Computer Networks

-> In ARPANET, NID = 8 bits, HID = 24 bits.

-> classA: 0-127, 0 for IPV4, 127 for loopback addresses.
(0)
NID: 7 bits, HID: 24 bits.

-> classB: NID: 14 bits, HID: 16 bits., no of ip addr = $2^{16} - 2$
(10)

-> classC: NID: 21 bits, HID: 8 bits, no. of ip addr = $2^8 - 2$
(110)

-> class D: Prefix: 1110, used for multicasting.

-> class E: Prefix: 1111, used for military purposes.

-> LBA: 255.255.255.255, DBA: last address of network.

-> CIDR: -> All IP addresses should be contiguous.
-> size of block should be in power of 2.
-> First IP address, if divided by size of block should give zero as remainder.
(same applied for supernetting)

-> If we want data to be sent through router always, we set SM = 255.255.255.255

-> Private addresses reserved by IANA:
cA: 10.0.0.0 to 10.255.255.255 = 2^{24} > 16M (NASA)
cB: 172.16.0.0 to 172.31.255.255 = $2^{16} \times 16$ (school/college)
cC: 192.168.0.0 to 192.168.255.255 = 2^{16} (popular).

B	L	
10^3	2^{10}	: k
0^6	2^{20}	: M
10^9	2^{30}	: G

$T_t = L/B$, $T_p = \frac{\text{distance}}{\text{velocity}}$

Effective bandwidth (or) Bandwidth utilization (or) Throughput

= no. of bits per second

$= \frac{L}{\text{Total time}}$ //

= efficiency × BW. $= \frac{B}{1+2a}$, $a = T_p/T_t$

S/w + Timeout timer + Sequence no's (Data, Ack) solves all the problems. TO timer $\geq T_t + 2 \times T_p + T_{ack} + T_{proc}$

Sequence no bits = $\lceil \log_2(1+2a) \rceil$, $W_S = \min(1+2a, 2^n)$, n is the no. of pre-defined bits for seq. no's

- GBN: Sender window = N, receiver = 1
  - uses cumulative acknowledgements., seq. no's = $\lceil \log_2(N+1) \rceil$
  - used in DLL, HDLC
- SR: sender window > 1, receiver window = sender window
  - used in TL, TCP, seq. no's = $\lceil \log_2(2N) \rceil$
  - independent & Negative acknowledgements are possible.

$\eta = \frac{1}{1+a}$, $N \times BW$ = Throughput, N is no. of stations

$\eta = \frac{T_t}{T_{poll} + T_t + T_p}$

CSMA/CD : $T_t \geq 2 \times T_p$; when 'p' is same for all stations

→ $P_{max.success} = \left(1 - \frac{1}{n}\right)^{n-1}$, $P_{succ} = nP(1-P)^{n-1}$, $P = \frac{1}{n}$ for max. success.

When 'p' is diff for all stations, $P_{succ} = P_1(1-P_2)(1-P_3)\ldots + P_2(1-P_1)(1-P_3)\ldots + P_3(1-P_1)(1-P_2)\ldots$

$$\eta = \frac{T_t}{T_p + T_t + C \times 2T_p} \quad , \quad C = \left(1 - \frac{1}{N}\right)^{1-N} - 1$$

(or)

$$C = \frac{1}{P} - 1$$

$$= \frac{1}{1 + 6.44a} \quad , a = T_p / T_t .$$

Time = Collision detection time + length of jamming signal

Timeslot (Backoff) = $T_t + T_p + T_p$.

→ B seconds = $\frac{\text{Bit time}}{BW}$, time = $\frac{\text{distance}}{\text{velocity}}$.

→ Token passing: $\eta = \frac{N \times T_t}{T_p + (N \times THT)}$, $THT_{DTR} = T_p + N \times B + T_t$.

By default, $THT_{ETR} = T_t$.

→ Pure aloha: $\eta = G \times e^{-2G}$, $\eta_{max}(G = 1/2) = 18.4\%$.

→ Slotted aloha: $\eta = G \times e^{-G}$, $\eta_{max}(G = 1) = 36.8\%$.

→ Encoding technique: Manchester in ethernet.

→ Manchester: ⊓ 1 ⊔ 0 Differential manchester: 1: ⊓, ⊔ 0: ⊓⊔, ⊔⊓

DLL = LLC : EC, FC
MAC : EC, AC, PA

NL packet

Ethernet frame:

7	1	6	6	2		4
Preamble	SFD	DA	SA	length	DATA	CRC

PL header — DLL header — DLL Tail

→ Min. size of a Dataframe ≥ 64 Bytes.

→ Min size of a IPDU ≥ 64 + 8 = 72 Bytes

→ Min size of the NL packet i.e datagram ≥ 64 − 18 ≥ 46 bytes

→ Max size of a Dataframe = 1518 bytes

→ " " " " NL Packet = 1518 − 18 = 1500 bytes

→ " " " " IPDU = 1518 + 8 = 1526 bytes.

Circuit switching: time = setup time + teardown time + T_t + T_p × (no of hops)
(done at PL).

Packet switching time = (no. of hops) T_t + (no. of hops) (T_p b/w 2 hops).
(done at NL & DLL).

Packetisation: Time = (no. of hops) T_t + (no. of packets − 1) · T_t.

→ Virtual circuit: Connection oriented, cost/time, One header

→ Datagram: Connection less, cost/byte, All have headers

IPV4:

Version (4)	Header length (4)	Type of service (8)	Total length (16)
Identification (16)		0	Do not fragment (1) / More (1) fragment / Fragment offset (13)
Time to Live (8)	Protocol (8)		Header checksum (16)

Scaling factor: (4)

Source (32)

Destination (32)

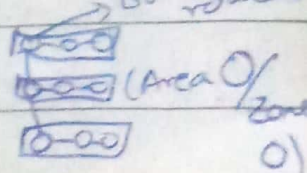
Options (0-40 B) (only 9 IPs are stored, but, not 10).
- → Record routing.
- → Source routing.
- → Padding.

Data

- → Any device with N/ decrements TTL.
- → If wiser segmentation of 1460 B is done, there will be no need of fragmentation at NL
- → MTU is the size of Data in DLL frame.
- → Fragment offset is scaled to 8. (Red/inc is done)
- → If DF=1, fragmentation is done by sender else, it is done by Router.
- → ARP: IP → MAC, RARP (Bootp, DHCP): MAC → IP
- → The problem of DVR is (Persistent probs) solved by split horizon.
- → Link state packet in LSR: (Transient problems)

TTL	
Seq	
B	2
C	3

- → DVR uses (Routing Information Protocol),
- → LSR uses (Open Shortest path first).



TCP:

Source port (16 bit)	Destination Port (16 bit)
Acknowledgement no. (32)	
Sequence no (32)	
Header length (4) | 6 bits reserved | URG PSH SYN RST ACK FIN	Window (or) adv. window (16)
Checksum (16)	Urgent pointer (16)
(adv. window can be increased (14 bits)). Options (0-40) Bytes (Timestamp, MSS, padding)	
Data	

DNS:53, HTTP-80, FTP-21, TELNET-23, SMTP: 25

0-1023: well-known, 1024-49,151: Reserved, 49,152-65,535: Client

-> TCP is a byte-stream protocol.

-> IP is a packet-stream protocol.

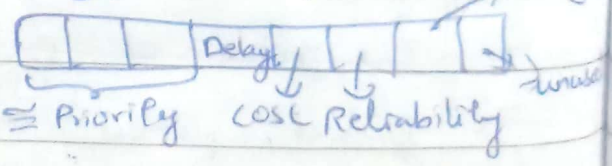
-> HDLC of LLC is bit-stream protocol.

-> $TL_{TCP} = TL_{IP} - HL_{IP}$, length of data in TCP = $TL_{TCP} - HL_{TCP}$

-> Wrap around time of seq. numbers $\geq$ lifetime.

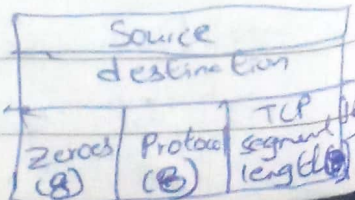
-> No. of bits to increase = $\lceil \log_2(LT \times BW) \rceil - 32$.

-> Type of service in IP:



-> PSH follows in-order, URG doesn't care for order.

-> On segment, Pseudo IP header is added and TCP header is added and on whole of this, checksum is found.



TCP uses SR + GBN (75%) (25%)
→ WS = WR
→ out of order
→ Acks used are cumulative.

TCP uses 3-duplicate ack retransmission (or) coarse grained transmission technique.

$W_{congestion}$ grows exponentially until $\lfloor \frac{W_R}{2} \rfloor$ and then, starts growing linearly till W_R.

If error occurs in slow start phase, we start congestion avoidance from $Th = \lfloor \frac{W_c}{2} \rfloor$.

If error occurs in congestion avoidance phase, we start slow start phase with $Th = \lfloor \frac{W_c}{2} \rfloor$.

Time-wait timer: 2 × LT

Basic alg. for TO ⇒ $NRTT = \alpha(IRTT) + (1-\alpha)ARTT$

$TO = 2 \times RTT$

Jacobson's algorithm for TO ⇒ $NRTT = \alpha(IRTT) + (1-\alpha)ART$

$AD = |ARTT - IRTT|$, $Next\ deviation = \alpha(ID) + (1-\alpha)$

Karn's modification to Basic & Jacobson's: If a packet has been delayed (i.e. > TO), we double the TO_{pre} and wait and continue this procedure until the packet has reached us.

Silly window syndrome is solved by Nagle's algorithm.

-> Token bucket: Avg. rate = $\frac{C+rt}{t}$, if bucket is already full

DP:

Source Port (16)	Destination Port (16)
length (16) header + Data	checksum (16)
Data	

called u protocol

Eg: DNS, BootP, DHCP, Network timeTness TFTP, RIP, OSPF

-> applied on UDPH, UDPdata, Pseudo IP header.

-> length ≤ 100m : MAN
>100m : LAN
≥100km : WAN

LAN segment: LAN wire having no attenuation. diffent wires have diffent ranges.

-> NSlookup <domain name> will fetch the dynamic IP's connected to this based on our server.

-> HTTP (80): stateless, inband, local HTTP cache to HTTP server

-> FTP (21): Teetia, Filezilla, Out of band, statefull.
control - Port.no 21
Data - Port.no 20

-> SMTP (25), POP (110): in-band protocols, uses MIME for conversions

-> HTTP & FTP, SMTP, POP => uses TCP for reliability.

-> DNS uses UDP for reliability, speed.

IPV6:

Version (4)	Priority/Traffic class (8)	Flow label (20)
Payload length (16)	Next header (8)	Hop limit (8)
Source (128)		
destination (128)		
EH1, EH2 ...		
DATA		

Pg

Priority:
0 - No specific data
1 - Background data
2 - Unattended traffic
3 - Reserved
4 - Attended traffic
5 - Reserved
6 - Interactive traffic
7 - Control traffic
(8 (11 a media) to 15 (LA media)).

→ Special privileges for a flow are set by control packets:
1) Resource reservation protocol
2) Real time transport protocol.

Next header:
0 - hop by hop
2 - ICMP
6 - TCP
17 - UDP
43 - Source routing
44 - Fragmentation
50 - Encrypted security payload
51 - Authentication
59 - Null header
60 - Destination options header

Convention Order: 0, 60, 43, 44, 51, 50, 60

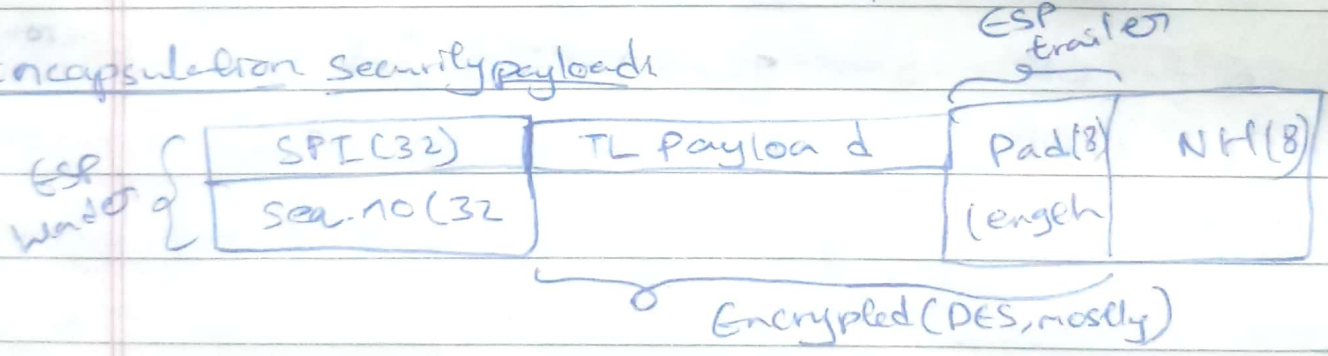
→ Only IPV6H can put NH as 3010 i.e hop-by-hop.

→ Hop-Hop options:

NH (8)	length of header (8)
Options	

→ AH provides message integrity, avoids replay attacks, source authentication

Encapsulation security payload



→ :: (Double colon) in IPV6 should be used only once.

→ Prefixes :

010 : Provider based unicast address

001 : Geographic " " "

1111 1110 10 : Link local address

1111 1110 11 : site " "

1111 1111 : Multicast address

0001, 100 : Reserved.

→ Provider based : | 010 | Registry (5) | Provider ID (n) | Subscriber ID (56-n) | Intra-subscriber 64 bits.

10000 - Multi-regional
01000 - RIPENCC
00100 - APNIC
11000 - INTERNIC
} done by IANA

→ Geography based : | 001 | Global routing Prefix (45) | Subnet ID (16) | Interface (64)

→ Multicasting : | 1111 1111 | Flag (4) | Scope (4) | Group ID (112)

0000 - permanent
0001 - Transient

0000 - Reserved
0001 - Node local
0010 - Link local
0101 - site local
1000 - Organisational
1110 - Global
1111 - Reserved.

Loopback : | 8 bits 0's | 119 bits 0's | 1 |

, IPV4 compt (V6 to V4) | 0's (8) | 0's (88) | IP (32)

(V4 to V6) | 0's (8) | 0's (72) | 1's (16) | IP (32)

(for single N/W)

Link-local : | 1111 1010 | 0's (70) | node address (48) - MAC |

Site-local : | 1111 1011 | 0's (38) | subnet (32) | Node address (48) - MAC |

(For subN/W of a N/W)

→ WIFI : IEEE 802.11

$a \equiv b \bmod n$ means a, b divided by n leaves same remainder and also a/n leaves b as remainder.

$ab \equiv 1 \bmod p$, b, a are multiplicative inverses of each other where (a, p), (b, p) are co-prime pairs.

If (a, n) are co-prime, then, $a^{\phi(n)t} \equiv 1 \bmod n$ - Euler's

(or) $\gcd(a, n) = 1$ $a^{\phi(n)t+1} \equiv a \bmod n$ - fermat's

If n is prime-factorised as $n = p_1^{e_1} \times p_2^{e_2} \times p_3^{e_3} \times \ldots p_n^{e_n}$

then, $\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \times \ldots \left(1 - \frac{1}{p_n}\right)$, $p_1, p_2, p_3 = p_n$

RSA Algorithm: p, q primes are choosen, $n = p \times q$, $\phi(n)$ is found at receiver, and e is choosen s.t $1 \leq e < \phi(n)$. then, (e, n) is publickey of receiver. Now, using this public key, P.T is converted to ciphertext $= P^e \bmod n$. At receiver, $P^{ed} \bmod n$ is found, where $ed = \phi(n).t + 1$. Private key (d, n).

Digital signatures < encrypt (sender's pvt. key). / decrypt (sender's publickey).

'b' is a primitive root of prime 'p', if powers of b include all the residues of mod p.

Diffiehellmann: Sender (or) receiver finds out (a, n) and shares with other.

Sender finds $a^{k_s} \bmod n$ and shares with receiver.

receiver finds $a^{k_r} \bmod n$ and " " sender.

Now, key $= (a^{k_r} \bmod n)^{k_s} \bmod n = (a^{k_s} \bmod n)^{k_r} \bmod n = a^{k_r k_s} \bmod n$

→ In UDP, if port-no is >1023, then it's client.

" " " " " <1023, " " server.