IPV6 addresses

diplication of IPV6 version.

length : 128

no. of addresses : 2^{128}

Translation : IPV4 to IPV6 converts.

for representation : each bit separated by colon.

FDEC : 0074 : 0000 : 0000 : 0000 : B0FF : 0000 : FFF

↓

16 bits of length section.

2 bytes

=> Zero compression is used only once.

=> source & destination addresses are 16 bytes.

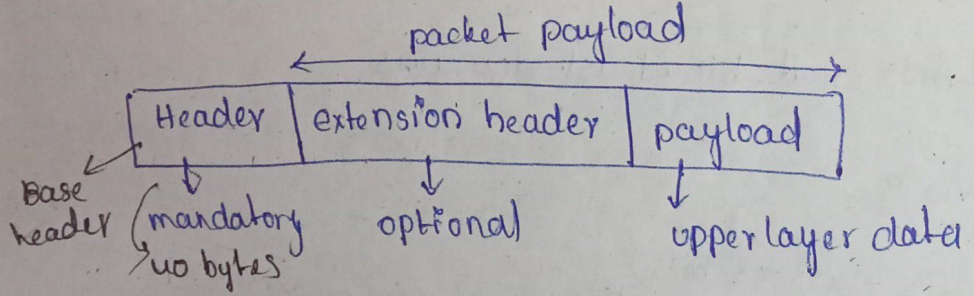
Total = 16 + 16 = 32 bytes.

version : is a number representing with 6.

packet = transport data + header

version length : 4 bits ;

value : 6



Traffic class : based on priority it will check and control the nlw.

=> Lowest priority packets - discord first

highest " " "

=> congestion is ntg but traffic class

congestion control traffic (0 – 7) (lowest – highest priority)

non " " " (8–15)

congestion control traffic table:

priority	meaning
congestion control traffic: 0	• no specific traffic. (no priority is assigned) process doesn't assign any priority.
1	• background data ~~priority~~ Ex: delivery of news.
2	• un attended data traffic Ex: email.
3	• Reserved
4	• attended bulk data traffic Ex: FTP & HTTP Here use is waiting for response.
5	• Reserved
6	• interactive traffic for user interaction user require tell net
7	• control traffic. Ex: OSPF, RIP, SNMP OSPF - open shortest path first RIP - Routing in protocol. → both are ~~routing protocols~~. unicasts. SNMP - simple nlw mgmt protocol.
non congestion control traffic: 8	• non congestional control traffic 1. minimum delay is expected. 2. discording of packets not desirable 3. retransmission is impossible in mann

8 ⋮ 15	Ex: realtime audio & video. ⇒ given data with greatest redundancy ⇒ data with least redundancy

⇒ traffic class length : 8 bits
flow label length : 20 bits

⇒ Flow Label can be used to speed up the processing of a packet by a router by referring the flow label table.

payload: field length is 16 bits.
⇒ the packet contain how much bits/bytes info are there it is specified.

next header: length is 8 bits.
⇒ used to indicate which type extension header if it has.

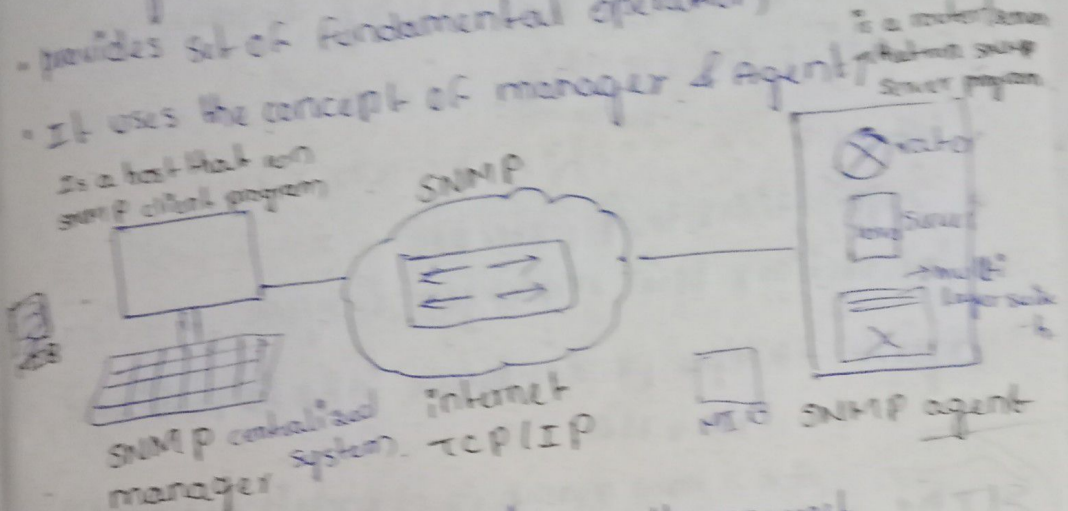
Hop (→ nodes) limit: length is 8 bits
⇒ This field is used to stop packet to loop (problem) in the n/w infinitely. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop).
⇒ when the field reaches 0 the packet is discarded.
⇒ every link is moved the Hop limit is decreased ~~to remove it we use it~~ packets are dropdown.
⇒ In looping HOP limit is varying

Source address: 128 bits
this field indicates the address of originator of the packet

Destination address: 128 bits
This field provides the address of intended recipient of the packet.

SNMP (Simple nlw mgmt protocol)

- Framework for managing devices on an internet
- manages & monitors devices on LAN & WAN
- provides set of fundamental operation
- It uses the concept of manager & Agent



SNMP ≠ controls the agent. send the request
manager

SNMP agent : collection of diff. components
(router, server, multi layer switch)

⇒ SNMP manager sends the request to check the SNMP agent is there any kind of congestion or any kind of traffic or not.
is there any kind of fault.

Components of SNMP:

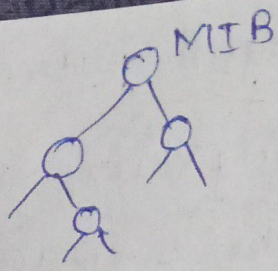
① SNMP manager: also called nlw mgmt System (NMS)

② SNMP Agent: nlw mgmt / nlw module that is actually installed on manage device (PC, router, switches servers)

③ MIB:
⇒ so track it easily.

management Information base : required to track about the resources.

⇒ It contain all the info that are required or used by the resources that are used for managing it



⇒ MIB always used hierarchy ~~naement~~ for organize information

⇒ ~~interact~~ use TCP/IP (Transmission control protocol) protocol rule

- application level protocol
- interaction blw manager & agent

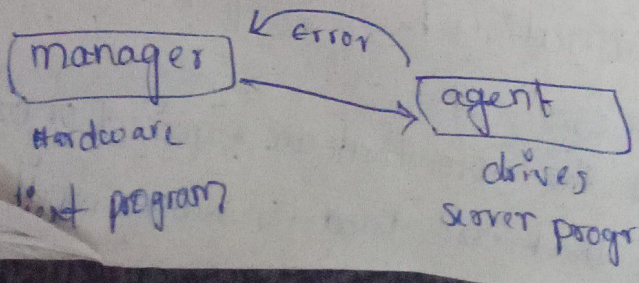
agent - keeps performance info in particular database.

Ex: how many no.of packets router can store & how many no.of packets being send. compare this values and check any kind of congestion or traffic or not

Mgmt of SNMP based on 3 basics ideas.

* manager —request→ agent
* forces agent to perform task by resetting the value in a particular db.
* warning about unusual situation
  Ex: ~~mutual sil~~

⇒ SNMP used for smooth traffic or flow of packet data can move easily.
used for managing internet/nlw.



→ manager check (all status) whether all components are working

⇒ MIB a text file to store all transactions (what are down) in a mib file.

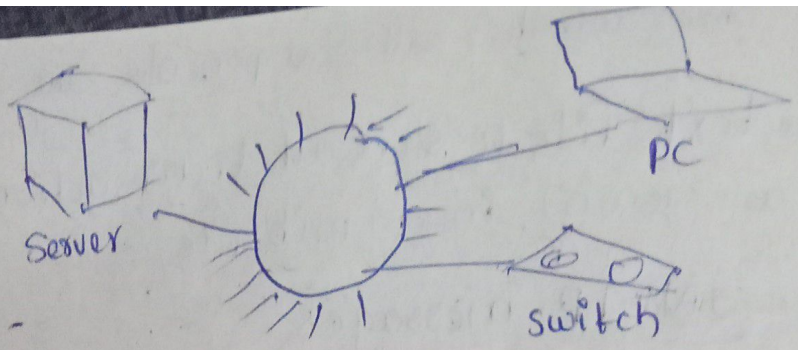
SNMP messages

- GetRequest - manager send this msg to request data from agent.
- GetNext Request - send to discuss wt data is available at SNMP agent.
- GetBulkRequest - retrieve large data from agent. used in version - 2c
- SetRequest - set the value of an object on agent
- Response - sent from agent upon a request from the manager.
- Trap - sent by agent without being requested by the manager. It is sent when fault is occurred in a nlw.
- InformRequest - used to identify if the trap msg is recieved by the manager or not. used in v-2c.

⇒ SNMP : control nlw devices

⇒ SNMP operates over UDP (connectionless user datagram protocol) and typically use port numbers 161 for general requests and 162 for recieving notifications or traps.

⇒ why SNMP is important means enables the monitoring of multiple devices from a single location & simplifying the mgmt of large nlws.



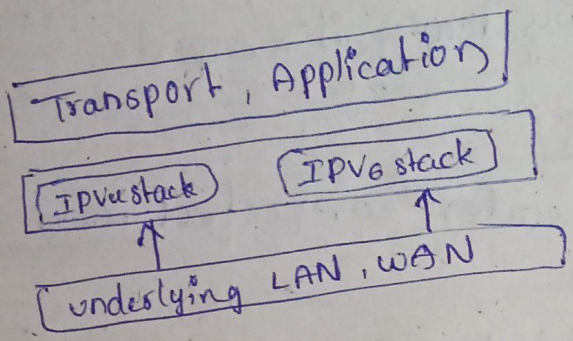
=> SNMP is application layer protocol (OSI layer 7)

Transition from IPv4 to IPv6

→ we are use 3 techniques for conversions.

1. Dual stack — 2 stacks maintain.
2. Tunneling
3. Header translation

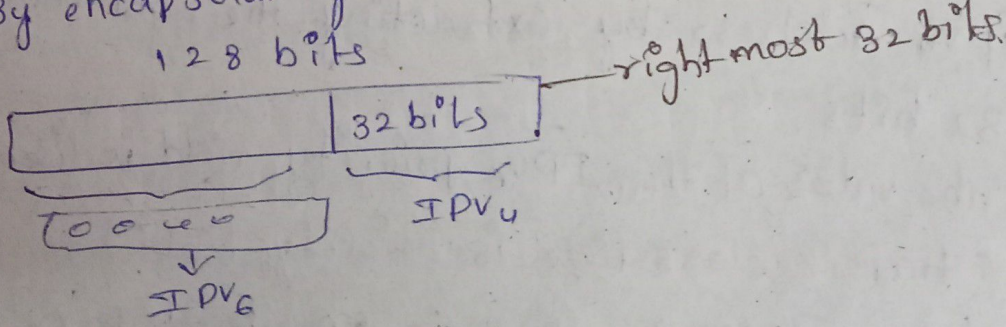
① IPv4 - stack & IPv6 stack are 2 stacks in dual stack.



⇒ packet is identified based on ~~header~~ version field in a header.
(either IPv4 header / IPv6 header version)

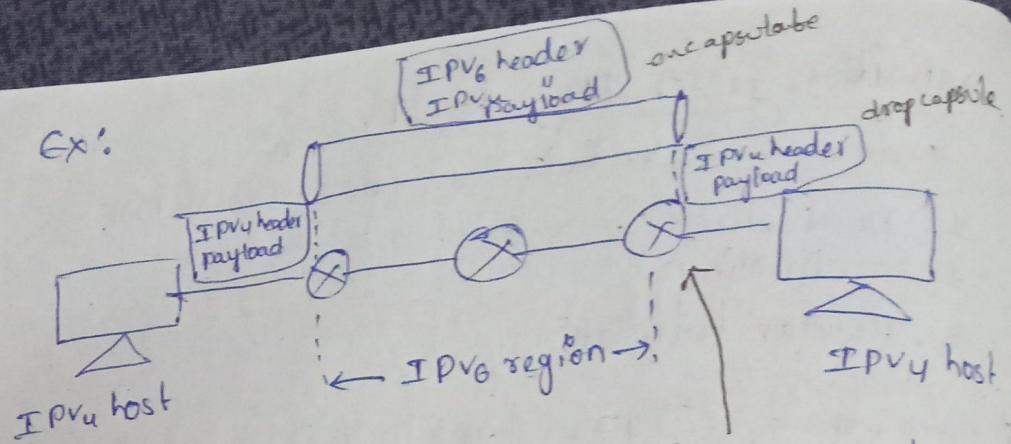
② Tunneling:

⇒ By encapsulating the bits.



⇒ Tunelling can be done by encapsulating IPv6 packet in an IPv4 packet when it enters the ~~when it enters~~ region (IPv6) and leaves its capsules when it enter into the region IPv4.

Ex:



⇒ router worked as translator to convert IPV6 to IPV4 by leaves it's capsule.

③ Header translation:

⇒ completely IPV6 header converted into IPV4 header.

⇒ used to convert header of IPV6 to IPV4.

① find out header fields & compare both. (IPV4 & IPV6)

rules / procedure:

1. the ipv6 mapped address is changed to an ipv4 address by extracting the right most 32 bits.
2. the value of the IPV6 priority field is discarded. (traffic class field is priority field)
3. the type of service field in IPV4 is set to 0.
4. the checksum for IPV4 is calculated and inserted in the corresponding field & the ipv6 flow label field is ignored.
6. comparatable extension headers are converted to options & inserted in the ipv4 header some

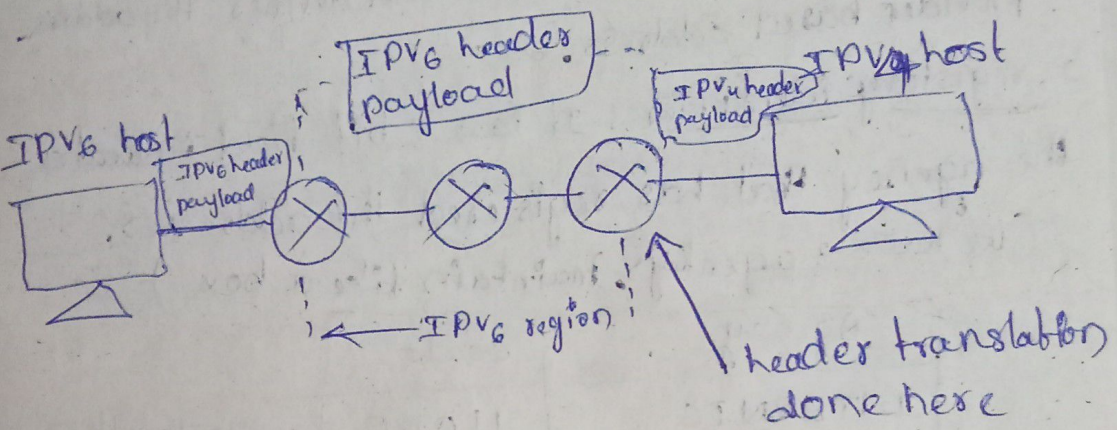
(6 types (fragmentation, authentication, encapsulation))

may have to be dropped.

7. The length of IPv4 header is calculated and inserted into the corresponding field.

8. The total length of IPv4 packet is calculated and inserted in the corresponding field.

IPV6 → IPv4



Types of IPV6 addresses: 4 categories.

unicast, multicast, reserved, mapped, local addresses.

Unicast address: the packet is delivered to only single node.

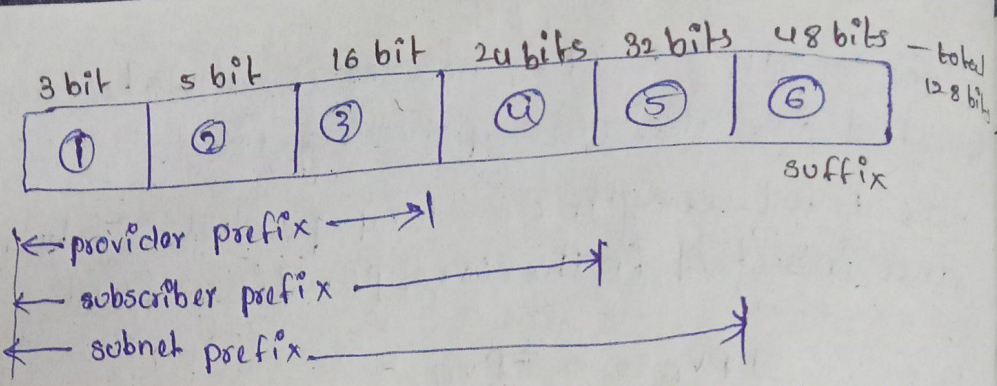
IPV6 defines two types in unicast

⇒ unicast addresses defined by 2 types.

1. geographically based - used in future.
2. providers based

providers based: similar to IPV4.

1. Type identifier.
2. registery "
3. provider identifier
4. subscriber "
5. subnet "
6. node "



1. type identifier : It is a 3 bit field defines the address as a provider based address.

2. registery identifier : It is a 5 bit field indicates the agency that has registered the addresses.
(we have 3 agenkys, maintain like a box)

Agency	code
INTERNIC	11000
RIPNIC	01000
APNIC	10100

← 5 bits length

INTERNIC - north american agency
RIPNIC - European agency
APNIC - Asian and pacific agency

3. provider identifier : is a 16 bit field. It defines the provider for internet access (ISP).

4. subscriber " : is a 24 bit field/length. we can subscribe the internet in an organization.

5. subnet " : is a 32 bit field. Each subscriber can have many diff subnetworks & each sub nkw can have an identifier.

6. node identifier : Is a 48 bits field. It defines identity of a node connected to a subnet

bits 48 bits — total 128 bits

	⑥

suffix

fines the address as

field indicates
addresses.
a box)

5 bits length

agency

fic agency
field. It defines the
P).
field/length -
n organization.
eld. Each subscriber
f each sub nlw can

field. It defines
subnet

2. Multicast addresses:

diff b/w multicast & broadcasting

⇒ multicast addresses are start with FF.
⇒ first 2 bits hexadecimal value of
⇒ group of nodes are ntg but multicasting addresses.
⇒ broadcast also a multicasting.
⇒ 1st right most 8 bits are checked.

multicasting:

8bits	4bits	4bits	112 bits
11111111	Flag	scope	group ID
FF			

Flag: 2 types of flags.

permanent	0000
Transient	0001

→ 4 bits

1. permanent – all zeroes 0000
2. Transient – 0001

scope:

Reserved	0000
nod local	0001
link local	0010
site local	0101
organizational	1000
Global	1110
Reserved	1111

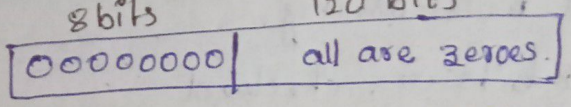
Reserved is 0000 & 1111.

3. Reserved addresses:

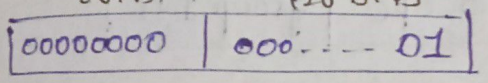
⇒ these are start with 8 zeroes.
⇒ these are sub categorized in 4 types.
1. unspecified address
2. loopback "
3. comparable "
4. mapped "

1. unspecified address: used when host doesn't known its own address and send an enquiry to find its address.

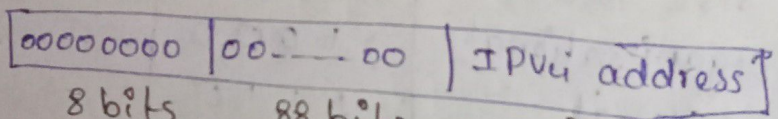
format:



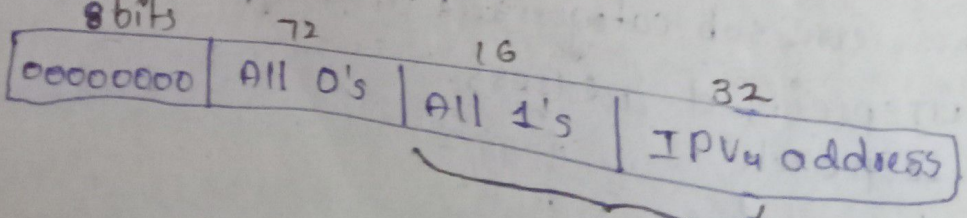
2. Loopback address: It is used by a host working or not to test itself without going into the internet.



3. compartable address: source & sender are IPV6 It is used during the transition from IPV4 to IPV6 when i.e a computer using IPV6 wants to send a message to another computer using IPV6 but the msg need to pass through a part of the nlw that still operates in IPV4.



4. mapped address: It is used when a computer that has mapped to IPV6 wants to send a packet to computer still using IPV4 it is also used during transition

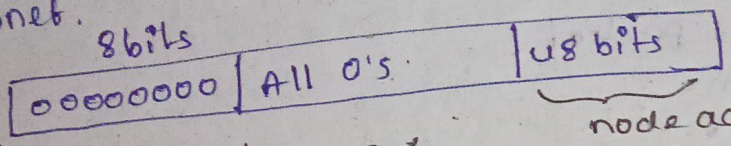


4. Local addresses:

⇒ providing addressing for private nlws.

⇒ 2 types of addresses defined

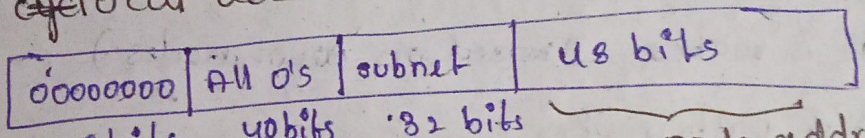
1. Link local addresses: used in isolated subnet.



2. site local addresses:

⇒ These addresses used in a isolated side with several subnet.

3. site local addresses:



Design issues in n/w layer

N/w layer:

- Addressing
- packetisation
- Routing
- inter-networking

N/w layer design issues:

1. Store & forward packet switching:
   store all bits of packet & service to node transfer
2. Services provided to transport layer:
   - UDP - connectionless in TCP.

⇒ In transport layer maintain port numbers.

connectionless: no dedicated path.

connection - establish - path - remove

→ irrespective of packet
no order, no setup.
⇒ in connection.
establish → transmission → teared on
Dedicate link physical or virtual
↓
for establish the connection -
ordering the packet..

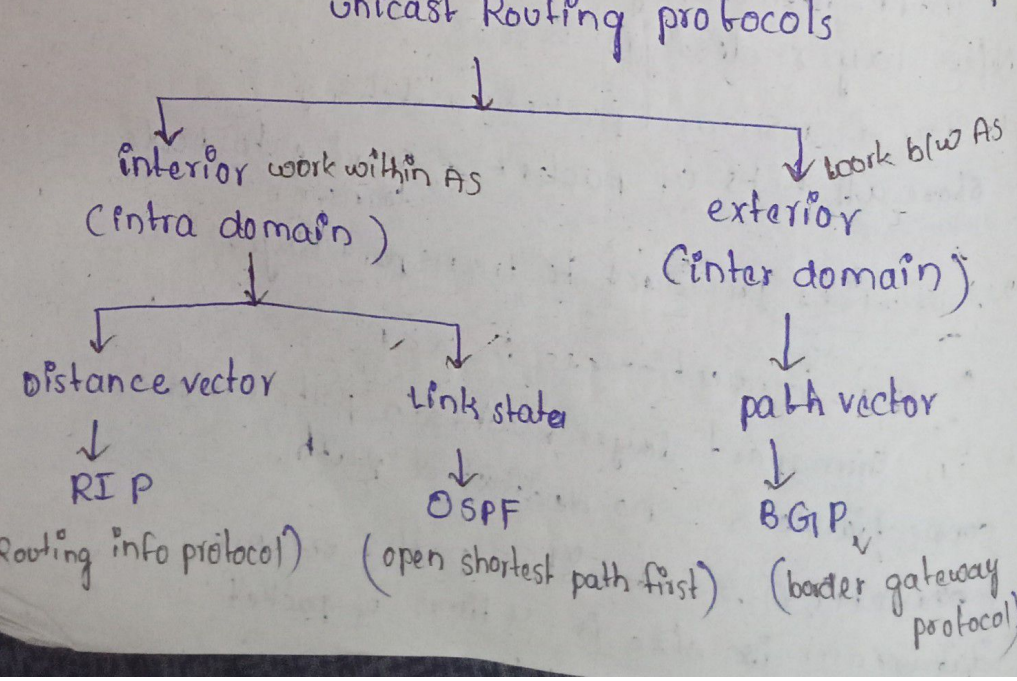
Routing Algms :
Flooding - see in systems all packets move like flooding.

① optimal principle : take shortest paths.
based on cost, along.
(a) B is router. (based on no. of nodes)

② shortest path router :
from A → D using dijkstras.

③ Flooding : If it is reached to the zero node
everytime drop 1.

Unicast Routing protcoles =



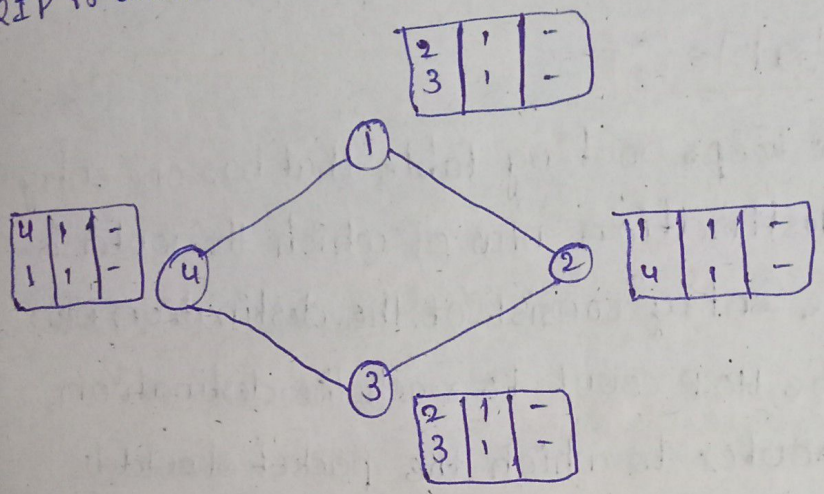
⇒ In RIP bellman-Ford algorithm is used.

⇒ In OSPF dijkstra's algorithm is used.

AS (Autonomous System) : only some nlws are connected

Distance vector :

RIP is used to find out path from src - destiny



⇒ in bellman-Ford, everytime HOP count increased by 1 & It maintain neighbouring info & take adjacent nodes.

⇒ RIP is dynamic configuration
↓
updated according to Time period.

Distance vector routing : In this each router periodically shares its knowledge about the entire internet with its neighbours.

The 3 key features for this routing is

① sharing knowledge about the entire AS : It means each router shares its knowledge about the entire AS with its neighbours whether it is important or not.

② sharing only with neighbours: each router sends its knowledge only to his neighbours through all its interfaces.

③ sharing at regular intervals: each router sends its knowledge to its neighbours at fixed intervals for Ex: every 30 sec.

Routing table :=

Every router keeps routing table that has one entry for each destination nlw of which the router is aware ⇒ the entry consist of the destination nlw address, the HOP count to reach the destination, the next router to which the packet should be delivered to reach its final destination.

Here the HOP count is the no. of nlws.

⇒ the routing table may contain other info like the subnet mask etc.

Destination	HOP_count (router)	next node
163.5.0.0	7	172.6.23.4
197.5.13.0	5	176.36.17.5
183.45.0.0	4	200.5.1.6

RIP updating table algorithms :=

⇒ the routing table is updated based on reciept of a RIP response message.

⇒ there are 3 steps ; they are ;

step①: Add one hop to hop count for each advertised destination.

step②: Repeat the following steps for each advertised destination.

If (destination not in the routing table)

Add the advertised information to the table

~~else~~

~~else~~ if (next-hop field is the same)

Replace entry in the table with advertised one.

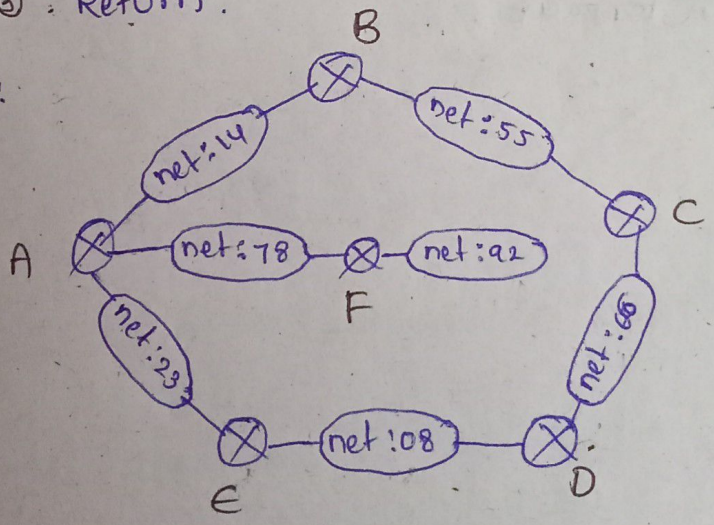
else

if (Advertised hopcount smaller than one in the table)

replace entry in the routing table.

step③: Return.

Ex:



A

14	1	-
78	1	-
23	1	-
92	2	E
08	2	E
66	3	E

B

14	2	A
78	2	A
23	2	A
55	1	-

C

14	3	B,A
78	3	B,A
23	3	B,A
55	2	B
CC	1	-

D

14	4	CBA
78	4	"
23	4	"
55	3	CB
66	2	C
08	1	-

E

14	5	DCBA
78	5	"
23	5	"
55	4	DCB
CC	3	DC
08	1	-

F

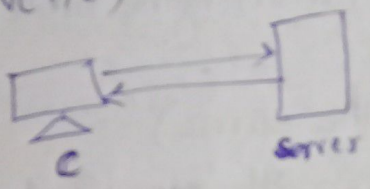
14	2	A
78	2	A
23	2	A
92	1	-

HTTP tran

2> FTP
3> SSH
4>
5> HDLC

Non persistant
V(1,0)

persistant
V(1,1)



status codes : 5 types

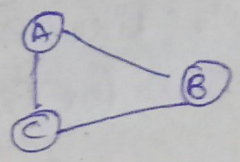
① 1 x x — 100 - continue
2 x x — 101 - switching
3 x x — 200 -
4 x x
5 x x

Header format.

Link state routing Protocol :-

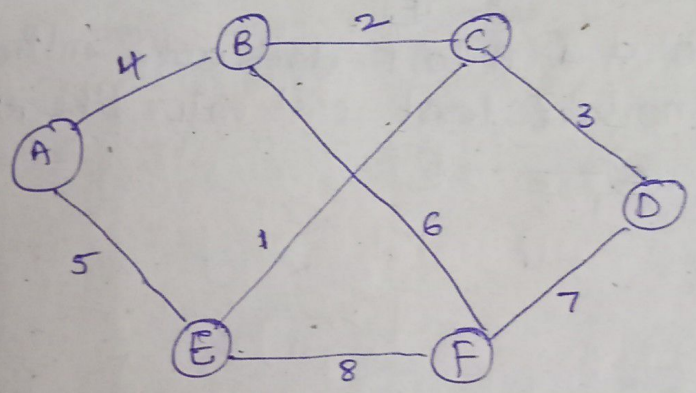
→ unicast algorithms
→ initially every router send hello msg (to know about adj node) & consist
  Routername
  sequence number
  Age

ex :



⇒
⇒ Every router maintain shortest path from one node to another node.

⇒



at router A : from that node to all other nodes.

	B	C	D	E	F
A	4	∞	∞	5	∞
AB	(4)	6	∞	5	10
ABE	(4)	6	∞	(5)	10
ABEC	(4)	(6)	9	(5)	10
ABECD	(4)	(6)	(9)	(5)	10

• take least cost

don't consider A's connection

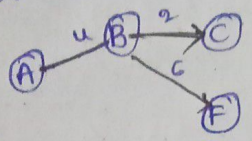
- from c which nodes are connected to it check.

⇒ upcoming cost is less than in the table cost then you can modify with upcoming cost only.

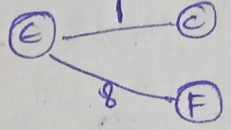
→ at A what is the cost from each node we can write there and if no cost is then keep ∞.

→ after giving all costs which cost is less take that node like AB.

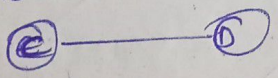
→ now for B check which nodes are connected to it



→ in AB row which cost is less take that corresponding node like ABE.



from A to C & A to F check cost & entire in ABE row. no take least cost value like ABEC.



3) pa

→ Eve

SP

kno

comp

Ex:

AS, tabl

A_1	AS1
A_2	AS2
A_3	AS3
A_4	AS4
B_1	AS1
B_2	AS2
B_3	AS3
C_1	AS3
C_2	AS3
C_3	AS3
C_4	AS3
D_1	AS4
D_2	AS4
D_3	AS4

⇒ Be

Internal

Exter

node
is then
take that
connected to it
e that
entire in
like ABEC.

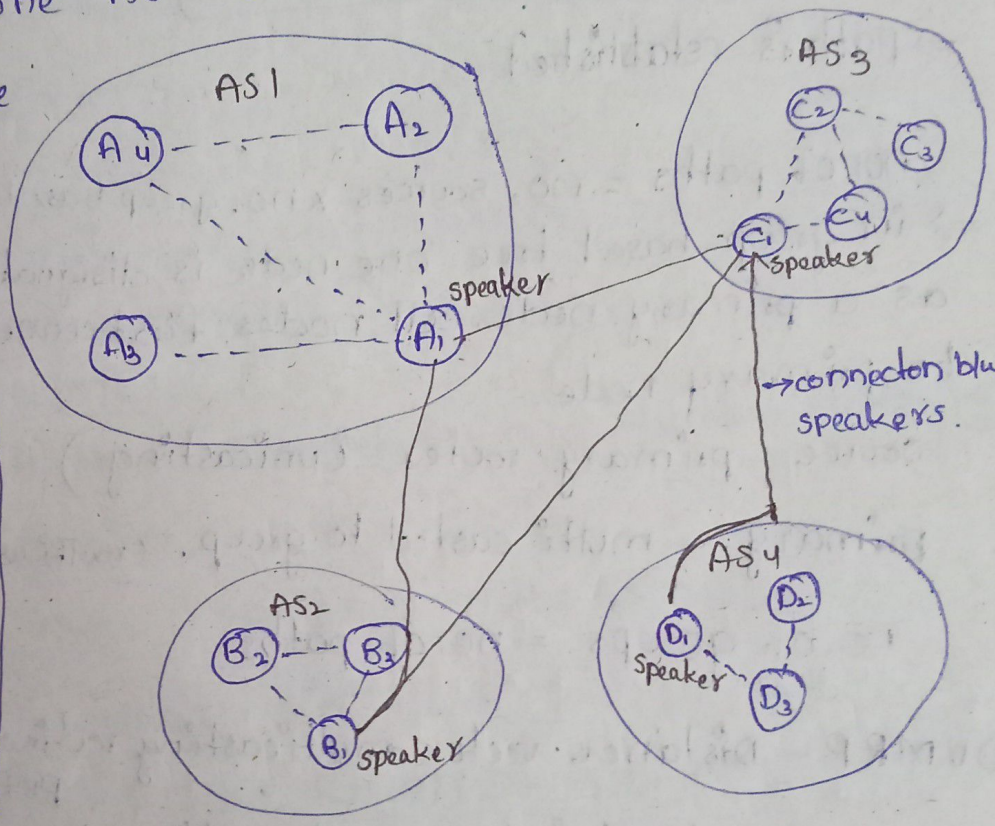
3) path vector routing protocol : BGP protocol is used to it.

→ Every AS one node is act as a speaker node. speakers in a each node can share their knowledge with adjacent speaker. This means complete info is maintain one node.

ex: one AS is ,

AS1 table

A_1	AS1
A_2	AS2
A_3	AS3
A_4	AS4
B_1	AS2
B_2	AS2
B_3	AS2
C_1	AS3
C_2	AS3
C_3	AS3
C_4	AS3
D_1	AS4
D_2	AS4
D_3	AS4



⇒ BGP session by using TCP connection.

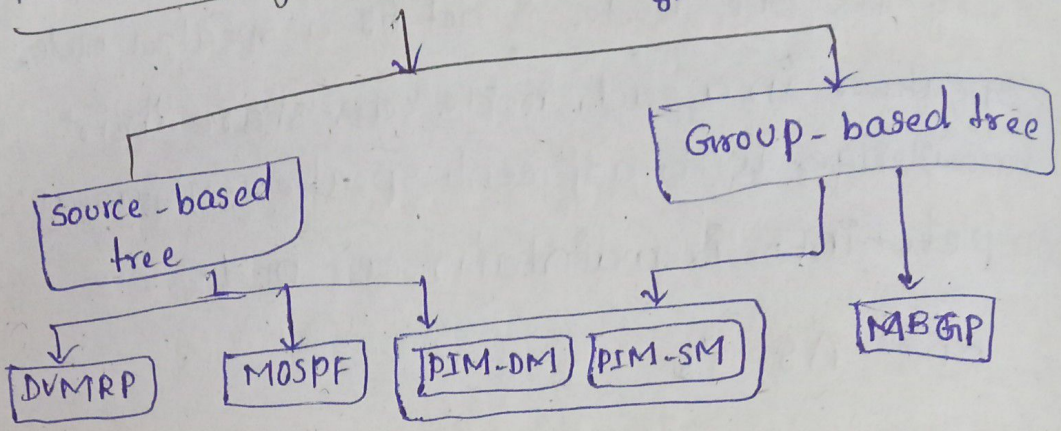
Internal ← I-BGP used within autonomous system

E-BGP " b/w " "

External

Multicasting protocols → one source node send packets to group of nodes

or



⇒ path is established

no. of paths = no. sources × no. group based

⇒ in group-based tree one node is disignated as a primary node. all nodes first connected to primary node.

source – primary router (unicasting)

primary – multi casted to group. (multicasting)

no. of groups = no. of paths.

DVMRP – Distance vector multicasting routing protocol.

PIM-DM – protocal independent multicasting dense mode

PIM-SM – " " " source mode.

MBGP – multicasting border ~~group~~ gateway protocol.

Transport layer :

→ blw application & nlw layer.
→ independent from the nlwing ~~layer~~.
→ only use port / application for transport.
→ it is called as independent from the nlwing protocol.
→ It is reliable : connection oriented.
⇒ SCTP, TCP — reliable
UDT — unreliable
⇒ we maintain port numbers. It is always 1 integer value.

Port Number :

IANA ~~Rano~~ : Internet Assigned Number Authority
port number length = 16 bits
" range range = 0 to 65,535
⇒ this range divided into 3 parts.
* 0 - 1023 : well known ports ⇒ assigned & controled by IANA
* 1024 - 49,151 : Registered ports ⇒ they can only registered by a IANA but not controlled by IANA.
* 49,152 - 65,535 : Dynamic ports ⇒ It is not registered & not controlled by IANA.

Socket Addresses (IP + port number)

Ex:-
IP adds : 200. 23.56.8 port number 69
socket add 200.23.56.8 60

protocols :

① UDP : user datagram protocol. faster

⇒ a connectionless, unreliable transport protocol.

② UDP - Data gram : header length 8 bytes.

→ 4 fields in header format.

Every field has 2 bytes of info

Ex : CB84000D001C001C

Source

0000
8421
16 1101

a>

b> 28

c> user datagram length = 28 - 8 = 20

total length = header length + datagram

② TCP : — ~~UDP~~

connection oriented. SCTP

⇒ 3 phases are required.

sending & recieving buffers :=

↓ to send data.

sliding window protocols are used to transfer the data.

URG	ACK	PSH	RST	SYN	FIN
↓	↓	↓	↓	↓	↓
urgent regest	ack	request for push	Resize	Syn	N

phase ① : connection establishment using 3-way handshaking.

SYN - used for connection establishment

② 2nd phase : Data transfer :
⇒ sliding window protocol is used.

③ 3rd phase : Teardown phase
FIN is used in 3-way handshake protocol to terminate

⇒ To find error we use
1) checksum
2) Time-out
3) ack

unit-4 : not include.
→ crash recovery, wireless nw 802.11

address mapping protocol { bootp, ARP, RARP, DHCP

CITER - classful address

MT-3
3. Routing protocols
conjuction control
tunnelling IPv4-IPv6
inter n/w, Routing
Packet fragmentation
↓
flag, seq no,