Module = III

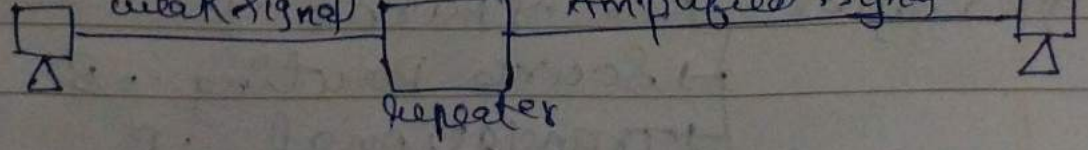
Network Layer

* Responsible for source-to-destina" delivery of a packet whereas the DLL oversees the delivery of the packet b/w 2 systms on the same ntwrk.
* If 2 systms are connected to same link there is no need for a n/w layr. If 2 systms are attached to diffrnt n/w, ~~with connecting devices b/w the n/w.~~ There is often a need for the n/w layer to accomplish s-to-d delivery.
* ( )s
  - Logical Addessing
  - Routing
  - packetizing
  - forwarding

I Networking & Internetworking devices

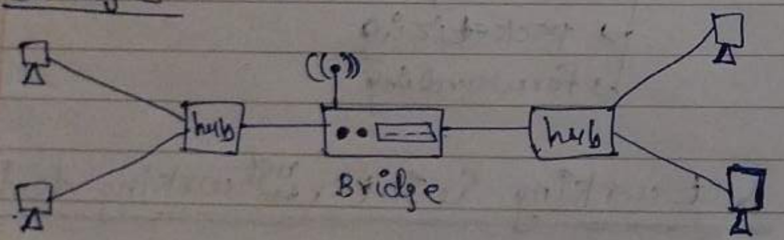
A comp N/w is built using several components. These components togethr makes it possible to transfer data from 1 device to anthr.

1) Repeaters :



* N/w-ing device that is used to amplify and generating the incoming signal.
* works at physical layr of OSI model.
* Main aim is to rise the n/w-ing distnce by rising strength & quality of signals.
* Using this hlps to reduce error & loss of data & provides with delivery of data at specified loc only.
* Adv → data trasfer with more security & over a long distnce.

2) Bridge :

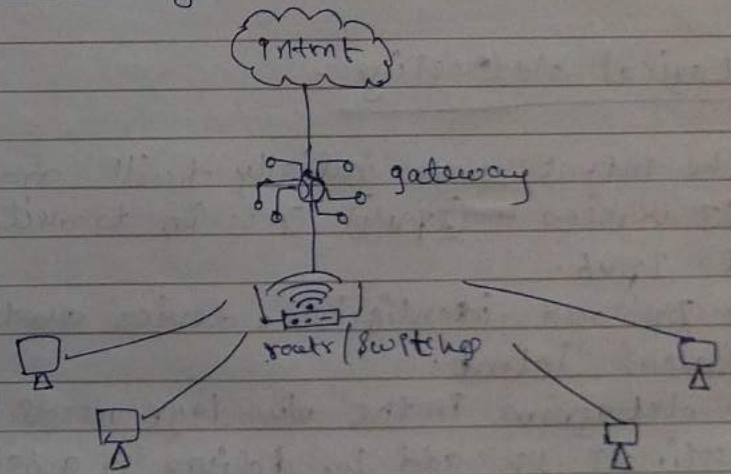


* N/w-ing device that is used to connect multiple LANs to a largr LAN.
* The bridge is a physical / hardware device it operates at the OSI model of DLL & also → layer of 2 switches
* 3 types → Transparent B
  → Source Routing B.

* Adv → used as n/w extnsion.
  ↳ has separate collision domain.
* dis → expensive compared to hub.
  ↳ slow in speed.

3) Routers : (refer 1st cp)

4) Gateway :



* Are the backbone for preserving the intellignce as well as the storage capacity of smart device.
* Can be a combination of both s/w & h/w
* It is a kind of gate that connects smart obj to the servers by appli- of fog computing.

* Create a structural temporary store room for the data transmitted by the srvr & data reqsts made by the user end.
* It make data more secure.
* 2 types → unidirn. cr
  → Bidirn. cr
* eg → IOT. cr, N/w. cr, cloud storage cr

II Logical Addressing :

* The intrnt was initially built around IP version 4 (IPv4) & is in transition to IPv6.
* A IP add identifies a device on the global intrnt.
* A datagram in the n/w layr needs a destinatn IP add for delivry & a source IP add for destinatn's reply.
* n/w addresses are always logical
* IP add identifies both, a n/w & the host on that n/w.

IPv4	IPV6
* Has 32-bit add length	has 128-bit add.
* Here, end to end connn integrity is unachievable	— — achievable.
* can generate 4.29×10^{9} add space	— — 3.4×10^{38} add space. (larger than ipv4)
* Fragmentatn is performed by sndr & forwarding routers	performed only by the sendr.
* Here, check sum field is available	— — not available.
* Has headr of 20-60 bytes	has header of 40 bytes fixed.
* consist of 4 fields separed by add dot(.).	consist of 8 fields separated by :

IPv4 add format :

1000000 0001000 0000011 0001111

128.11.3.31

IPv6 add format:

ABCD : EF01 : 2345 : 6789 : ABCD : B201 : 5482 : D023

← 16 bytes →

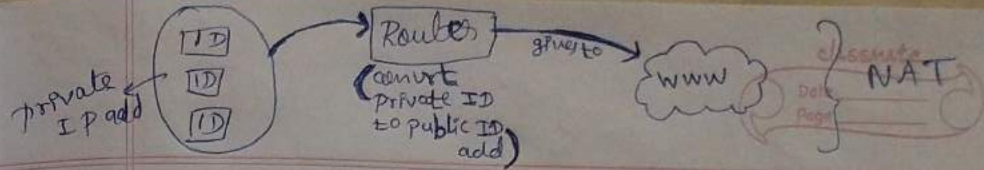
III N/w add Translation (NAT):

* Allow multiple devices to access the intrnt through a single public add. To achieve this, the translation of a private IP add to public IP add is required.
* NAT is a process in which 1/more local IP add is translated into 1/more global IP add & vice versa, in order to provide intrnt access to local hosts.
* Also it does not translate port no.
* Generally operates on router/firewall.

* Types —

1) <u>Static NAT:</u>

- Also → 1-to-1 NAT
- here, a single unregistered (private) IP add is mapped with a legally registered (public) IP add. (i.e) 1-to-1 mapping b/w local & global add.



- Generally used for web hosting.

2) <u>Dynamic NAT:</u>

- here, mapping of IP frm an unregistered private n/w is done with the single IP add of registered n/w.

3) <u>port add Translation (PAT):</u>

- Also → NAT overload.
- many local (private) IP add can be translated to a single registered IP add.
- port non are used to distinguish the traffic.

4) <u>overlapping NAT:</u>

- occurs whn an IP add is assigned to a device on a n/w that is already legally owned & assigned to diffrnt device on intrnt.

VI <u>Transition mechanism frm IPv4 to IPv6:</u>

* Instead using IPv6, we use combinaⁿ

of both & transition means not replacing IPV4 but co-existing of both.

* whn we want to send a reqst frm an IPV4 add to an IPV6 add, bt it isn't possible, bcz IPV4 & IPV6 transition is not compatible. for this we use some technologies —

a) **Dual-stack routers**:
Here, a router's intrface is attached with IPV4 & IPV6 add configured are used in ordr to transition frm IPV4 to IPV6. It is the complete duplication of all levels in (P) stack frm appln to the N/w layer.

b) **Tunneling**:
It is used as a medium to communicate the transit n/w with the diffrnt IP versions.

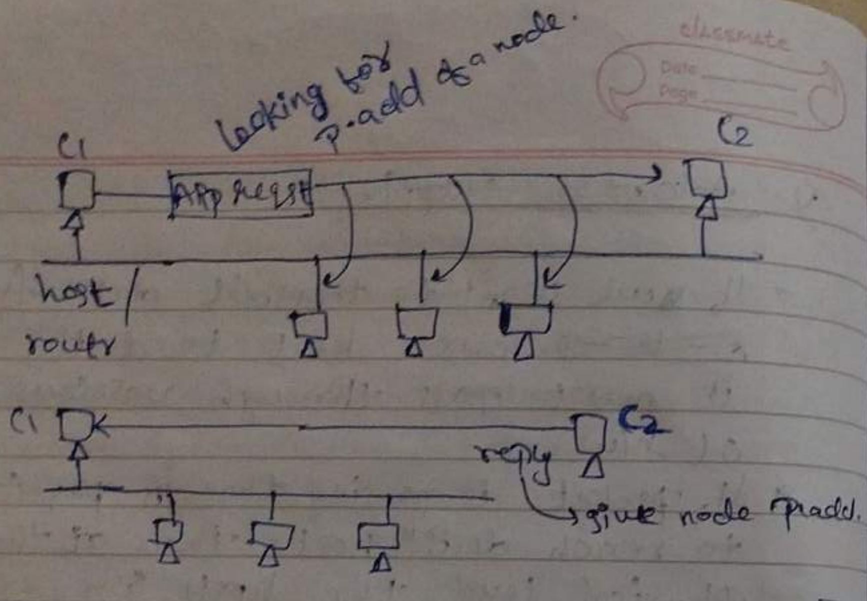
c) **Headr Translation**:
It is necessary whn the majority of the intrnt has moved to IPV6 bt some systms still use IPV4.
Here, the headr of IPV6 packet is converted to IPV4 headr.

V) Address Mapping:

- If we want to transmit data from ~~S to D~~ source host to destin host it must pass through various layr of n/w
* If packet is passing through physical n/w to reach destin hosts then at the physical level, the hosts are recognized by their physical add.
* A physical add is a local add.
* The delivery of a packet to a host requires 2 levels of addressing — physical & logical.
* So we need to map a logical add to its corresponding physical add & vice versa.

* ARP (Add Resolun protocol):

- converts L. add into P. add.
- used to associate an IP add with the MAC add.
- Each device on the n/w is recognized by the MAC add imprinted (processor) on the NIC.



IPv6 : An internet v6 add. is a numerical label a n/w interface of a comp/ new node participating in an IPv6 comp n/w.

3 add method
- unicast → act as identifies for a single interface
- multicast → defines grp of ~~devices~~ comp receiving a copy.
- Any cast → defines a grp of comp that all share single add.

VI Error reporting :

* ICMP (Intrnt control msg protocol) does not crt errors, it simply reports thm.
* Error - correction is left to the highr-level protocols.
* Error msgs are always snt to the original source, bcz the only info. available in the datagram about the route is the source & destin ip add.
* ICMP uses the source Ip add to snd the error msg to the source of the datagram.
* 5 types of errors are handled —
  * Destina^n unreachable
  * Sourch quench
  * Time exceeded
  * paramtr prbms
  * redirection.

  } error reporting.
* Snding a msg to a grp → multicasting & its roting (al) → multicasting routing
* Snding a packet to all destina^n simultanously → broadcasting.

IPv4 binary nottation :

	1st byte	sec	third	4th
cls A	0			
cls B	10			
cls C	110			
Cls D	1110			
cls E	1111			
~~cls F~~				

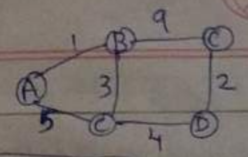
~~cls A 0-127~~
~~cls B 128-191~~
~~cls C 192~~

Routing

non adaptive (r)
* Static.
* does not change

Adaptive (r)
* dynamic
* link state (r).
* Distnce vector.

* Distnce vector :



	Receivr	Cost	Intrmediate
B→C	C	3	C
B→E	E	9	E

1) Dynamic (R) attempts to solve this prblm by constructing (R) tables automatically based on info carried by (R) protocol allowing the n/w to act nearly autonomously in avoiding n/w failure & blockages.
2) In distance vector (R), the least cost route b/w any 2 nodes in the node with min distnce. Here, each node maintains a table of min distnce to evry node.

VII) Delivery =

* Moving data frm 1 n/w device to anthr requires several diffrnt ( )s.
* Each ( ) has its own protocols that define hw it is accomblished.
* Main facter in data delivry is determining whethr 2 devices are directly connected / remotely connected.
* If the sending & receiving devices are connected to the same broadcast domain, data can be exchanged using switch & MAC add.

If they are connected to diffrnt broadcast domains, the use of routr & Ip add are required to exchange data.

IX) Forwarding & Routing =

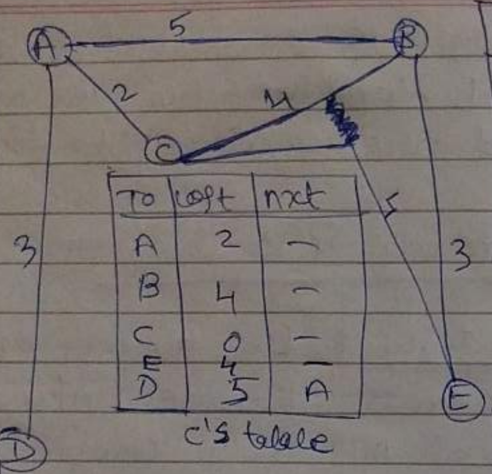
* N/w routing is the process of selecting a path across 1/more n/w.
* A physical n/w is a combinª of n/w (LAN & WANs) & routers that connect thm.
* Routers use routing (al) to construct routing / forwarding tables.
* Forwarding is the action applied by each router whn a packet arrives at 1 of its interfaces.
* Forwarding requires a host / a router to hve a forwarding / routing table.
* Forwarding techniques
  - → Nxt-Hop Routing
  - → N/w-Specific Routing
  - → Default Routing.

* Routing (al) —

1) Distnce-vector Routing :

* Here, each node maintains a vector (table) of min distnces to every node.

nxt → intermediate node



To	cost	nxt
A	0	–
B	5	–
C	2	–
D E	3 6	– C

A's table

To	cost	nxt
A	5	–
B	0	–
C	4	–
E	3	–
D	8	A

B's table

To	cost	nxt
A	2	–
B	4	–
C	0	–
E	4	–
D	5	A

C's table

To	cost	nxt
A	3	–
B	8	A
C	5	A
D E	0 9	– A

D's table

To	cost	nxt
A	6	C
B	3	–
C	4	–
D	9	C
E	0	–

E's table

* Initializaⁿ : At the beginning, each node can know only the distnce b/w itself and its intermediate neighbours, those directly connected to it.
* Sharing : whole idea of distnce V. routy is the sharing of info b/w neighbours. Each node share its routing table with its immediate neighbours.
* updating : when a node receives a 2-column table from a neighbour,

it need to update its routing table

2) Link state routing :

* Here, 4 sets of actions are required to ensure that each node has the routing table showing the least-node to every other node –

1) Create LSP (link state packet)
2) Flooding
3) Formation of shortest path tree of each node
4) calculation of a routing table based on shortest path tree.

↳ LSPs are generated on 2 occasions –
- when there is a change in the topology of the domain
- on a periodic basis.

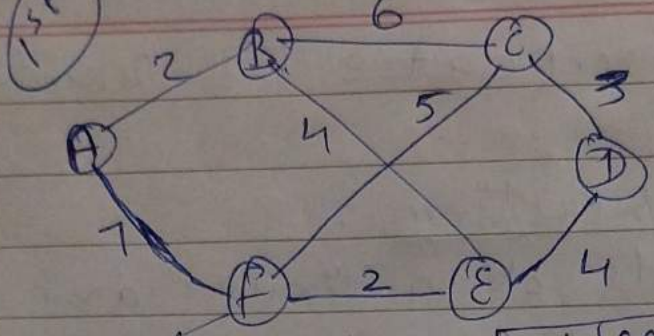
↳ After a node has prepared as LSP, it must be desseminated to all other nodes not only to its neighbors.

↳ using Dijkstra (al)

N	cost
A	2
C	6
E	4

→ B

1st



N	cost
B	2
f	7

A's table

f ൽ നിന്ന് എവിടേക്ക്
കുറഞ്ഞ പോകും
put that in table

node	cost
C	5
A	7
E	2

F's table

N	cost
D	3
B	6
F	5

→ C

N	cost
E	4
C	3

D

N	cost
D	4
f	2
B	4

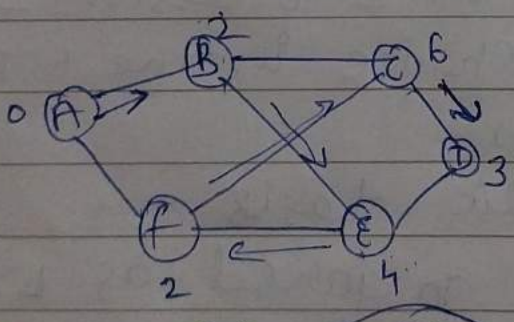
E's table

2nd → flooding

3rd step

A(0) → B (2), f (7)

ഇതിൽ B-ആണ് short.



ഒരു വട്ടം മാത്രമേ
പോകാൻ പാടൂ.

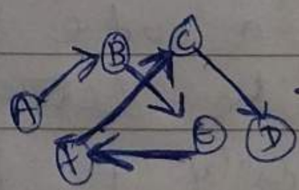
A, B → C(6), E(4)

A,B,E → f(2), D(3)

A,B,E,f → C(5)

A,B,E,f,C → D(3)

~~A,B,E,F,C,D~~



→ shortst path