced on Distance vector routing ~~protocol~~ algo
- RIP is an intradomain routing protocol.
- OSPF is baced on link state routing ~~protocol~~ algo.
- OSPF is an intradomain routing protocol
- BGP is baced on path vector routing algo.
- BGP is a interdomain routing protocol.

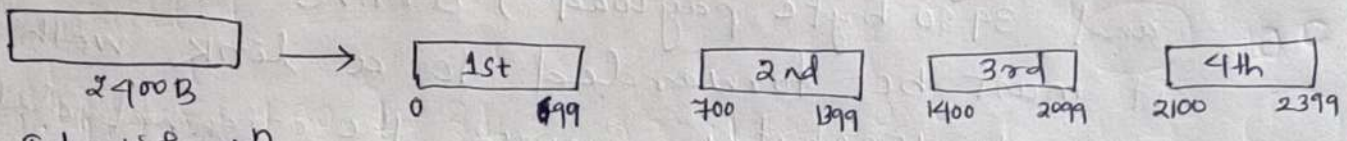
— END —

Q-1- Consider sending a 2400 Byte datagram into a link that has MTU of 700 Byte. Suppose the original datagram stamped with identification number 422. How many fragments are generated? What are the values in values in various fields in the IP datagram generated related to fragmentation?

Sol^n- Size of datagram = 2400B

MTU = 700B

Fragmented generated = $\frac{2400}{700} = 3.42 \approx 4$



Identificat^n →	422	422	422	422
M flag →	1	1	1	0
Fragmentat^n Offset →	$\frac{0}{8} = 0$	$\frac{700}{8} = 87$	$\frac{1400}{8} = 175$	$\frac{2100}{8} = 262$

Q-2 - Assume that an IP datagram can remain in the network for a maximum of 40 second before being delivered to destination. Calculate the maximum data rate achievable in Mbps at host so that no confusion will arise during the reassembly of fragment at destination. Assume the size of each datagram in the network is 1000 Bytes.

Sol^n- It is needed to be ensured that the identification field does not wrap around within 40 sec to avoid confusion.

- Identificat^n field is 16 bit.
- This means host can sent 2^{16} datagrams maximum in 40 sec before wrap around occurs.
- One datagram = 1000 B = 1000×8 bits
- So in 40 sec data that can be send is $2^{16} \times 1000 \times 8$ bits

" 1 sec " " " " " " = $\frac{2^{16} \times 1000 \times 8}{40}$ bits

- So data rate = $\frac{2^{16} \times 1000 \times 8}{40}$ bits per sec

$= 13107 \times 10^3$ bps

$= 13.107 \times 10^6$ bps

$= 13.107$ Mbps

- So maximum achievable data rate should be less than 13.1 Mbps.

Q-3 - A datagram of 4000 bytes (20byte IP header and 3980 byte payload) arrive at a router. It must be forwarded to a link with an MTU of 1500bytes. Suppose the original datagram stamped with identificatn number of 777.

i. How many fragments will be there?

ii. How many bytes of data will be there for each fragment

iii. Find the value of identification, M flag and offset value of each fragment.

Sol^n - Datagram = 4000B (20B header + 3980B data)

MTU = 1500B (20B header + 1480B data)

(i) Number of fragment = $\frac{4000}{1500} = 2.66 \approx 3$

(ii) First fragment data byte = 1500 - 20 = 1480 byte

Second " " " " = "

Third " " " = 3980 - (1480 + 1480)

= 1020 Byte.

(iii)

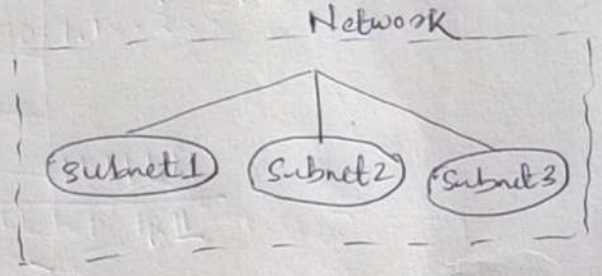
Fragment	Data	Identificatn	M flag	Offset
1	1480	777	1	$\frac{0}{8} = 0$
2	1480	777	1	$\frac{1480}{8} = 185$
3	1020	777	0	$\frac{2960}{8} = 370$

Q-4- Suppose you are working as a n/w engineer in an IT firm. You are asked to create 3 subnets over a given IP 130.1.2.3/255.255.0.0 A subnet supporting 254 hosts, a subnet supporting 1023 hosts and a subnet supporting 2048 hosts. Find the network id, broadcast id, IP address range and subnet mask of each subnet.

Sol^n - IP is 130.1.2.3/255.255.0.0.

3 subnets will be designed

- First subnet 2048 host
- Second " 1023 "
- Third " 254 "



(a) Subnet with 2048 hosts

- Total number of IP required 2048+2 = 2050 ~~host~~.
- $2050 \simeq 2^{12}$
- Prefix length, $n = 32 - 12 = 20$
- IP of n/w is 130. 1. 2. 3 i.e.

  10000010 00000001 00000010 00000011
- Network id is

  10000010 00000001 00000000 00000000

  (12 bit set to 0)

  130.1.0.0/20
- Broadcast id is

  10000010 00000001 00001111 11111111

  (12 bit set to 1)

  130.1.15.255/20
- Range of IP addrem. is 130.1.0.1/20 to 130.1.15.254/20

- Subnet mask is

  11111111 11111111 11110000 00000000

  255.255.240.0/20

(b) Subnet with 1023 hosts

- Total number of IP = 1023 + 2 = 1025
- $1025 \approx 2^{11}$
- Prefix length, $n = 32 - 11 = 21$
- Network id is

  10000010 00000001 00010000. 00000000

  130.1.16.0/21

  (Previous subnet id ended at 130.1.15.255. So this subnet will start with next)
- Broadcast id is

  10000010 00000001 00010111 11111111

  (11 bit to 1)

  130.1.23.255/21.
- Range of ip is 130.1.16.1/21 to 130.1.23.254/21
- Subnet Mask

  11111111 11111111 11111000 00000000

  255.255.248.0/21

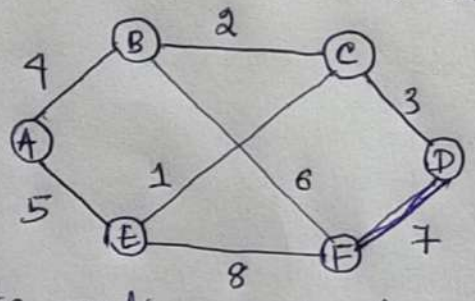
(c) Subnet with 254 host

- Total ip required = 254 + 2 = 256
- $256 \approx 2^8$
- $n = 32 - 8 = 24$
- Network id is

  10000010 00000001 00011000 00000000

  130.1.24.0/24
- Broad cast id is 130.1.24.255/24
- Range of ip 130.1.24.1/24 to 130.1.24.254/24
- Mask is 255.255.255.0

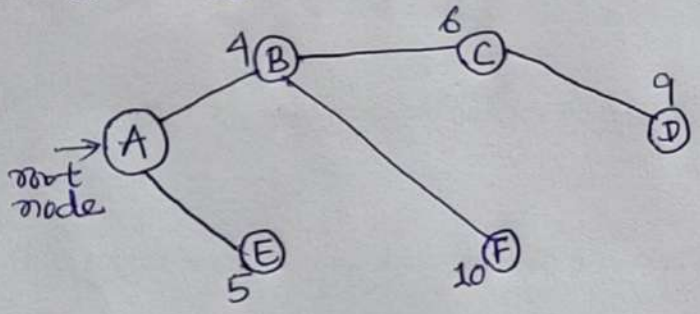
Q-5- Construct the routing table at node A using link-state routing protocol to determine shortest paths from node A to all other nodes in the network.



Sol^n: Link state routing protocol uses Dijkstra algorithm:

Steps	Tree	D(B), P(B)	D(C), P(C)	D(D), P(D)	D(E), P(E)	D(F), P(F)
1	A	(4, A)	—	—	5, A	—
2	AB		6, B	—	(5, A)	10, B
3.	ABE		(6, B)	—	•	10, B
4	ABEC			(9, C)		10, B
5.	ABECD					10, B
6.	ABECDF					

- Least cost tree is.



Q-6- Longest prefix matching numerical

Classless inter domain routing (CIDR) recieves a Packet with addren 131.23.151.76. The routing table has following entries:

Address	Interface
131.16.0.0/12	3
131.28.0.0/14	5
131.19.0.0/16	2
131.22.0.0/15	1

In which interface the packet will be forwarded?

Sol^n - First find all the addren in binary.

Address	Interface
10000011 00010000 00000000 00000000/12	3
10000011 00011100 00000000 00000000/14	5
10000011 00010011 00000000 00000000/16	2
10000011 00010110 00000000 00000000/15	1

- Binary of addren 131.23.151.76 is.

10000011 00010111 10010111 01001100

This prefix is matching with every address in table

- Then we have to find longest prefix which match.

10000011 00010111 10010111 01001100

This prefix is matching with interface 3, 2, 1

- Find the longest that this prefix

10000011 00010111 10010111 01001100

This prefix match with interface 1.