COMPUTER NETWORKS — SHORT NOTES:

IP Address = 32bits
Total no. of IP Addresses = 2^{32} = 4G IP addresses.

USEFUL ADDRESSING:

CLASS A: NID BITS : 8-bits → 1 reserved (0)
(0-127)
∴ Total IP addresses = 2^{31}
∴ Total no. of Networks = $2^7 - 2$ excluded { 00000000 (0.x.x.x), 01111111 (127.x.x.x) }
= 126.
∴ Total no. of Hosts/N/w = $2^{24} - 2$. (NID and DBA)

(128-191) **CLASS B**: (10) reserved, #IPs = 2^{30}, #N/w = 2^{14}, #Hosts = $2^{16} - 2$.
(192-223) **CLASS C**: (110) reserved, #IPs = 2^{29}, #N/w = 2^{21}, #Hosts = $2^8 - 2$ = 254.
(224-239) **CLASS D**: 1110 - reserved, No NID/HID, used for multicasting.
(240+) **CLASS E**: 1111 - reserved, —"— , used for R&D.

- COMMUNICATION : (A) UNICAST (1:1) : SIP: 74.3.4.1
  DIP: 128.0.1.2
  (B) Multicast (1:many) : uses CLASS-D IPs.
  (C) BROADCAST : (1-All):
  - Limited (LBA) : Same network.
    DIP: 255.255.255.255.
  - Direct (DBA) : different Network.
    DBA: NID bits remains same but all 1's in HID part.

SUBNETTING:
↳ Borrowing bits from HID to generate SID.

- For n subnets, borrow $\lceil \log_2 n \rceil$ bits.
- For each Subnet, 2 IP addresses are wasted

SUBNET MASK: (1) No. of 1's in S.M indicates NID + SID part.
(2) —"— 0's —"— HID part.

- When no. of subnets (n) with m systems belonging to Class B. Ips given and SM = ? $SM = m \times n \leq 2^{16} - 2$.

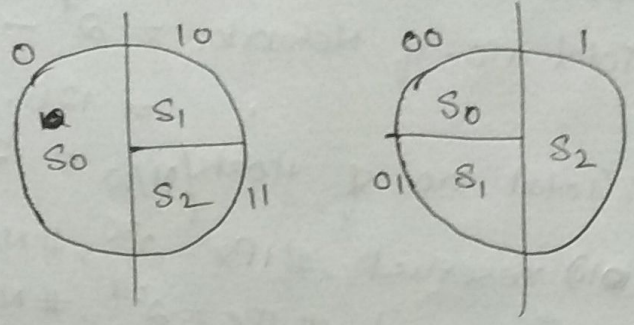
Best Answer: SM = 255.255.255.11100000 (224)
Also, = 255.255.255.00000111 (7)
⋮
many other combinations.

- Speed of fibre cable = 70% of Light Speed $= 0.7 \times 3 \times 10^8$ m/s.
  $= 2.1 \times 10^8$ m/s.

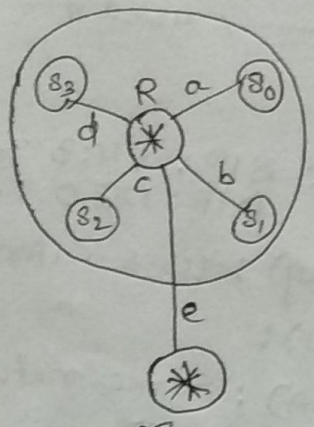
VLSM TECHNOLOGY:

① First reserve 1 bit of HID for SID and try adjusting the larger chunk of systems.

② Then think about adjusting the second largest chunk of systems by reserving 1 more bit and so on...



ROUTER PROCESSING:



NID	SM	I/f (Interface)
—	—	a
—	—	b
—	—	c
—	—	d
0.0.0.0	0.0.0.0	e

- Every Router has a Default Entry.

DIP & SM = NID.
↳ Bitwise AND.

- If DIP & SM matched with more than one NID's Interface, then router forwards to the interface which has longest SM (more no. of 1's in SM).

- SHORTCUT: Start checking from longest SM first (if matched then no need to check further).

* HOW TO CHECK 2 IP's belongs to the Same Network or not?

→ $NID_{AA} = IP_A \& SM_A$ $\quad NID_{BA} = IP_B \& SM_A$

If $\boxed{NID_{AA} == NID_{BA}}$, then A assumes B to be in same N/w.
else, A assumes B to be in different N/w.

→ $NID_{BB} = IP_B \& SM_B$ $\quad NID_{AB} = IP_A \& SB_B$

If $\boxed{NID_{BB} = NID_{AB}}$, then B assumes A to be in same N/w.
else, B assumes A to be in different N/w.

CLASSLESS IP Address: a.b.c.d/n when n = NID bits

Rules for Creating Blocks:

(i) All IP addr. in the block must be contiguous.
(ii) Block Size = 2^n. where n is any Integer ≥ 0.
(iii) FIRST IP % 2^n(B.S) == 0.

E.g: 1101 % 2^1 = 1
1101 % 2^2 = 01
1101 % 2^3 = 101

- Same Approach for Subnetting and VLSM.

SUPERNETTING / AGGREGATION:

- We can only combine same Sized Networks.

Rules for Supernetting:

(i) NIDs must be Contiguous.
(ii) No. of networks = 2^n and Same Sized Networks. (Supernet Size.)
(iii) FIRST NID % 2^n (Supernet Size) == 0.

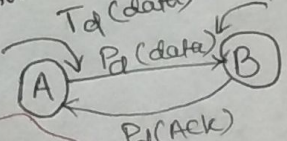
DIP & Sup. Mask = Sup. ID.

FLOW CONTROL: Don't Send packets at a rate the receiver can't receive them.

Rcvr does not send Ack as it is busy.

(1) STOP and WAIT (SNW): Sender sends 1pkt then STOP & WAIT for Ack. [Independent Ack]

Holding Ack Condition (Busy Receiver)



- Total time/ RTT/ Min. waiting Time = Td(data) + 2×Pd + Td(Ack) + Qd + Prd.

$\therefore$ RTT = $T_d + 2 \times P_d$

when piggie-backing is used $T_d(data) = T_d Ack$

Let 'p' = error prob.
Total transmissions = $n\left(\frac{1}{1-p}\right)$

- $W_S = 1$.
- Ack No = Seq. no of Next Packet.
- η = useful work / Total work = $\frac{T_d}{T_d + 2P_d} = \frac{1}{1+2a}$ Efficiency
- Throughput = No. of bits sent/sec = $\frac{L}{RTT} = \eta \times B$ Throughput

SPECIAL CASE: (1) If $\eta \geq 0.5$, then $T_d \geq 2 \times P_d$

- It can only be used for LAN not WAN. ($\eta \propto \frac{1}{P_d}$)
- Good for large Sized packets. ($\eta \propto T_d$)

(1) Lost data Pkt — use TO timer.
(2) Lost Ack — use TO timer + Seq no. + Ack. no.
(3) Delayed Ack — use —"— + —"— + Ack. no.

Some General Concepts!

Half duplex Full duplex

$\eta = \frac{Td}{Td+2Pd}$ $\eta = \frac{Throughput}{(B)}$

- Capacity of a link. = (Pd × B) or (2Pd × B)
  - ★ If Capacity of a link (2×Pd×B) is high, then Pd↑ and B↑ Then SAW protocol is useless.
- Max. W_S = min ((1+2a), 2^K) where 2^K (can be) ≥ 1+2a is min Seq.no. required
  - ∴ Min. no. of bits reqd in Seq. no. field = $\lceil \log_2(1+2a) \rceil$ = K.
- Throughput is also known as Effective Bandwith / Bandwidth utilization.
  - In CN, no. of bits sent/sec.
  - In BOS, no. of processes executed/sec.
  - In COA, no. of instructions executed/sec.

- $(1+p+p^2+p^3+\ldots+\infty) = \frac{1}{1-p} = \frac{1}{1-r}$
- $(1+p+p^2+p^3+\ldots+p^{n-1}) = S_n = \frac{a(r^n-1)}{(r-1)}$ ⇒ $\frac{p^n-1}{p-1}$ in GP.

---

② Go-BACK-N (GBN): If packet is lost, then GO-BACK-N & retransmit. [Commutative Ack] / [Independent]

$W_R = 1$, $W_S = N$.

- $W_S = N$ where N>1, if N==1 then GBN → SNW.
- Ack. Timer < TO Timer.
- Used for Link to link Conn. — Datalink layer.
- $\eta = \frac{N}{1+2a} = \frac{N \times Td}{Td+2 \times Pd}$
- If all Acks are lost in GBN, then to detect duplicate packet we need N+1 Seq. no.
  - ★ Available Seq. no (ASN) ≥ $W_S + W_R$ ≥ N+1

Selective Repeat

③ S-R Protocol: When a packet is lost retransmit only the lost packet. (Independent Acks).

- Available Sequence no (ASN) ≥ $W_S + W_R$ ≥ 2N
- Needs more Seq.no. than GBN.
- ☺ If packets are corrupted, then SNW ignores, but S-R protocol sends NACK, thus packets can be retransmitted even before TOT.
- $W_R = W_S = N$
- Ack Timer is not required.

Routing Algorithms: → Static (Not Generally used, as a bigger picture of Internet is not known).
→ Dynamic → DVR → RIP (uses UDP)
→ LSR → OSPF (uses neither TCP nor UDP, It sends directly via IP).

• Distance Vector Routing
① Developed in 1980's
② Less Bandwith is required
③ Local neighbour knowledge
④ Traffic is low
⑤ Bellman-Ford Algorithm.
⑥ Count to Infinity problem.
⑦ Persistent/Permanent loops.
⑧ Uses RIP
⑨ Convergence is very slow.

Link State Routing.
① developed in 1990's
② More Bandwith is reqd.
③ Global knowledge.
④ Traffic is high.
⑤ Dijkstra's Algorithm.
⑥ No problem of Count to Infinity.
⑦ May have transient/temporary loops
⑧ Uses OSPF.
⑨ Convergence is fast.

RT(A)

A	0 A
B	2 A
C	2 B
D	∞ -

STEP 1: Prepare Routing table at every router based on local knowledge.
STEP 2: Share distance vectors to immediate neighbours.
STEP 3: Find new routing table for each node based on the obtained distance vectors from neighbours.

STEP 1: Prepare Link state packets based on local knowledge.
E.g

A	
B	2
D	1

B	
A	2
C	3
D	7

. . . .

STEP 2: Every Router floods the Link state packet to every other router.

NOTE: For 'n' nodes, DVR calculates Routings Completely in ≤ (n-1) steps.

☹ of DVR: ① If a node gets connected, dist. vectors are updated easily. But ② If a node gets disconnected, then other nodes faces Count to infinity problem.

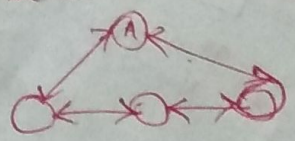
RIP used by DVR
• Routing Metric → hopcount (lowest)
• Max. Hop Count ← 15 (16 is considered as Unreachable)
• It uses UDP to send distance vectors.

OSPF used by LSR.
• It uses IP (not TCP/UDP) to send link state packets.

☺ Split HORIZON METHOD: → ☹ Just Remember- It is not a good solution because sometimes even reachable nodes may become unreachable

→ A node should check the next hop and should depend on next hop for updation of distance vectors.
• Count to Infinity — ✓ solved.
• Looping — ✓ solved
• Convergence is very slow — ✗ Not solved yet.

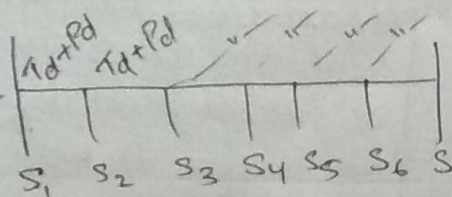
E
B∞
D
B 2 C
B ∞ -



ACCESS CONTROL:

① TIME DIVISON MULTIPLEXING (TDM):

→ ① Divide time into slots and assign each slot to a station.



$\eta = \frac{T_d}{T_d+P_d} = \frac{1}{1+a}$

Time slot = $T_d + P_d$ (time taken to finish transferring the last bit.

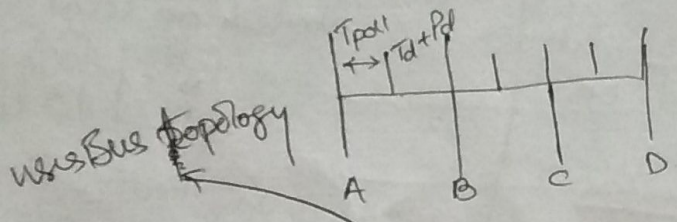
☹ Time slot goes wasted when a station does not uses it.

☺ At any time only one channel can use the link.

② POLLING:

→ No reservation of time slot. ☺

→ Polling time (the time taken to check the carrier) is a waste ☹



$\therefore \eta = \frac{T_d}{T_d + P_d + T_{poll}}$

③ CSMA/CD: (Ethernet) (IEEE 802.3)

CS → Sense the carrier before sending Data.

MA → many channels tries to access the same link.

- Any station can transfer data at any time.
- There are no Acks in Ethernet so sender needs to detect collision before transmission finishes.

∴ Min. Size of packet to detect collision $L \geq (2 \times P_d \times B)$ bits (Term Signed)

NOTE
- At Router, Packet processing time = $T_d/2$;
  ⇓ (time stamp)
  0 to $(T_d/2 - 1)$

Here, collision may happen any time,

∴ Contention/Collision time = $C \times 2 \times P_d$ (C ← no. of collision/contention; $2P_d$ ← time taken to detect each collision.)

∴ Total time (taken to finish transferring the last bit) = Collision time + $T_d + P_d$
= $T_d + (1+2C)P_d$.

Let Total Stations = N
& every station wants to transfer with prob 'p'

$\therefore P_{succ} = {}^{N}C_1 \, p^1 (1-p)^{N-1}$

(Generally) At $p = \frac{1}{N}$, $P_{succ(max)} = \left(1 - \frac{1}{N}\right)^{N-1}$

(Max success when N not given) $\lim_{N \to \infty} P_{succ(max)} = \frac{1}{e}$

∴ No. of times, we should try before getting success is 'e'. (1st)

$$\eta = \frac{T_d}{T_d + (1+2C)P_d} = \frac{1}{1 + 6.44a}$$

When C = e, (when C is not given in exam).

Throughput = $P_{succ} = \eta \times B$.

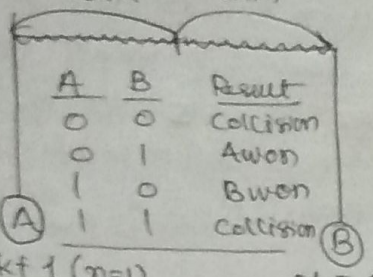
EXPONENTIAL BACKOFF ALGO for CSMA/CD:

- After collision both stations must wait before they start transmitting

waiting time = $K * 2 * P_d$ = $K * RTT$ = $K *$ Slot duration.

where $K = 0$ to 2^{n-1} where n = collision no. w.r.t data packets
no. of times of occurrence
q

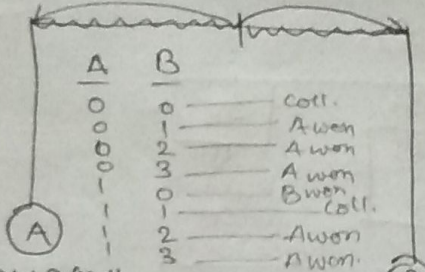
CASE 1: Both A & B tries,



Pkt 1 ($n=1$)
$k = 0$ to 2^{n-1}
$= 0$ to 1
$P(A) = \frac{1}{4} = P(B)$

Pkt 1.
$k = 0$ to 1.
$P(\text{coll}) = \frac{2}{4}$

CASE 2: A succeeds in Case 1.



Pkt 2 ($n=1$)
$K = 0$ to 1
$P(A) = \frac{5}{8}$ $P(B) = \frac{1}{8}$

Pkt 1 ($n=2$)
$K = 0$ to 3
$P(\text{coll}) = \frac{2}{8}$

☺ Prob. of collision decreases

☹ One of them will end up capturing the link.

This effect is known as Capture effect.

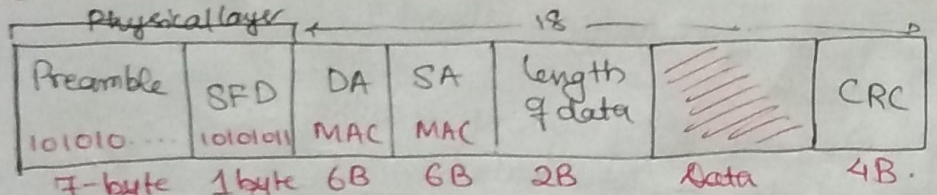
Solution to Expo. Back-off Algo: (with PURGING)

↳ If WT of $X = 0$ [i.e, $0 *$ Slot duration], then X has to wait atleast 1 prop. delay.

MANCHESTER ENCODING: Encoding techniques used in Ethernet.
(has a transition at the middle of each bit.)

Baud rate = 2 × bit rate. but generally **bit rate = Baud rate.**

FRAME FORMAT OF ETHERNET:



used for Synchronization.

unicast Source MAC Addr.

Min: 46B
MAX: 1500B.

Total HS = 22B.
Tailer Size = 4B.

DLL = 26B
NL = 20B
min at DLL ← 46B

DA: → unicast: AB:CD:DE:FA:10:20
1010 1010
unicast (segment)

→ Multicast: Last Bit of 1st Byte is 1.
10101011
AB:....

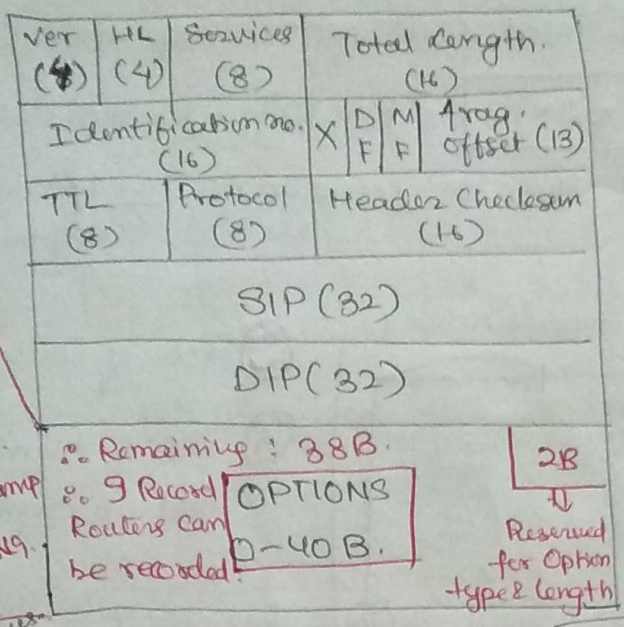
→ broadcast: FF:FF:FF:FF:FF:FF

NOTE

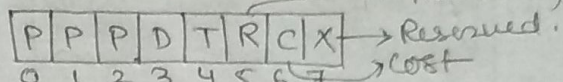
AL → Message
TL → Segment
NL(IP) → Datagram
DLL → frame.

} Packets

IPv4 Header:



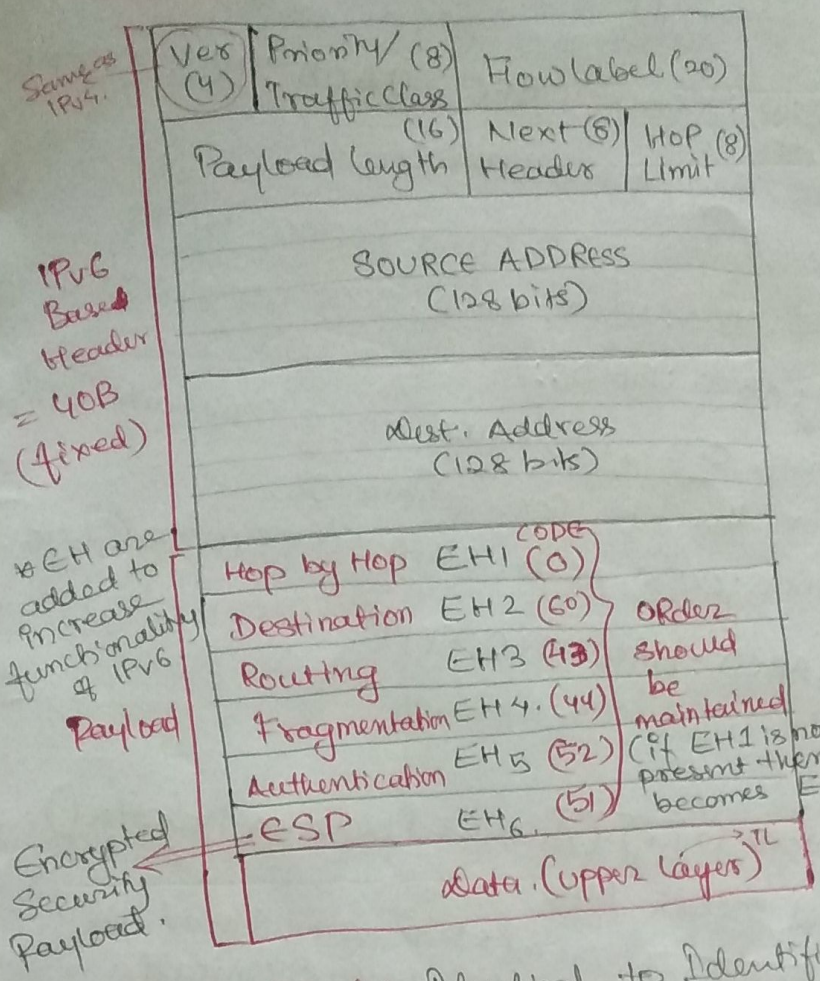
- Ver → (4 bits)
  - → It is used to Indicate IPV4 or IPV6. (0100) or (0110).
- HL → (4 bits)
  Header size = 20B to 60B.
  ∴ HL = 5 to 15 (0101, 1111)
  With, Scaling factor = 4 (no. of bits in HL)
  CASE: If HS = 30B then add 2B in padding and make HL = 8 (0100).
- Services: (8 bits)



  - In any datagram, only one out of 4 services is provided at a time.
- ID no. → (16 bits) used to identify all fragments of the same datagram.
- Total Length = Data + Header.
- Flags: (3 bits) X — Reserved.
  DF → 1: Cannot be fragmented
  → 0: Can be ——"——.
  MF → 1: more fragments are there.
  MF → 0: last/only fragment.
- Time to Live: (8-bits)
  → Standard TTL = 128.
  → used to avoid infinite looping.
  → TTL value = # Hops b/w Sender & Receiver
- Fragment OFFSET: (13-bits)
  → indicates no. of data byte ahead of the fragment.
  → frag. offset of 1st fragment is always 0.
  → Scaling factor = $\frac{\text{Total Data } (\approx 2^{16})}{\text{max. frag. off. val } (\approx 2^{13})} = 8.$
  (*) → ∴ Frag. offset = (No. of databytes ahead of the fragment + padding (if required)) / 8.
- Protocol: (8-bits) IANA has assigned some numbers to each protocol. (IGMP < ICMP < UDP < TCP) → Priority to be kept at router.
  Protocol: (2) (1) (17) (6)
- Header Checksum: (16)
  → IP Header Checksum is only calculated for Header and is changed at every Router (as some fields are updated at every router like TTL, etc.).
  → In TCP Header, nothing changes till the end of transmission
  ∴ TCP Checksum = TCP Header + TCP Data + Pseudo-H

To find delay at every router.

IPv6 HEADER:



- **Payload length:** (16-bits) (Data excluding Header) at NL.
  Max Data = $2^{16} - 1$ = 65535 B. (or ≈ 64KB.)
  - Two fields of IPv4, HL & TL not present here.

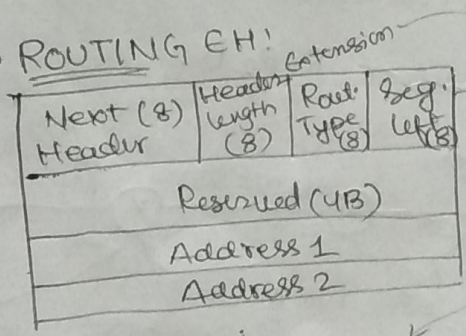
  Payload length = EHs + Data. (upper layer)

- **Hop Limit:** A packet is discarded if Hop limit gets decremented to 0.
- **Traffic Class:** (8 bits) (a.k.a Priority Class).
  → distinguishes different payload on the basis of delivery requirements.
  → First 6 bits for Services (in IPv4) and 2 bits for Explicit Congestion Notification (ECN).

- **Flow Label:** Identical to Identification No. in IPv4. (20bits) → To a router FLOW is a sequence of packets that share the same characteristics.

- **Next Header** (8 bit): It defines the type of 1st EH. header (if present) or the type of data that follows the base header in the datagram.

- **ROUTING EH:**

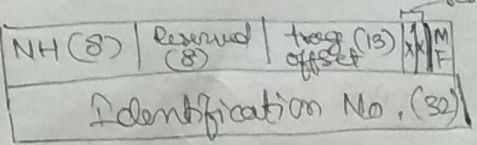


If Header Extension length = n,
Then, total Addresses present = $\frac{n}{2}$.
& Total Routing EH Size = (8 + n8) Bytes
(Header Size)

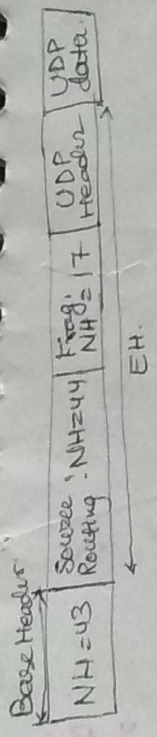
- **Segment Left:** No. of intermediate nodes that has to be visited before visiting destination.

- **FRAGMENTATION EH:**

NOTE: In IPv4, fragmentation was done by source or/and Routers depending upon MTU.
But In IPv6, fragmentation can only be done by source as the source (here) uses a path MTU discovery Technique to find the least MTU among all networks in the Internet.



8-Bytes Frag. EH.



CODES	NH
0	H-H
1	IGMP
2	ICMP
6	TCP
17	UDP

CODES	NH
43	Src. Routing EH
44	Frag. EH
51	ESP
52	Authentication
59	No Header
60	Dest EH

IPV6 Addressing: → Does not Support Broadcasting.

① Remove leading Zeroes

E.g: AA00:000A:0000:00B0:0BAC:0A0A:0000:FFFF.

→ AA00:A:0:B0:BAC:A0A:0:FFFF.

② Check the longest Consequitive 0's and Replace by ::

E.g:- AA00:0000:0000:ABCD:BACD:0000:0000:0000

→ AA00:0:0:ABCD:BACD::

If no longest Consq. 0's, i.e, equal no. of consequitive 0's, then put :: for the left Consequitive 0's.

E.g: AA00:0000:0000:ABCD:BACD:0000:0000:FFFF.

→ AA00::ABCD:BACD:0:0:FFFF.

Some MISC. CONCEPTS:

- **ARP Protocol**: (at Network layer) (Address Resolution protocol).

→ Has IP-Address and needs MAC.

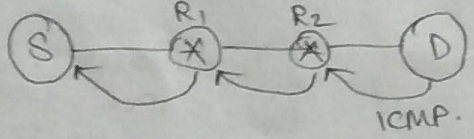
IP_B	MAC_B = ?	MAC_A	FF:FF:... FF MAC BR. Add

- ARP request: Broadcast.
- ARP reply: Unicast.

- ARP can be used between Host-Host, Host-Router, Router-Router & Router to HOST.

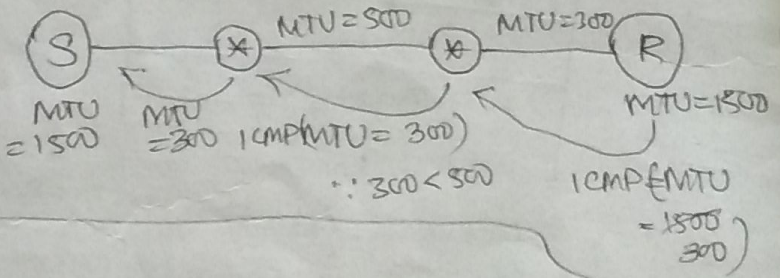
APPLICATION OF ICMP: → Generated only for the 1st Segment.

① To trace Route: Send ICMP Pkt from Dest to SRC saying destination port unreachable.



∴ Route = $R_1 R_2 D$.

② PATH MTU Discovery:



∴ MTU = 300B.

Some Concepts:

① **Segmentation**: It is done at **TL**, to meet the requirements of Lower layers. (NL)

② **Fragmentation**: It is done at **NL**, to meet the — u — of MTU. (Maximum Transmission Unit) in Bytes.

TCP HEADER:

Source Port (16)	Destination Port (16)
Sequence Number (32)	
Acknowledgement No. (32)	
HL (4) / Reserved (6) / URG ACK PSH RST SYN FIN	Window Size (16)
Checksum (16)	Urgent Pointer (16)
OPTION (0 to 40B).	

- **Source & Dest. Port (16 bits each)**
  - → Ports are used to identify process in a host.
  - → 0-$2^{16}-1$ ports
  - → 0-1023 well known port.

  SMTP = 25 FTP < 20, 21
  HTTP = 80 DNS-53.

- **HL (4 bits):**
  Same as IPv4 header
  [5 to 15 HL → 20 to 60 HS]

- **Sequence Number (32-bits)**
  - Seq.no. of the 1st byte of the current packet.

- **Ack. No:** Seq. no. of the next Expected Byte.

- **Flags:**
  1. URG — 0 – no urgent data. 1 – urgent data pointed by urgent pointer.
  2. ACK — 0 – no Ack. 1 – Ack Segment
  3. SYN — 0 – 1 – send req. for sync.
  4. PSH → To push data Bytewise. E.g. – live telecast.
  5. RST → Reset / Refresh the TCP Connection.
  6. FIN → Terminate the Connection.

- **Window Size (16):**
  → The maximum Sender window size.

- **URGENT POINTER**
  → It indicates Last urgent byte

  E.g.
  Seq. no of data = 1000
  Urgent pointer = 100
  ∴ Last urgent Byte = 1100.
  ∴ Total urgent data = 1100 − 100 + 1 = 101 B.

TL
↓
ensures order of packet delivery

DLL
↓
recovery of lost packet between two directly connected node

- **TCP CHECKSUM:**

  TCP CHECKSUM = TCPDATA + TCPHEADER + Pseudo Header (12B)

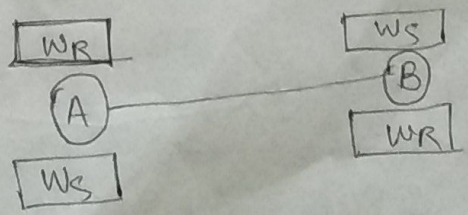
  - The Pseudo-Header is constructed at TL on both Sender & Receiver side, the checksum is calculated & the pseudo header is deleted.

SIP (32)	4B
DIP (32)	4B
Reserved (8) / Protocol (8) / TCP Segment length (16)	4B.

- Seq. no & Ack no are also used for Authentication.

NOTE: Each TCP Connection Has 4 windows,

- TCP always uses Full Duplex Connection.



WRAP AROUND TIME: → depends upon Bandwidth.

↳ Wrap Around is needed because sometimes sequence number gets exhausted within lifetime.

→ 2^{32} sequence number → in WAT s.

Suppose B = 1MB/s.

∴ 1MB in 1s. ⇒ 10^6 B in 1 sec,

⇒ 10^6 seqno. in 1 sec.

∴ 2^{32} seqno = $\frac{1}{10^6} \times 2^{32}$ sec. = WAT.

→ To wrap Around the sequence no, we use some extra bits from Time stamp.

Consider: WAT = 4s and LT = 180s. and B = 1GBps.

∴ In 1 sec = 1GB. = 10^9 seq no.

∴ In 180 sec = $10^9 \times 180$ seq. no ≈ 2^{38} seq. no.

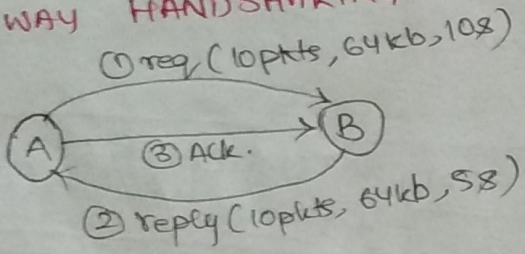
∴ Extra bits required to WA = 38 - 32 = 6.

∴ TIMESTAMP = 6 bits = 0 to 63.

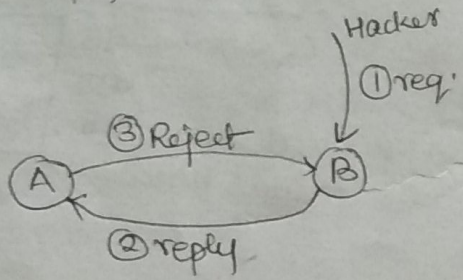
The receiver assumes two packets of the same sequence no. to be different because of different timestamps.

CONNECTION ESTABLISHMENT:

- THREE WAY HANDSHAKING:

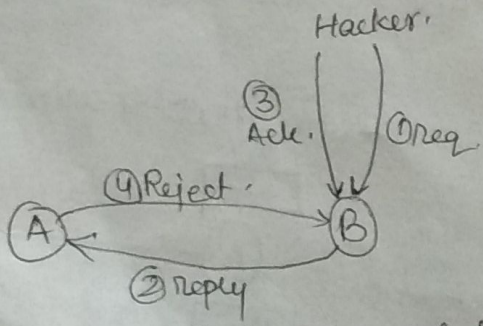


CASE 1:



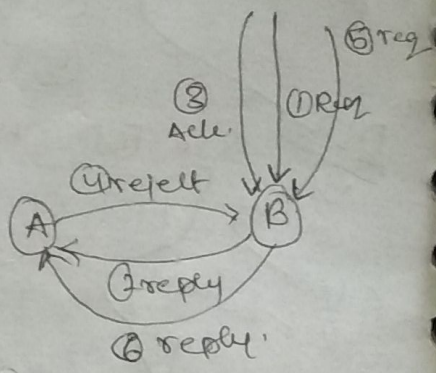
∴ No Issue.

CASE 2:



∴ Receiver takes ④ Reject as a data.

CASE 3:



∴ After some time the capacity of B is full and B cannot take any more user. So, if a genuine user requests it denies.

This is called denial of Service.

☺ use Seqno. & Ackno

used for Synchronization.

① Seq.no=100, SYN=1
MSS=1460, |W|=14600 → Sender window size.

② Reply.
SYN+Ack.
Seq.no=200, SYN=1
MSS=500, |W|=10000
ACK=1, Ack no=101

③ Ack.
Seq.no=101
Ack flag=1
Ack no=201.

- SYN / FIN / SYN+ACK / DATA / DATA+ACK all consumes seq. no.
- Only Ack does not consume a seq. no.

Connection Establishment.

C → S
Seq.NO=1001
Ack=1, Ack no=200 — Data.

S → C
Seq.no=2001
Ack=1 Ack no=1101 — Data+Ack → Piggie Backing

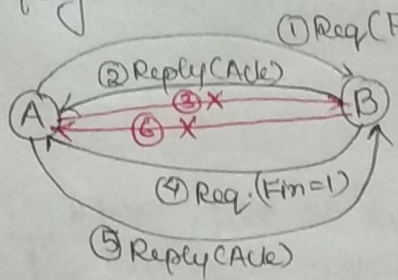
C → S
Seq.no=1101
Ack=1, Ack no=2001 — Pure Ack.

C → S
Seq.no=1101
Ack=1, Ack no=2001 — Data.

Data Transfer.

CONNECTION TERMINATION:

→ To Free up the reserved resources.
→ FIN flag is used here.



← 4 way Handshaking.

PHASE A

Seq.no=1201
FIN=1, Ack=1
Ack=2201

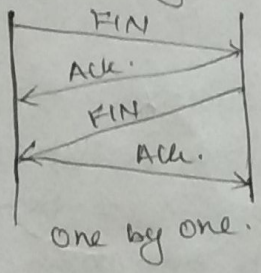
Seq.no=2201
Ack=1, Ack=1202

PHASE B

Seq.no=2201
Ack=1, Ack no=1202
FIN=1

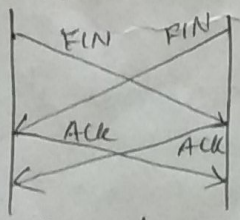
Seq.no=1202
Ack=1, Ack no=2202

→ three general ways:



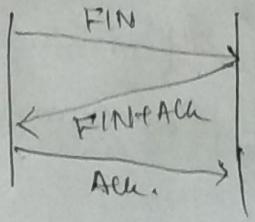
One by one.

OR.



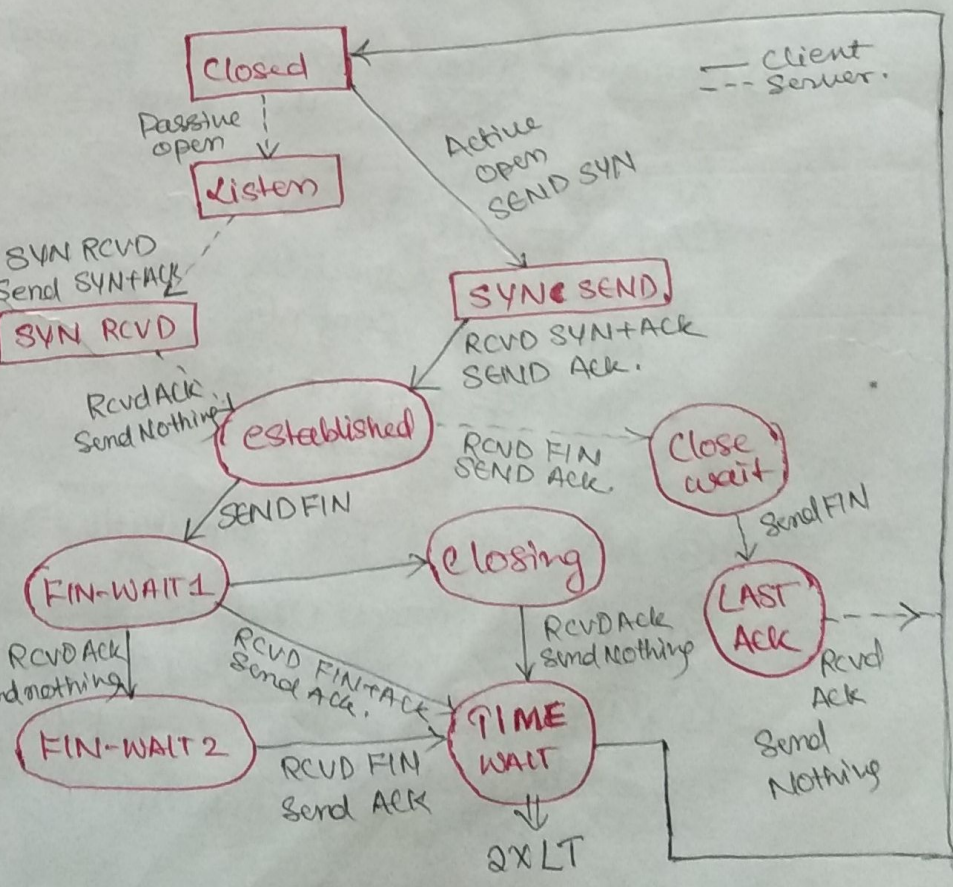
Both at once

OR



FIN+Ack together.

TCP STATE TRANSITION DIAGRAM.



(Domain Name System)

TCP vs UDP

TCP	UDP
① Connection Oriented	① Connection less.
② Reliable	② Not Reliable.
③ Seq. no & Ack no. of pkts.	③ No. Seq. no & no Ack no.
④ Overhead is high, thus slower.	④ Overhead is low, thus faster.
⑤ Used by HTTP, FTP, SMTP, POP3, etc.	⑤ used by DNS, SNMP, NFS (Network file system), TFTP (Trivial FTP), etc...

USER DATAGRAM PROTOCOL (UDP):

4B	Source Port (16)	Destination Port (16)
4B	Total Length (16)	Checksum (16) (=0, to disable error checking)

UDP HEADER SIZE = 8B. (Overhead)

- Message oriented Connection less protocol with less overhead than TCP.
- used for Apps that require 1 req. & 1 reply.
- used for multicasting & broadcasting.
- For constant dataflow.
- For Fastness over reliability.
- UDP is used in SNMP, RIP, etc...

TCP TIMER MANAGEMENT:

① Keep Alive Timer: It is used to keep track of ideal connection. A receiver closes the connection, if sender does not send any data for some time.

E.g: NETBANKING, IRCTC, etc.

② Persistent Timer: When the receiver advertises its window size to be zero, the sender starts a Persistent timer. At the end of PT, the sender sends a 1-Byte Probe Segment and the receiver sends its Advertising window. If receiver Advertising window is still 0, then start PT again and continue.

- The probe segment has a sequence no. but its seq. no is not concerned.

③ ACK Timer: For commutative ACKs.

④ Time Out Timer: Used if data packet is lost.

⑤ Time-wait Timer: Used to wait for finishing the transfer of the last byte before completely closing the connection.

TIME-OUT TIMER ADJUSTMENT:

① Basic Algorithm: $NRTT = \alpha(IRTT) + (1-\alpha)ARTT$

$TOT = 2 \times NRTT$.

② JACOBSON'S ALGORITHM: $AD = |IRTT - ARTT|$ = Actual deviation.

$ND = \alpha(ND) + (1-\alpha)AD$. $NRTT = \alpha(IRTT) + (1-\alpha)ARTT$

$TOT = 4 \times ID + RTT$

③ KAR'S MODIFICATION:

(i) Do not consider the RTT of a retransmitted segment in the calculation.

(ii) Do not update the value of RTT, until you send a segment & receive an ack without any need for retransmission.

(iii) If a retransmission occurs, the value of TOT is doubled for each retransmission.

CONGESTION CONTROL: Occurs when the routers are filled up and they cannot take any more packets.

- W_C = Congestion window = Size/capacity of the Network.
- No. of Segments $= \frac{W_S}{MSS}$ · $TH = \frac{W_S}{2}$ $\in \min(W_C, W_R)$ · $NTH = \frac{TH}{2}$
- SLOW START: 1MSS, 2MSS, 4MSS, 8MSS, ... TH. (1RTT, 2RTT)
- Congestion Avoidance: TH, TH+1, TH+2, ..., W_S. (1RTT)
- CONGESTION DETECTION: Congestion is detected in two ways.

(i) Three duplicate Acks: It indicates Mild Congestion.
Step 1: $NTH = \frac{W_S}{2}$. and congestion Avoidance.

(ii) Time Out Timer: It indicates severe Congestion.
$NTH = \frac{W_S}{2}$ & Slow Start.

TRAFFIC SHAPING:

Another method of congestion control, i.e, shape the traffic before it enters the network.

Leaky Bucket:

- Shape the Bursty traffic into a fixed Rate traffic by averaging the rate.
- Here, Resources/data cannot be allocated & kept for future transmission.
- Outgoing Size is fixed.

Token Bucket: Credits an Idle host.

- Tokens enter into the bucket (at constant interval) at a rate = 'r' tokens/sec
- Bucket has a min. capacity = 'c'
- Max. #packets that can enter a n/w during time 't', i.e, Max. Avg. Rate for Token Bucket $\Rightarrow M = \frac{c + rt}{t}$.

or $t = \frac{c}{M - r}$.

- If no token in bucket, then packet cannot be sent.

CONCEPT OF SWITCHING:

→ The process of forwarding a packet from one port to another port is called as switching.

(1) CIRCUIT SWITCHING: (Generally for Telephone lines)

- Works at physical layer.
- Dedicated physical path or connection is established and maintained for entire duration of communication. The link/path is released only when the transmission ends.
- Here, a series of switches is connected by physical links.

(Better for large sized pkts.)

$$\text{Total Time} = \text{Setup Time} + T_d + X \cdot P_d + \text{Tear-down Time}.$$

Note: Here, $P_d = \frac{\text{length of only 1 hop}}{\text{velocity}}$, X = no. of hops.

(2) PACKET SWITCHING:

- Works at Network layer.
- For fast transmission and minimize T_d, the data is broken into small packets.
- No pre-setup or reservation is needed.

(Better for small sized pkts)

$$\text{Total Time} = X \cdot T_d + X \cdot P_d + (X-1)(Q_d + Pr_d)$$

APPLICATION LAYER PROTOCOLS:

1. DNS (Domain Name ~~Service~~ System) → uses UDP → Port = 53
2. SMTP (Simple Mail Transfer Protocol) → TCP at Port: 25
   → Two components: User Agent (UA) & Mail Transfer Agent (MTA)
   - TEXT BASED Protocol.
   - Create a message & put it in envelop
   - Transfer the envelop
   - Sends images with the help of MIME.
   - To send a mail, the system must have client MTA and to receive the mail, system must have a server MTA.
   - POP3 is a standard protocol to pull the mail from mail-server.

③ FTP (File Transfer Protocol): uses TCP on Port 20 (Data Connection)
Port 21 (Control Connection)
↳ It is an Out-of Band protocol as it uses two ports for data & control.
→ It is a stateful protocol.

④ HTTP (Hypertext Transfer protocol):
→ used mainly to access data on world wide web.
→ Client - server protocol using TCP on Port 80.
→ InBand & stateless protocol.

ⓐ Non-persistent HTTP: (1.0):
→ Clients opens TCP connec. & sends the request
→ Server sends the response & closes the TCP Connection.

for 'n' pictures, 'n' times TCP connec must be opened & closed.

ⓑ Persistent HTTP (1.1):
→ Clients opens TCP connection & sends the request.
→ server closes the connection only after the request of Client or TOT has reached.

COMMANDS:

SMTP: HELO, EHLO, RCPT.TO, TURN, ATRN, SIZE, ETRN, PIPELINING, DATA, QUIT, RSET, VRFY, HELP, MAIL FROM.

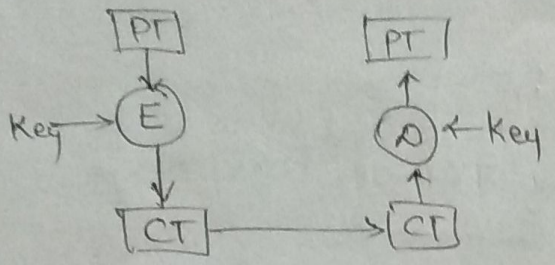
FTP: USER, PASS, ACCT, CWD, REIN, QUIT, PORT, PASV, TYPE, MODE, PROMPT.

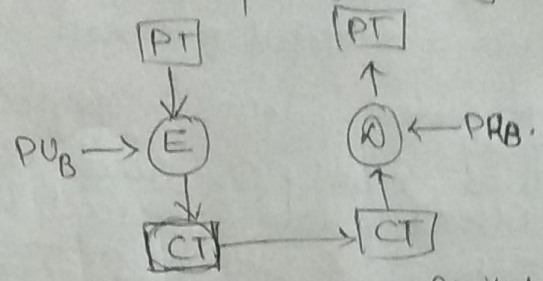
HTTP: HEAD, GET, POST, PUT, TRACE, DELETE, CONNECT, OPTION.

APPLICATION LAYER PROTOCOLS	TL Protocols	Port No.
DNS	UDP	53.
SNMP	UDP	161
TFTP	UDP	69
(Out of Band) DHCP	UDP	67, 68 → Server Port, Client Port.
TELNET	TCP	23
(Out of Band) FTP (Stateful)	TCP	20, 21 → Data Conn, Control Conn.
HTTP (Stateless)	TCP	80
SMTP	TCP	25
POP3	TCP	110

NETWORK SECURITY:

① Symmetric key Cryptography ② Private key Cryptography.

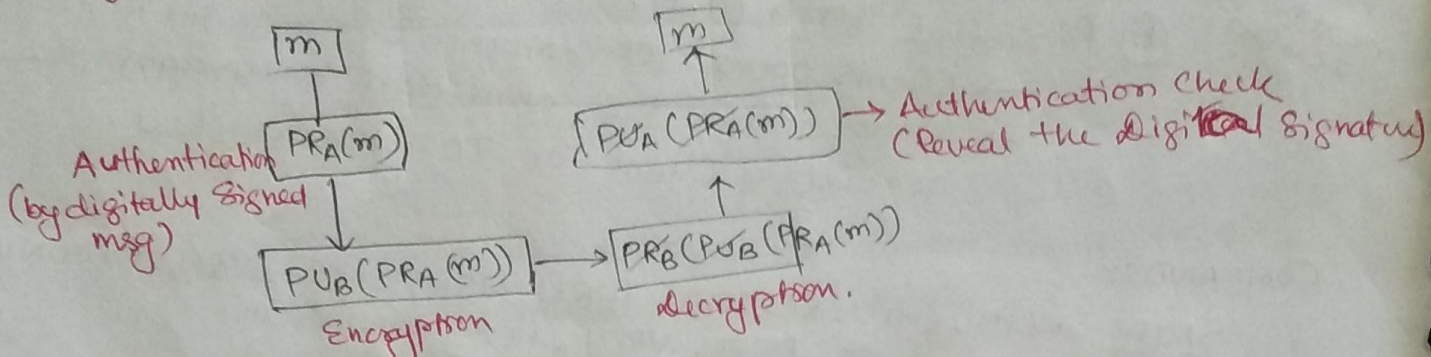




Eg: DES and AES.

E.g: Deffie Hellman, Digital Signature Standard, RSA, etc..

- Digital Signature Standard: used for private key Authentication.



- EULER TOTIENT FUNCTION:
  - → It counts 'tve' number upto 'n' that are relatively prime to 'n'.
  - → $\phi(p \times q) = \phi(p) * \phi(q)$ only iff p & q are relatively prime or GCD (p, q) = 1.
  - → If n is a prime no, then $\phi(n) = n-1$.
  - → If a and n are relatively prime, then

$$a^m \equiv 1 \bmod n$$

  - → If Order/period of $(a^m \equiv 1 \bmod n) = \phi(n)$, then a is called primitive root of n.

E.g: $3^m \equiv 1 \bmod 7$ ∴ 3 is a primitive root of 7.

RSA Algorithm:

(distinct prime nos.)

① Compute $n = p \times q$.

② $\phi(n) = (p-1)(q-1)$

③ Choose e such that GCD $(e, \phi(n)) = 1$

④ $d = e^{-1} \bmod \phi(n)$ or $ed \bmod \phi(n) = 1$

Here (e, n) → Public key

(d, n) → Private key.

⑤ $C = M^e \bmod n$.

$M = C^d \bmod n$.

DEFFIE HELLMAN: (Key Exchange Algorithm).

- ~~Prime no~~ & α is a primitive root of q.
  $q \rightarrow$ prime no

① $Y_A = \alpha^{X_A} \bmod q$ → Private key of A.

② $Y_B = \alpha^{X_B} \bmod q$.

Public key of A

③ $K = (Y_B)^{X_A} \bmod q = (Y_A)^{X_B} \bmod q$.

Shared key.

- Choose 'q' in such a way that both 'q' and $(q-1)/2$ are prime nos.

ERROR CONTROL: #Corrupted/affected bits = Data Rate × Noise Duration

- To detect or correct any error we have to add extra bits called redundant bits. These bits are added by the sender and removed by the receiver.

① **Simple Parity:** One extra bit is added to each code.

→ **Even parity:** no. of 1's in each code word must be even.
  ↳ It can detect all single bit error.
  ⇒ All odd no. of errors can also be detected.

E.g: 1010 – Even parity = 0. (valid)
1110 – Even parity = 1. (not valid).

P=0 [11101] ←1-bit error→ [11111] ⇒ P = 1 (detected)

P=0 [11101] ←2-bit error→ [10100] ⇒ P = 0 (Not detected)

P=0 [11101] ←3-bit error→ [00001] ⇒ P = 1 (Error detected)

→ **Odd Parity:** no. of 1's in each code word must be odd.
  ↳ It can detect all single bit errors.
  → All even no. of errors can also be detected.

→ **HAMMING Distance:** HD = SENT DATA (XOR) RCVD DATA.

$HD = S \oplus R$

- Choose the min. among all HD and get min Hamming distance.

NOTE: To detect 'd' bit errors, min HD = 'd+1'

To correct 'd' bit error, min HD = (2d+1).

② 2D-PARITY: It can detect 1, 2 or 3-bit error that occur any-where in the information matrix (IM).

→ One parity bit is associated with each row and each column of the IM.

→ It can only detect some 4 or more bit errors.

Eg: 010010 | 010011 | 100101 | 111001 | 100100

Based on even parity ⟹

0	1	0	0	1	0	0
0	1	0	0	1	1	1
1	0	0	1	0	1	1
1	1	1	0	0	1	0
1	0	0	1	0	0	0
1	1	1	0	0	1	

← Row parity.

Column parity →

Similarly, it can also be based on odd parity.

③ Cyclic Redundancy CHECK (CRC): only when (x+1) is a factor of G(x).

→ It can detect all odd / single bit / burst error of length equal to polynomial degree.

Consider, Given data = 1011011 (m) and CRC = 1101 (n+1).

∴ CRC = x^3+x^2+1 (∵ degree of CRC poly = 3).

useful bits = m (7, here)

total bits = m+r (10, here)

$\eta = \frac{m}{m+r}$.

$= \frac{7}{10} \times 100\%$

$= 70\%$.

① Add 0's to data when #0's = degree of CRC poly.

② Perform Binary division using XOR operation.

③ Keep ignoring the leading zeroes at each step.

④ From the final ~~data~~ remainder, choose n-bits (→ from LSB) = degree of CRC polynomial, and Append it to original data and send it to receiver.

∴ Receiver Data = Actual Data + Remainder (upto n-bits only)

⑤ At receiver side, perform the same XOR operation using the Received data as Dividend and CRC poly. as Divisor.

If remainder becomes 0, then NO error.