a.b.c.d → class addressing
NID | HID
32

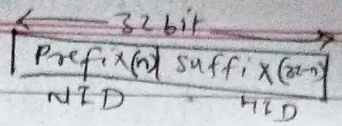
Computer Network - EndSem

classmate
Date
Page

• classless addressing :

a.b.c.d/n

n → NID or subnet mask



eg: 10.19.32.54/22 ⇒ NID = 22 bit
HID = 32 − 22 = (10) bit
∴ No. of IP addresses = 2^{10} = (1024)
But no. of Host = $2^{10} - 2$ = (1022) ★

or can be said
no. of IP addresses in the BLOCK = 2^{32-n}

first address & last address

Keep the n leftmost bits & set the (32−n) rightmost bit all to 0

Keep the n leftmost bits & set the (32−n) rightmost bits all to 1

CIDR

When ever any customer wants a BLOCK of IP addresses IANA or ISP will create the BLOCK assigned to customer.

Rules to be followed by IANA or ISP for creating a BLOCK

1. All the IP address in the block must be contiguous.
2. Block size must be power of 2.
3. first IP address of the block must be divisible by size of the block.

first IP address must be used as block-Id

Representation of CIDR

Block size = 2^6 HID = 6 bit, NID = 26 bit

n−K bit	K-bit
NID	HID (K bit)

Supernetting :

- Combines multiple smaller networks into one large address block.
- Allows fewer routing table entries & more efficient address range

Subnetting → divides a single network into multiple subnets (adds more 1s to mask)

Supernetting → Combines multiple networks into one large network (removes some 1s)

classmate

Supernetting eg

Needs 1050 IPs

(i) We are assigning 4 contiguous class C blocks to a company

192.60.128.0/24
192.60.129.0/24 → as it's class C
192.60.130.0/24
192.60.131.0/24

each block has 256 addresses (since it's /24)
So total = 1024

(ii) Represent in Binary (to understand supernetting)

IP address	3rd octet (Binary)	3rd octet decimal
192.60.128.0	128	10000000
192.60.129.0	129	10000001
192.60.130.0	130	10000010
192.60.131.0	131	10000011

They differ only in last 2 bits of the 3rd octet. So these addresses can be combined into a supernet that covers all 4

(iii) find the supernet prefix

we need a prefix that → covers all 4 networks & shared same first bits in binary

∴ shared prefix length = (22) bits.

∴ Supernet = 192.60.128.0/22

Netmask = 255.255.252.0 → 11111111.11111111.111111 00.00000000
first 22 bits are network | last (10) bits for hosts.

(iv) what's happening in routing tables?

- At Higher level routers

Instead of 4 entries just one is needed (This reduces routing table size)

192.60.128.0/22

- At internal routers

192.60.128.0/24 – router 1
192.60.129.0/24 – router 2
192.60.130.0/24 – router 3
192.60.131.0/24 – router 4

Q. Why communication can't only happen with IP? why MAC is needed?

→ IP address works at Network layer (3) & actual physical delivery of data (sending frames over ethernet or wifi) happens at DLL (2) – and layer 2 needs MAC addresses.

IP is a logical address (Not tied to H/W). They can change when we switch networks (home, cafe, college). IP is used for routing across networks – it tells where the data

MAC → physical address (tied to H/W)

- MAC address is burned into the NIC at the factory.
- It never changes (unless spoofed).
- MAC is used for actual delivery inside a LAN.

Analogy :

IP – Postal address that tells which city or street to deliver mail to.
MAC – Name on a mailbox at a specific apartment.

So, post office (Router) delivers the mail to the building (IP), but inside the building, someone needs to know exactly which person gets the letter.

ARP

Every node has a ARP module, that can create a MAC frame for knowing the MAC address of receiver's node.

Ethernet frame format :

PA (Preamble)	SFD	DA	SA	LEN	LLC PDU	PAD	FCS	IEEE 802.3
(7)	(1)	(6)	(6)	(2)	← 46-1500 →		(4)	

PA	DA	SA	Type	DATA	PAD	FCS	– DIX Ethernet
8	6	6	2	← 46-1500 →		4	

Let Node N needs to know the MAC addr of node R.
↳ N already knows R's IP.

So, N's ARP module creates & sends a MAC/ethernet frame where

- Type field is set to 0x0806 (ARP packet)
- DA (Dest MAC) address → FF:FF:FF:FF:FF:FF
- SA (Src MAC) → set ✓
- frame contains the IP address of R (whose MAC is required), IP & MAC address of N.

This MAC frame reaches ARP module of all nodes, but only R replies.

255.255.254.0
11111111.11111111.11111110.00000000
23
128.96.170.0
10000000.01100000.10101010.00000000
10000000.01100000.10101011.11111111
128.96.170.0 – 128.96.171.255

169 = 10100100 . 0000
167.255

Hierarchical & geographical Routing

↳ Aim → to reduce the size of forwarding/routing tables

1. Hierarchical routing:

- Problem – If every router on the internet stored routes to every other host, the routing table would be too large & slow.

Sol → Hierarchical routing

Instead of flat routing, we divide the internet into layers of ISPs

Structure:

- Local ISP (Small) → connects homes, offices & universities
- Regional ISP (Med.) → connects many local ISPs
- Global ISP (Large) → connects regional ISPs.

Every level knows routes only within its zone, not the whole internet

② Geographical Routing:

To decrease the size of Routing table even more, the hierarchical routing is extended to geographical routing.

↳ entire address space is divided into few large blocks.

- Assign a block to Asia, one to Europe, one to America
  - → Router of ISPs outside Europe will have only one entry for packets to europe in the forwarding table.

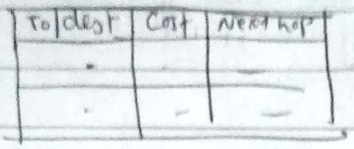
& so on.

Distance vector Routing

- Each Router (say node N) keeps a distance vector, i.e. a table that tells:
  - cost to reach every other node
  - next hop to use for each destination.

Working:

- each node periodically shares its distance vector with all its nbrs
- updates help routers in learning new/better paths
- even if nothing changes, the dv is still sent regularly.
- If a node doesn't receive updates from a neighbour for some time, it assumes the link is down (broken or unreachable).



assuming node N crashed:
- all routes with next hop N are set to have metric INF,
- Routes with metric INF will be deleted after some time, defined by actual protocol.

eg. of distance vector Routing protocol → RIP (Routing information protocol)

In case of RIP: INF = 16
Cost of each link = 1
Broadcast interval = 30s
Route expiry time = 180s. ⇒ If no message received from a neighbour N for 180s, change metric to INF for all routes through N

Time to delete route = 120s after a metric is set to INF

Routing protocol
- Interdomain
  - Path vector (BGP)
- Intradomain
  - distance vector (RIP)
  - link state (OSPF)

RIP → Routing information protocol
OSPF → open shortest path first

eg. Routing
- fixed or static
- Random
- flooding
- Dynamic or adaptive.

Updating rules of a routing Table :

- No news → do not change
- Same next Hop → replace
- A new router added → add
- Different next Hop
  - new Hop count smaller → replace
  - new HC same (don't change)
  - new HC larger → not change

- If next hop is same → replace
  else replace only when new HC is smaller

Link state Routing

- Discover your nbrs. (by sending a 'hello' packet (contains own IP) on all outgoing links).
  - gets the reply from all nbrs with info about (IP) & of nbrs
- Measure the delay of your neighbours using ping
  - send our echo packet, nbrs send it back immediately.
  - RTT/2 = delay of your nbrs
  - Take average delay by doing it several times.
- Build the LSP.
  - LSP sent by node N contains identification of the generating route, sequence Number, TTL, list of identifications & distances to the nbrs
  - Ensure that no older ver of LSP (sent prev. by N) is used by other nodes. A Node sends LSPs with continuall ↑ sequ numb
- Distribute the LSP by flooding to all Nodes.
  - LSP not sent to the neighbour from whom it is obtained.
- Calculate the shortest path to all other nodes.
  - once a node has received LSPs from all other nodes, it knows the entire network
  - uses Dijkstra's algo locally to comp. Shortest path to all nodes.

TCP/UDP

TCP connection establishment

Purpose: both side needs to know ① that the other side is ready for data transfer ② Port used by other side ③ MSS ④ other side's (ISN)

What is ISN? - initial sequence number

- It's the first sequence number used by each side when a TCP connection starts.
- It makes the starting point of numbering the data bytes sent.

1. ISN Cannot always be 0 or 1. Why?

Because TCP connection can close & reopen quickly using the same IPs and ports. If we always start with 0 then,

- old segments from the previous connection might still be in the network
- The new connection might accept those old packets by mistake, thinking they belong to the new session.

How TCP avoids this problem?

Each side chooses an ISN randomly. This makes it very unlikely that 2 connections will have the same ISN & confuse old data with new.

How ISN is generated randomly?

often based on a 32-bit clock ticking every 4 µs. Since its 32-bit clock it wraps around after $2^{32} \times 4\,\mu s = 4.55$ hours.

So, even if connection closes & a new one starts soon after, the ISN will likely be very different.

3 way Handshake :

1. client send a SYN segment (no data)
   - specifies port no., sequence number of client (isn_c)
2. Server responds with (SYN+ACK)
   - Seq No: Server's ISN (isn_s)
   - Ack no: isn_c+1
3. Client send ACK
   - ACK no: isn_s+1

Connection termination

- Any of the two parties involved in exchanging data (client or server) can close the connection, although it is usually initiated by client.

Most implementations today allow two options for connection termination:

(i) 3-way Handshaking
(ii) 4 ———— with a half-close option

Normal connection Termination:

(i) client sends FIN

- when the client process is done, it tells TCP to close the connection
- TCP sends a FIN segment (with the FIN flag set)
- FIN segment can include the last chunk of data sent by the client or it can be just a control segment.

(ii) Server send FIN+ACK

- The server receives the FIN, and says: "Okay! I got your FIN"
- It replies with a FIN+ACK segment.
  - → ACK confirms Client's FIN
  - → FIN announces that the server is also ready to close
  - → This segment can also include last data from server

(iii) client sends final ACK

- The client receives the server's FIN & sends a final ACK
- Now both sides know the connections fully Closed.

NOTE: Even if no data is sent with FIN, it still uses one seq
TCP ensures a clean shutdown, with no data loss.

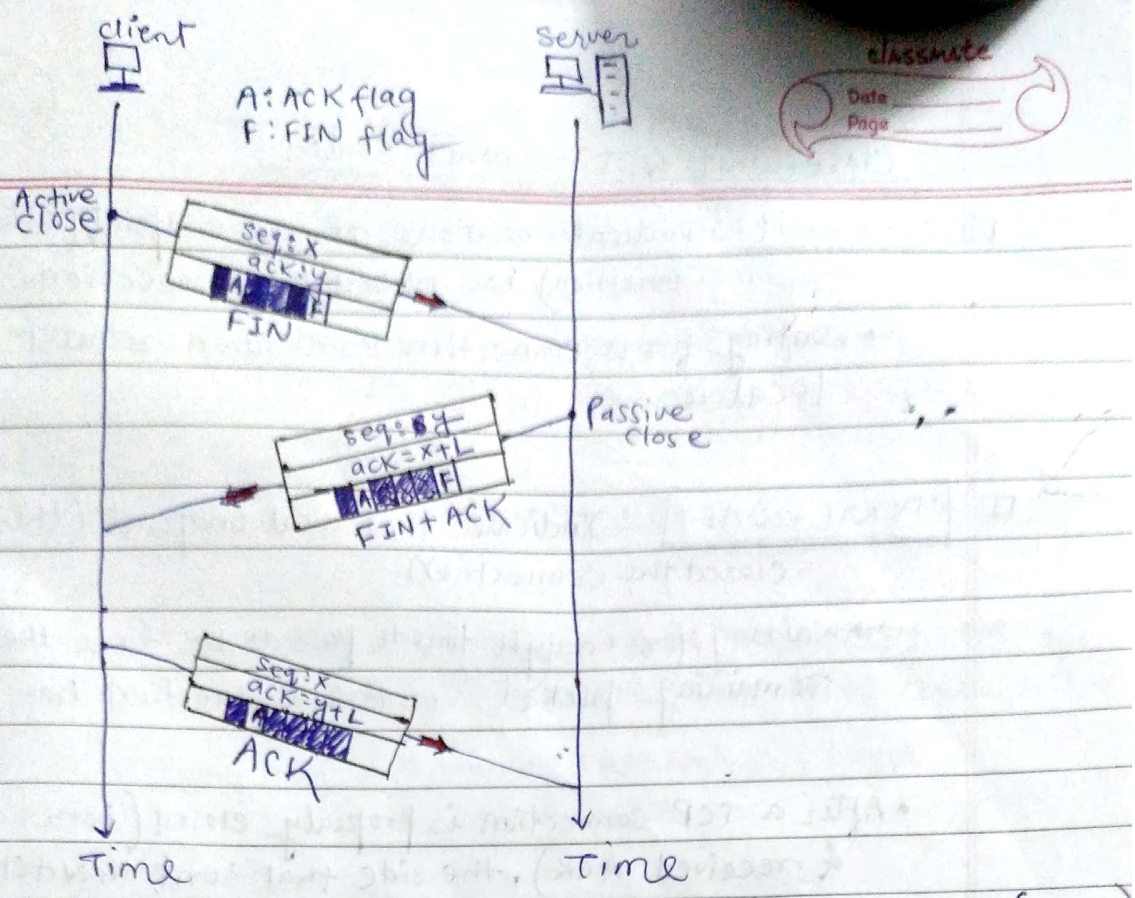


fig: connection termination using 3-way Handshaking (normal)

Half-close connection termination:

TCP connection is duplex: data flows in both directions

A Half-close means:

- one side says - I am done sending data
- But the other side can still send data

client		server
FIN segment → with sequence no.		(client done sending)
	← ACK segment (server confirms client's FIN) with Acknowledgement NUM = FIN segment SN+1	
	← Data (~~server still sending~~)	
	← FIN	(server done too)
ACK	→	(client confirms)

- see diagram from pdf.

Close-wait & Time wait states

- **Close-wait:** Indicates that the remote endpoint (other side of the connection) has closed the connection.
  - → waiting for a connection termination request from the local user.

- **TIME-WAIT:** Indicates that local endpoint (this side) has closed the connection
  - → waiting for enough time to pass to be sure that all remaining packets on the connection have expired.

- After a TCP connection is properly closed (both side sent FIN & received ACK), the side that sent the last ACK enters the TIME-WAIT state.

How long → 2 x MSL (max. segment lifetime)

MSL is usually 2 minutes

∴ TIME-WAIT lasts for about 4 minutes

Analogy – we've closed the door, but let's wait a bit to make sure no leftover messages are coming.

def, Adv, disadv, layer, example, delay, switch type

Circuit Switching → (establish, data transfer, disconnect)
- all data follows same path, arrive at dest in order
- Switching at physical layer in traditional tele phone network.

Packet Switching:

Data broken into packet, each packet of (DIP / SNO) approach

(i) Datagram approach → Packet treated independtly, switch uses routing table
- No dedicated path, arrive out of order, lost ✓
- Switching at Network layer in Internet.

(ii) Virtual-CKT approach
- Cross b/w CKT SN & datagram network
- Packet follows same path, In-order arrive.
- Technique → (Setup Phase, DT Phase, Teardown Phase)
- Each packet contains VCI, not DIP. (like datagram)

eg:
- Switching at data link layer (X.25, Frame Relay, ATM)
- links in path are not dedicated, may be shared among different virtual circuits.

Network address → all bits of hostid postion equal to 0

DBA →

* No host in network should be given a IP whose hostid is either all 0 or all 1.

Loopback address → The network prefix 127.0.0.0 (a value from class A range) reserved for loopback.
- self communication on the same machine.
- used for TCP/IP and for IPC within a host.
- 127.0.0.0 | 8

limited Broadcast → 255.255.255.255. (LBA on this (local) network)

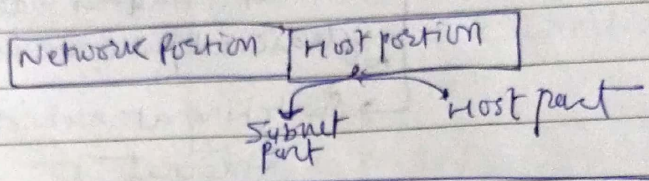
this host on this network → 0.0.0.0

Limited & directed Broadcast address:

LBA → limited because it is never forwarded across routers
used within a local network.

DBA → directed because can be forwarded across routers to reach a specific host.
used for broadcasting from a remote network.

Subnet



Last byte	Host num range	Network addr	Subnet Mask

IP addr AND Subnet Mask = N ID

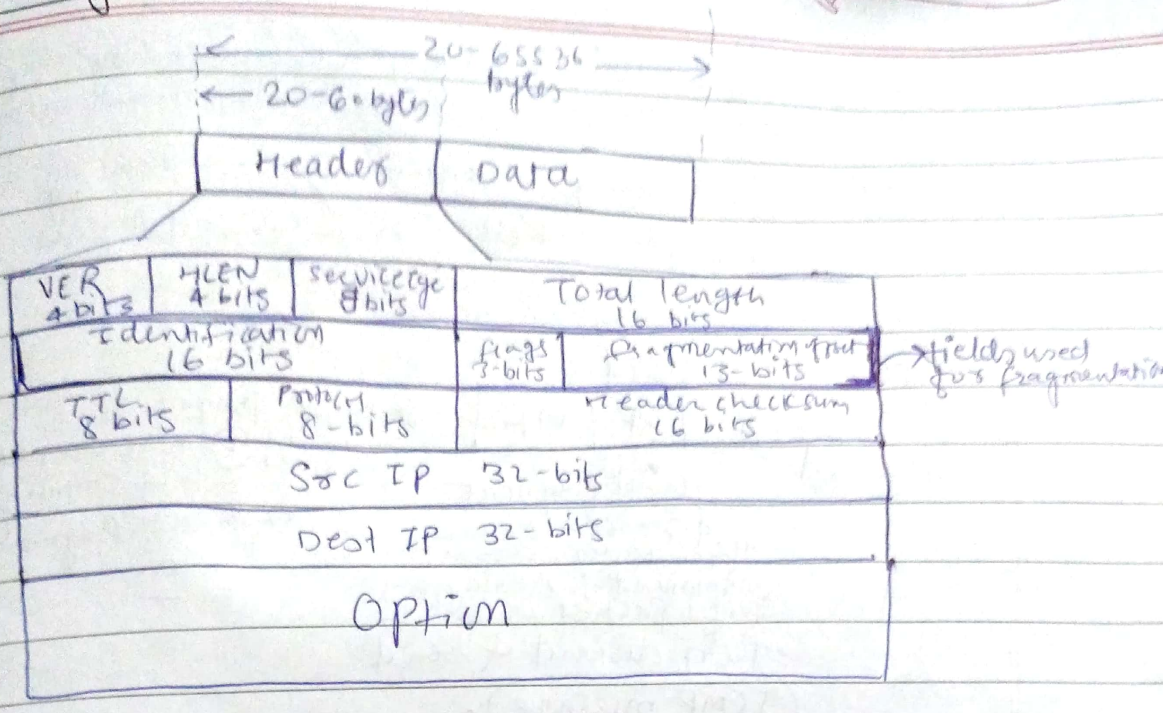
ARP Packet format

Hardware type		Protocol type
Hardware length	Protocol length	Operation 1: Req, 2: Reply
Sender H/W address		→ eg: 6 byte for ethernet
Sender Protocol addr		→ eg: 4 bytes for IP
Target H/W address		→ It is not filled in a req
Target Protocol address		

Hardware Type → 1 ⇒ Ethernet interface
node can have multiple H/W interface like ethernet, FDDI interface.

Protocol Type → which Higher level protocol is being used (IP or some other)
containing value $(0800)_{16}$ for IP

IP datagram :



Checksum Calculation :

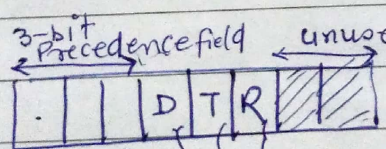
- Break IP header into 16-bits units (checksum field = 0)
- Sum these units, using 1's complement arithmetic.
  Sum → . . . .
- Take 1's complement of the sum

checksum = 1's complement of sum.

This checksum will be verified & re-computed at each router & at the final destination.

NOTE: IP checksum computed only on IP header, NOT on the data in the packet.

Service field

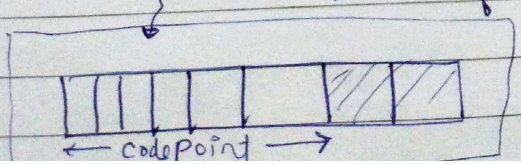


datagram Precedence
0 = normal data
7 = Network control

D → delay
T → Throughput
R → Reliability

Out of D, T & R → only one bit can be set
If no one set ⇒ Normal service

1990 में redefined किए ⇒ redefined this field to Provide Diffserv use for QoS.

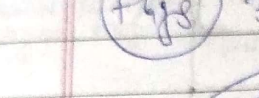


Every network has a MTU (maximum transfer unit).
If the Router has to transmit datagrams too large (larger than MTU of the outgoing link), datagram is fragmented.

identification → unique no added by sender for each datagram.
Helps the dest knows which fragments belongs to which.

dest node uses <source IP, identification> to identify which arriving fragment belongs to which datagram

flgs 3 bits field



reserved

DF → do not fragment
if source set it to 1, then routers does not fragment the datagram & if packet is too big then discard & send ICMP message.

MF → more fragments
if 1 more fragments are coming
So, in last fragment it will be 0.

fragmentation offset : 13-bits
↳ shows where this fragment's data fits in the original message.
eg: if offset = 2 ⇒ this fragment starts at byte 2×8 = (16) in the original datagram.
▲ Given in units of 8-byte blocks ?

→ **option field** → options included primarily for network testing & debugging
eg: Source routing, Record route.

Q. Short Notes of ICMP (4 Marks)

ICMP is a required support protocol used by routers along with IP & ARP to
- report errors (does not fix)
- Provide feedback (like diagnostics)
- Monitor the network (with ping)

When any node detects an error, an ICMP packet sent back to the source.
eg (router can't deliver a packet because its too big or unreachable)

When you run the Ping command, your system sends ICMP Echo Request & the destination replies with echo reply.

- ICMP packet contains ICMP header + some message specific data.
- ICMP packet is sent inside an IP packet in the data part.

So, ICMP doesn't have its own delivery system, it relies on IP to carry it.

use-cases of ICMP:
- Echo reply (to see if host is up)
- Subnet mask req & reply (among routers) → used by routers to share subnet info.
- Router informs sender about packet drop using this.

- Types of Routing → fixed / static, flooding, Random, dynamic / adaptive → IMP for short notes (see)

DLL functionalities :

- This layer is responsible for node to node comm. & ensures reliable data transfer over a physical link.
- framing - converts RAW bits into structured frames
- Error detection, error control → Implement CRC & ARQ
- flow control → prevents fast senders from overwhelming slow receivers
- medium access control - Manages how multiple devices share the same medium (CSMA/CD, CSMA/CA)

Network Layer :

Transport Layer :

UDP → 1. Process to process delivery

TCP → 2. Connection Establishment & Termination (End-2-end)
3. Guarantee Reliable, In-order delivery
4. flow control (provides end-2-end)
5. Congestion control.

flow control → Rx can limit how many bytes the sender can send at a given

Congestion control → Control how fast Sender can send data, to prevent the Sender from overloading the network.

2D even parity:

a) Row parity — 1 parity bit per [illegible]

Col P —

→ this helps detects single bit [illegible]

Given bitstream:

	col 0	col 1	col 2	col 3	col 4	col 5	col 6	col 7	
Row 0	1	1	0	0	1	0	1	0	
Row 1	0	0	1	1	0	0	1	1	3
Row 2	1	0	0	0	1	0	0	0	2
Row 3	1	1	1	0	1	0	1	1	6
	1	0	0	1	0	0	1	0	5

~~Column~~ Parities.

for actual data, we need to ~~extract~~ remove row & col parities.

data → 1100101 0011001

As there we no errors in actual info part

Row 0 ✓ Row 1 ✓ 2 ✓ 3 ✓

col 0 ✓ col 1 ✓ 2 ✓ 3 ✓ 5 ✗ 6 ✓ 7 ✗

Frame discarded ✓

Just: Even though we assume no error in the actual dpart, the parity check fails ----, so the receiver will reject the frame as it violates the even parity rule.

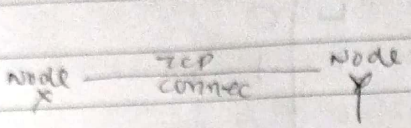
1 2 3 ✓

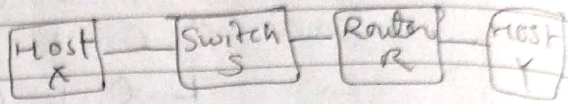
4 5 6

5 6 7 (11)

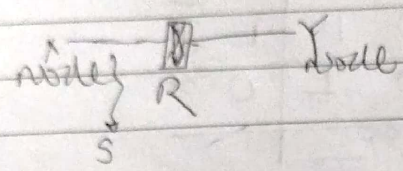
8 9

(c)





As per ques: X sends packet to Y.



Dest IP Address :

- destination IP address is the IP address of Y.

Even if the pckt passes through intermediate devices, the IP header's destination address does not change.

Dest MAC address

- MAC address of router's R's interface on X's local network.

MAC address is updated at every hop & points to the next device in the path. When the packet leaves X, the dest. MAC address is set to the MAC address of R (the next-hop router), not S or Y.

(e)

26.23.71.22 / 24

NID = 24 bit
HID = 8 bit
∴ No of IP addressing a block = 2^8 = 256

first Host = 26.23.71.1
last = 26.23.71.254

Network address = 26.23.71.0
DBA = 26.23.71.255

110.33.61.193 / 28

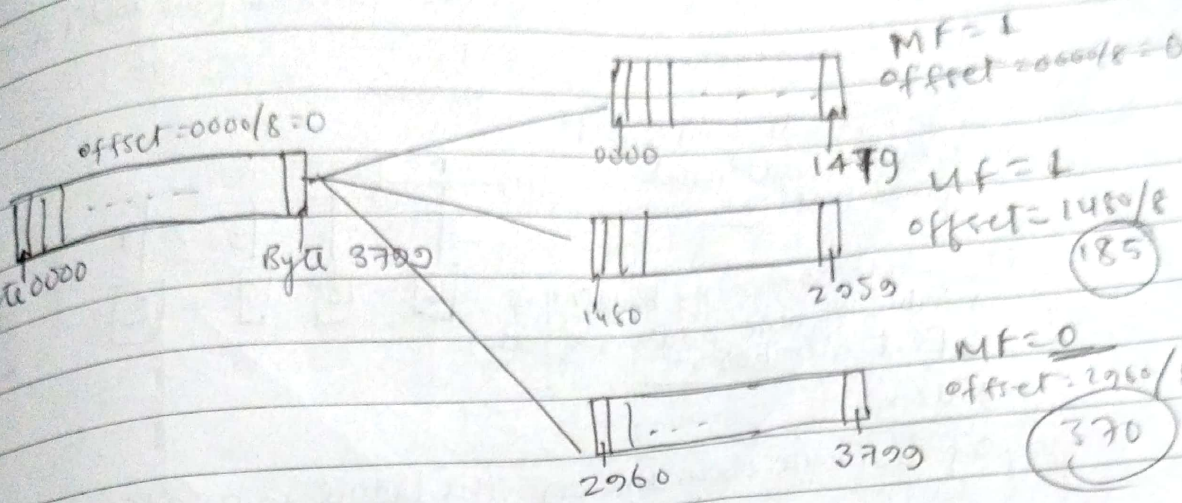
1100 0001

NID = 28 bit
HID = 4 bit
Block size = 2^4 = 16

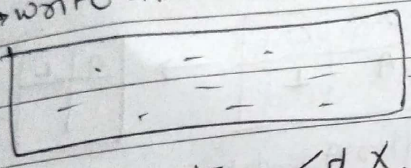
first Host = 110.33.61.193
last = 110.33.61.206

Network add = 110.33.61.192
DBA = 110.33.61.207

Payload = 3800 bytes, DF = 0 ⇒ fragmentation allowed, IP Header = 20, MTU = 1500



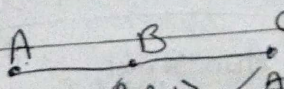
DVR → write about this in short



<dest, NH, cost>

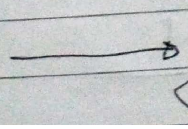
for every entry <d, X, K> in the B's distance vector
  if <d, –, –> is not in A's routing table
    add <d, B, K+L> to A's routing table
  else if <d, y, K'> exists in A's RT
    if K+L < K'
      replace <d, y, K'> with <d, B, K+L> in A's RT
    else if Y == B # trust your next hop
      replace <d, B, K'> with <d, B, K+L> in ——
    end if
  endif
end for

Count to ∞ problem



n down & B detects it has crashed

If B's routing table reaches C then no pr—

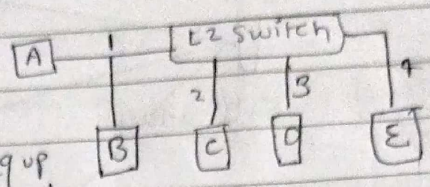


<A, B, 2> ← If C's RT reaches first

Count to ∞ Problem occurs in DVR Protocols when routers slowly update incorrect route info after a link fail.

→ when a route to dest node becomes unreachable, routers mistakenly believe another router still has a valid path & keep ↑ing the distance metric to that dest forever (or until a matrix INF does not reaches 16 in RIP).

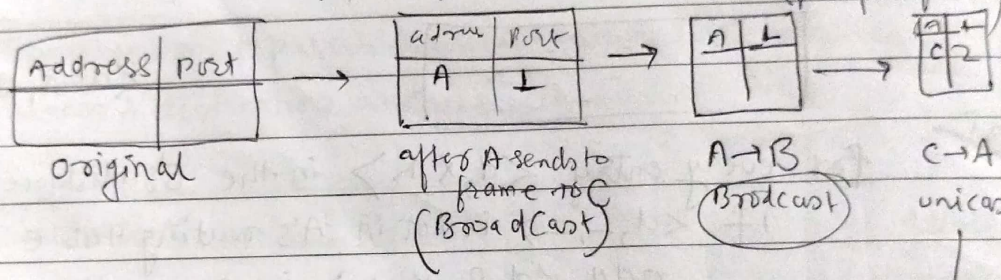
(4) A layer 2 switch is an N-port bridge with additional sophistication.

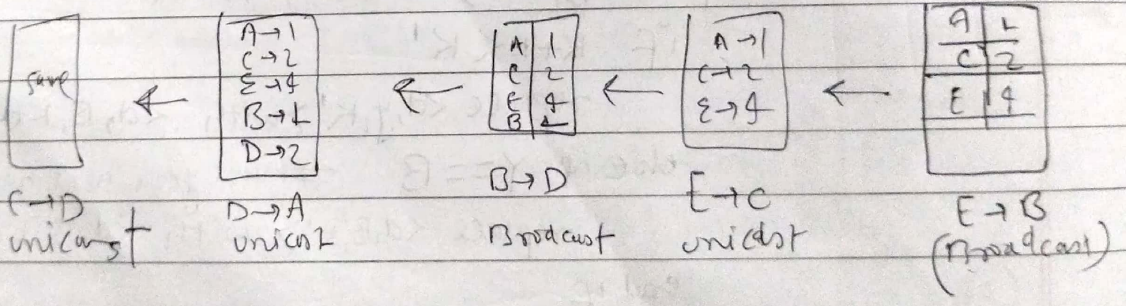


L2 switch learns by building up a table of MAC address of nodes & port numbers.

Rules:

- If dest. MAC is unknown, the frame is broadcast to all ports except the incoming port.
- Known, it performs unicast (sends only to that specific port)





(5) Subnet number = network address

↳ network part of an IP address after subnetting

Cal after bitwise AND between dest IP & subnet mask

∴ dest IP AND subnet mask = subnet number

- If one or more route match → select match with longest prefix
- If no match → default.

1 000 0000

IP block: 80.70.90.128/25

orgA → 56, B → 22, C → 30

(1) Sort the requirements (desc)

• orgA → 56 IPs

we need atleast 64 address → /26

Subnet: 80.70.90.128/26

Netmask: 255.255.255.192

Range → 80.70.90.(129 – 190)

DBA → 80.70.90.191

NID → 80.70.90.128

orgC → 30

atleast (32) addresses → /27

Subnet → 80.70.90.192/27

Netmask = 255.255.255.224

Range = (193 – 222)

DBA = (223)

(B) → /27

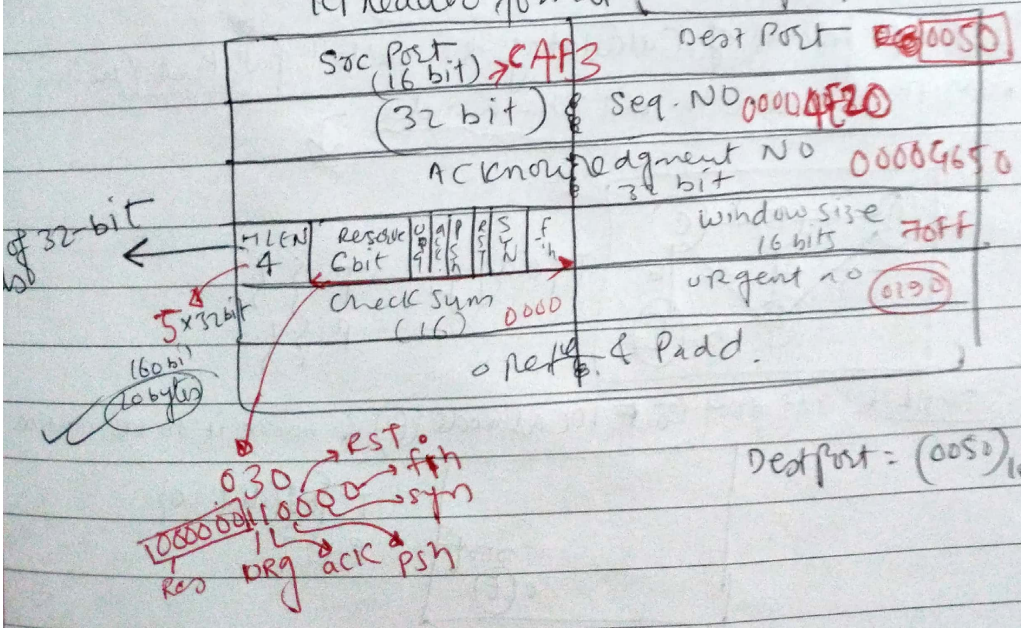
Subnet → 80.70.90.224/27

Netmask → 255.255.255.224

Rang → (225 → 254)

DBA → (255)

TCP header format (min 20 bytes)

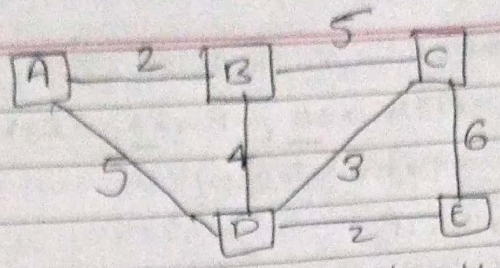


Dest Port = $(0050)_{16} = (80)_{10}$

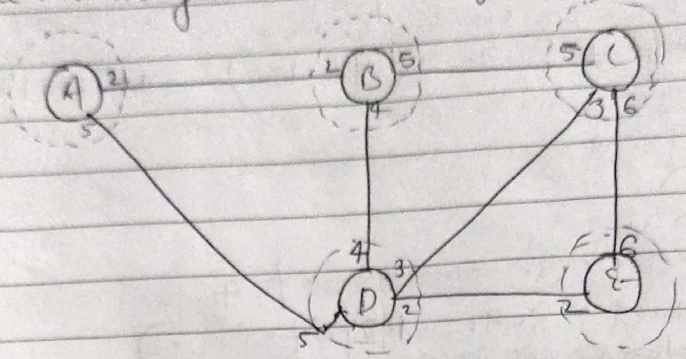
HTTP listens here

LS Routing

Q



link state knowledge – status & cost – of its directly connected links.



Initial LSN of every router

final RT of each Router after convergence:

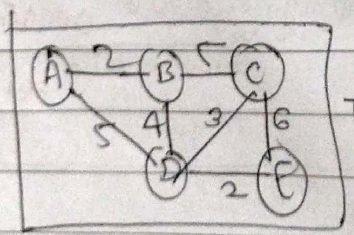
Routing table of A

destNode	Next Hop	cost
A		
B		
C		
D		
E		

for B, C, D, E →

D	NH	cost
A	D	7
B	D	6
C	D	5
D	—	2
E	—	0

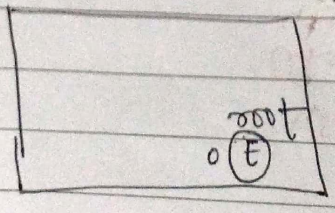
Step by step process of calculating the shortest path tree (using Dijkstra) only from node E to rest of the nodes



Topology

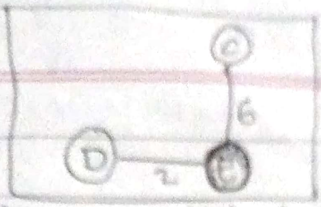
T = { }
P = { }

Step 1: Set root to local node (E) & move it to tentative list



T = {E (dist = 0)}
P → { }

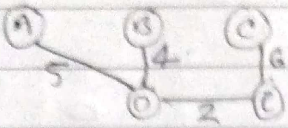
Step 2:



move E to permanent list & add D, C to tentative list
tentative → {D, C}, Perm → {E}
(dist=2) (dist=6) (dist=0)

~~Among nodes :~~

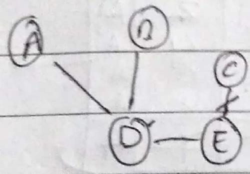
Step 3: move D to PL & add A, B ~~C~~ in TL



PL = {E, D} → lowest distance in TL (dist=2)
TL = { A (dist=7) ✓, B (dist=6), C (dist=5) }

Step 4: move C to ~~tentative~~ Perma. list (because C has the list distance in TL) & update their neighbours with new distance if less.

PL = {E(dist=0), D(dist=2), C(dis=5)}
TL = {A(dist=7), B(dist=6)}



2023,

when ~~ever~~ a collision occurs in CSMA/CD, the exp. backoff algorithm is used
: Each sender sends a jam signal & waits K × 51.2 μs (where 51.2 μs is the fixed slot time in 802.3 ethernet) and K is chosen randomly from 0 to $2^N - 1$.

N = Current transmission number
K = no. of collision

for this ques, N = 1 for A as this is A's first retran & N = 2 for B as this is B's 2nd Retrans. So possible value of K are {0, 1} (A), {0, 1, 2, 3} (B);
giving 8 possible combination of values.

A's Backoff (K)	B's backoff (K)	who wins?
0	0	collision
0	1	A
0	2	A
0	3	A
1	0	B
1	1	collision
1	2	A
1	3	A

the race is won by A if its K is lower than of B as this directly corresponds to the waiting time.

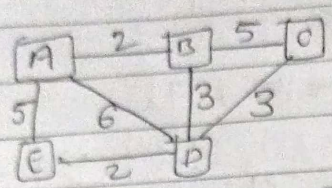
Mbps = megabits per sec

MBps = megabytes per sec

distance / signal speed

Transmission time ≥ 2 × Propagation delay

Transmission time = frame size / Bandwidth or ethernet speed



10,000 bits / 500 × 10^6

dist | NH | cost

A — 0

B — 2

C — 5

D — 6

E — 7

B

A — 2

B — 0

C — 5

D — 3

E — ∞

C

A

B — 5

C — ∞

D — 3

E

D

A — 6

B — 3

C — 3

D — 0

E — 2

E

A — 5

B —

C —

D — 2

E — 0

A, B, C, D.

A announces → B, E update. No other change

A sends its DT to B.

To	cost
A	0
B	2
E	5
D	6

Received from A

To	cost	next
A	2	A
B	4	A
C	∞	A
D	8	A
E	7	A

B' modified table

To	Cs	NH
A	2	—
B	0	—
C	5	—
D	3	—
E	∞	∞

B' old table

A	2	—
B	0	—
C	5	—
D	3	—
E	7	A

A sends to E

E → B → A (via A)

port

telnet = 23

FTP = 21, 20

web server / HTTP - 80

VLC media play → 1234

Radius auth. Protocol → 1812

TCP

MSS = MTU − IP Header size − TCP Header size

MSL lifetime = 120s

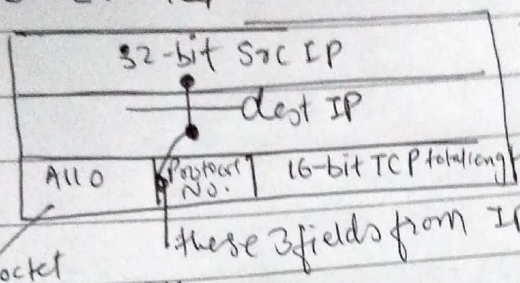
classmate Date Page

UAPRSF

TCP pseudo Header:

TCP pseudo header in Transport layer is used to verify that the segment has been delivered between the correct 2 endpoints.

It consists of 3 fields from TCP



these 3 fields from IP header

one octet of zeros for padding the segment so that it is multiple of 16-bits.

TCP pseudo header (& the octet used for padding) are not transmitted along with the TCP segment, nor are they included in length.