Short Notes

IP Address

1. IP addr : to identify nlw & host
   Port no. : " " process.

2. IP addr : 32 bits

8bit	8bit	8bit	8bit

3. IP addr is divided into nlw ID & host ID.

N/w ID	Host ID

4. Initially organization like ARPANET did static division of IP addr. NID = 8bit, HID = 24bit

5.

	Starts with	Total no of IP's	1st octet range	(bits) NID	(bits) HID	No of N/w	No of hosts/nw	No of IP per Nw
Class A	0	2^{31}	1-126	8	24	126	$2^{24}-2$	2^{24}
B	10	2^{30}	128-191	16	16	2^{14}	$2^{16}-2$	2^{16}
C	110	2^{29}	192-223	24	8	2^{21}	2^8-2 (NID & DBA)	2^8
Multicasting ← D	1110	2^{28}	224-239	Not divided into nlw ID & host ID				
Research work ← E	1111	2^{28}	240-255					

6. Class A two nlw are reserved
   0.0.0.0 → Dynamic host configuration protocol
   127.x.y.z → Loop back address

7. 255.255.255.255 is Limited broad cast address

8. We use class D range 224.0.0.0 to 239.255.255.255 for multicasting.

9.

NID	HID		
–	0's	→	NID
–	1's	→	DBA
1's	1's	→	LBA
1's	0's	→	subnet mask / Network mask

} valid for class A, B & C as class D & E IP addr dont have division into NID & HID

1's 1's ⇒ all isolated group error

* casting → Unicasting, Multicasting, Broadcasting → LBA, DBA

10. Subnetting is borrowing bits from HID.

11. Subnet Mask :
NO. of 1's = NID + subnet ID
NO of 0's = HID

* If we don't want any host to directly send pkt to anyone give subnet mask 255.255.255.255 to all.

∴ Length of subnet mask = no. of 1's in subnet mask.

12. SubnetMask (AND) IP add. = N/w ID

13. ISP provides IP addr, subnet mask, default gateway & DNS. to a host connected over internet.

14. VLSM → variable length subnet masking

15. NO. of host = NO. of IP addr − 2

16. CIDR → classless interdomain routing.

17. CIDR representation: a.b.c.d /n → Block ID/ N/w ID

18. Subnetting HID ↓ BlockID ↑
→ borrowing bits from HID

19. Subnetting : length of subnet mask ↑
Supernetting : ↓
→ borrowing bits from NID

20. At a time we aggregate 2^n n/ws i.e. power of 2.

21. Private IP addr : 1 n/w
1 n/w → 10.0.0.0 to 10.255.255.255
16 n/ws → 172.16.0.0 to 172.31.255.255
256 n/ws → 192.168.0.0 to 192.168.255.255

22. NAT: Network address Translation.
Default gateway does. (when host with private IP wants to communicate with public IP host)

23. Rules of CIDR block:
- IP addr should be continuous
- block size → 2^x
- First IP addr must be divisible by block size.

24. Rules of Aggregation (Supernetting) of CIDR blocks:
- All blocks should be continuous.
- All block size = 2^x & equal.
- 1st block NID should be divisible by block size.

25. Subnetting done for security
Supernetting done to reduce routing table size.

26. 2 host in different private nws may have same private IP address but no two host can have same public IP address.

$E \in [d, n-1]$ ⇒ all isolated double error
can be detected

ERROR control :

1) No. of corrupted bits = data rate * Noise duration

corrupt pkt is discarded

Error control

- Detection
  1. Simple Parity
  2. 2D Parity
  3. Checksum
  4. CRC
  5. Data + Data
- Correction
  1. Hamming code

2) **Simple Parity Check:**

Data + 1 Parity bit

- Even parity → Parity bit added to make no. of 1's even
- Odd parity → Parity bit added to make no. of 1's odd.
- can not detect even bits error.

3) **2D Parity Check:**

- For each row & column 1 parity bit is calculated.
- can not detect some pattern of error.

4) **Cyclic Redundancy Check:**

- CRC bit is added to data so that data is exactly divisible by CRC generator.
- based on XOR / mod 2 sum / binary division.
- If CRC generator is n bits, add (n-1) 0's to data. Perform XOR b/w data & CRC generator till we get (n-1) bits CRC.
- Remainder of division = CRC
- (Data + CRC) is send
- At receiver side perform XOR, if CRC = 0 ⇒ data is correct.
- If CRC generator has more than 1 terms & coefficient of $x^0 = 1$ ⇒ all single bit error can be detected.
- If generator has (x+1) its factor ⇒ all odd numbered error detected.
- If generator does not divide $x^t + 1$ $t \in [2, n-1]$ ⇒ all isolated double error can be detected.

NOTE: CRC is computed by NIC (H/w)
checksum is " " Software

5) **Checksum:**

- If 8 bit checksum is used, data is divided into 8-8 bits grp & added, it's 1's comp. is checksum.
- (Data + checksum) is send.
- If carry comes while adding, add to result.
- Receiver also does same, if result comes 0 then data is correct.

6) Hamming distance: no. of bits different. d(100, 011) = 3
Take XOR & count no. of 1's.

7) Minimum Hamming distance: min of all possible pairs.
If a, b, c data given, find min of d(a,b), d(a,c), d(b,c)

8) To detect 'd' bit error, min distance required = (d+1)

9) To correct 'd' bit error, min distance = (2d+1)

10) Hamming code can correct only 1 bit error

11) If 'n' bit data is to be send, 'p' bit parity is attached with it
$2^p \geq n+p+1$
p is minimum value.

12) Parity bit is placed at locations 2^m, $m \geq 0$

13) Find the data on which parity bit is dependent, from it find parity bit value.

Flow control:

1) Transmission Delay:

$$T_d = \frac{\text{length of pkt}}{\text{Bandwidth (Transfer rate)}} = \frac{L}{B}$$

2) Propagation delay:

$$T_p = \frac{\text{distance}}{\text{velocity}} = \frac{d}{v}$$

3) For Bandwidth use $1K = 10^3$, for data use $1K = 2^{10}$.

4) In optical fibre, velocity = 70% of speed of light = $2.1 * 10^8$ m/s

5) If v not given in ques, use $2.1*10^8$ m/s for wired medium (optical fibre) and $3*10^8$ m/s for wireless.

6) Types of delay:
- Transmission delay (T_t)
- Propagation delay (T_p)
- Queuing delay (T_Q) } consider 0 if not given
- Processing delay (T_{pd}) }

Stop and wait

1) Sender send 1 data pkt & waits for Ack.

2) Efficiency = $\frac{\text{useful time}}{\text{Total time}}$

$$\eta = \frac{T_t}{T_t(\text{data}) + 2T_p + T_t(\text{Ack}) + T_Q + T_{pd}}$$

$$= \frac{T_t}{T_t(\text{data}) + 2T_p}$$ considering T_Q, T_{pd}, $T_t(\text{Ack})$ as 0.

$$= \frac{1}{1+2a}, \quad a = \frac{T_p}{T_t}$$

3) Efficiency → Line utilization → Link Utilization → Sender Utilization.

4) Throughput → Effective Bandwidth → Bandwidth utilization → Max data rate possible → Eff. transfer rate

5) Through put = η * Bandwidth
= no. of bits per unit time.

6) Round trip time, RTT = $2T_p$

7) window size, $W_s = W_R = 1$

8) For $\eta \geq 50\%$

$T_t \geq 2T_p \Rightarrow L \geq 2T_p * B$

9) $\eta \propto \frac{1}{d}$, $\eta \propto L$ Stop & wait is suitable for LAN

10) Stop & wait ARQ (Automatic Repeat Request) = Stop & wait + Time out timer

11) Time out time, $T_o \geq 2T_p$

12) Sequence no. field → 1 bit
Seq no. = 2

13) Ack no. tells about next pkt to be send.

14) If n packets r to be send with error probability 'p' then total pkts send = $n\left(\frac{1}{1-p}\right)$

Sliding window Protocol: Pipelining

1) window size is increased.

GO BACK - N SR

2) capacity of the link
= B x Tp Half duplex
= B x 2Tp Full duplex
(B: Area of cross section; 2Tp: length of pipe)

3) In stop & wait, $\eta \propto \frac{1}{\text{capacity of link}}$

4) In sliding window protocol size of sender window = 1+2a

5) Min no. of seq no. required = 1+2a
↳ size of seq no field = $\lceil \log_2(1+2a) \rceil$

6) If given seq no field is 'n' bits, Sender window size = $\min[1+2a, 2^n]$

GO BACK - N

1) Sender window size = N > 1
Receiver window size = 1

2) $\eta = \frac{\text{Sender window size}}{1+2a}$

3) Out of order pkt rejected.

4) As window size increases, η increases, no. of retransmission increases.

5) GB-N uses Cumulative Ack
To timer ≥ Ack timer + $2T_p$

6) Sequence no. ≥ N+1

7) In general
Seq no ≥ $W_s + W_R$

8) If seq field = k bits in GB-N
No. of seq no = 2^k
$W_s = 2^k - 1$, $W_R = 1$

NOTE: If given Ack is piggybacked then consider T_t of Ack as 0.

Selective Repeat:

1) $W_s = W_R$
2) uses independent Ack.
3) Also accepts out of order pkts
4) uses -ve ack for corrupted pkts.
5) Seq no = $W_S + W_R = 2N$
6) Sender does searching, receiver does sorting.
7) CPU↑ BW↓ : SR (searching & sorting)
   CPU↓ BW↑ : GB-N (more retransmission)
8) wireless → SR preferred (BW↓)
   wired → GB-N " (BW↑)
9) $\eta = \frac{\text{sender window size}}{1+2a}$
10) If seq no. = k bits
    $W_S = W_R = 2^{k-1}$

Medium Access control:

1) Required for broadcast link.

TDM | Polling | CSMA/CD | Aloha

Time division Multiplexing:

i) Size of time slot = $T_t + T_p$
   based on max pkt size ↓ & max T_p i.e host at extreme corner
ii) $\eta = \frac{T_t}{T_t+T_p} = \frac{1}{1+a}$; $a = \frac{T_p}{T_t}$
iii) $T_t = \frac{L_{max}}{B}$, $T_p = \frac{d_{max}}{v}$ common to all
iv) Throughput = η * Bandwidth
v) Max available eff Bandwidth (Throughput) = no. of stations * BW requirement of 1 station.
   ↳ common to all

Polling:

i) $\eta = \frac{T_t}{T_t + T_p + T_{polling}}$

CSMA/CD:

1) Carrier sensing multiple Access / collision detection
2) used in Ethernet. (on ques, consider Ethernet & CSMA/CD same)
3) $T_t \geq$ collision return time
   $T_t \geq 2T_p$
   $L_{min} = 2T_p * B$
4) Monopoly: prblm with bigger pkt size, entire link is overtaken by single host.
5) Station send jam signal as soon it detect collision. → 48 bits
6) Back off algo:
   - If station goes nth collision, it chooses a random no. in range $[0, 2^n - 1]$
   - Back off time = Random no. * Time slot
7) Also known as binary exponential backoff algo. → works with 2 station
   ↳ prob of collision exponentially decreases
8) Capture effect: Every time A wins prob of winning of A increases.
9) $\eta = \frac{T_t}{T_t + T_p + C*2T_p} = \frac{1}{1+6.44a}$, C → avg no. of collision
10) $C = e = 2.71$
11) Prob of successful transmission $= {}^nC_1\, p(1-p)^{n-1}$
    n → no of stations
    p → prob of each station to transmit data
12) $\eta \propto \frac{1}{d}$ (∵ used in wired LAN)
    $\eta \propto L$

Aloha:

1) used in wireless network.

Pure Aloha | Slotted Aloha

2) Pure Aloha:

(i) Any host can send data anytime if collision occurs, it waits for back off time & again retransmit.
(ii) $\eta = G e^{-2G}$ G → no. of stations willing to transmit in T_t
(iii) $\eta_{max} = \frac{1}{2}e^{-1} = 0.184 = 18.4\%$ ($G = \frac{1}{2}$ i.e 1 station transmit in $2T_t$)
(iv) Vulnerable time = $2T_t$

NOTE: In wireless we consider $T_p = 0$

3) Slotted ALOHA:

i) Any station can transmit only at starting of its time slot.

ii) $\eta = GE^{-G}$

iii) $\eta_{max} = e^{-1} = 0.368 = 36.8\%$ ($G = 1$)

iv) Vulnerable time = T_t

v) Halfs no. of collision & doubles η.

* Load of channel = No. of Request per time slot.

- LAN works at Data Link layer.
- Time to live is checked at Network layer.

OSI Model:

↳ Open system interconnection

1) ISO: International standards organization.

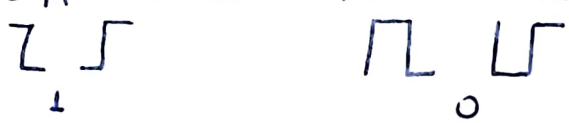
Physical Layer:

1) • copper: electric signal
   • Optical fibre: Light signal
   • wireless: electromagnetic signal

2) Transmission mode:-
   • Simplex
   • Half duplex
   • Full duplex (Default)

3) Topology:
   • Bus • Ring • Star • Tree
   • Mesh • Hybrid
   ↳ $^nC_2$ links, $n(n-1)$ ports.

4) Encoding:
   • Manchester Encoding: ⎍ ⎎ (1, 0)
   • Differential Manchester encoding → not unique



5) Baud rate = 2 * Bit rate
   ↳ for 1 bit → 2 signals

Data Link layer:

1) moves frame from 1 hop to next hop.

2) LLC → Error, flow control, synchronization
   MAC → Access control, physical addressing, framing

3) Flow control: GBN BACK-N (BT)

4) Any access control

5) Error control: CRC

6) Frame contains MAC address.

7) Start frame delimiter: 1010...11, NIC detects it.

8) To know end of frame:
   have fixed length frames / have frame size in header / end delimiter
   ↳ in ethernet, in token ring

9) If end delimiter occurs in data
   (i) Character stuffing
   (ii) Bit stuffing
   • ED used is 100...0 or 011...1
   • If ED is 0111, 0 is added after every 011

Network Layer:

1) moves pkt from source to destination.

2) Logical addressing: IP addr

3) congestion control → Open loop (prevent)
   → Closed loop (treat)

4) HOST to host connectivity

5) Routing, switching

7) Fragmentation

- NL - HOST to HOST connectivity
- TL - Process to process
- DLL - HOP to HOP

* Addressing:
Physical addr: MAC (48 bit) (Hexadecimal) } globally unique
Logical addr: IP (32 bit)
Port addr: (16 bit)
Socket addr = IP addr + Port addr

- MAC addr is used for identification in LAN. It does not help in routing. (permanent number) Present in NIC

Transport Layer:

1) Process to process delivery
2) Flow control : SR (B↓)
3) Error control : checksum
4) Segmentation.
5) Multiplexing & Demultiplexing

1) DNS : Domain Name server
   Domain name → IP addr.
2) ARP: Address resolution protocol
   IP addr → MAC addr.
3) PL, DLL r present in NIC ; NT, TL r present in OS.
4) Router has only 3 layers : PL, DLL, NL
5) Port no. of server = 80
6) FF.FF.FF.FF.FF.FF MAC addr is used for broadcasting.
7) Limited broadcasting is done at DLL
8) For a router, no of IP addr = no of interface
9) For a router, IP is permanent. MAC is temporary.
10) Every interface has different IP addr, MAC addr & subnet mask.
11) ACK is send from TL.

Session Layer:

1) Also called Network dialog controller
2) Authentication & Authorization
3) check point, Synchronization

Presentation Layer:

1) takes care of syntax & semantics.
2) Data translation
3) Encryption, Decryption
4) Data compression

- Bandwidth↑ : Freq ↑ : less range of towers
- Data send from 1 layer to other is called payload.
- PL addr start frame delimiter.
- Unicast MAC add : LSB of 1st byte is 0
- Multicast MAC add : LSB of 1st byte is 1
- Broadcast MAC add : FF.FF.FF.FF.FF.FF

ISO-OSI Model:

Please do not touch
Sachin's patni
Anjali

Application (Message)
Presentation
Session
Transport (Segment (TCP) / Datagram (UDP)) Gateway
Network (Fragment/Datagram (IP)) → Router
Datalink (Frame) → Bridge / Switch
Physical (MTU) → Hub / Repeater
↳ Max transmissable unit

- Gateway — all layers

TCP/IP Model:

Application : DNS, HTTP, etc
Transport : TCP, UDP, SCTP
Network : ARP, RARP, ICMP, IGMP, IP
Datalink
Physical

Ethernet (IEE 802.3) → LAN

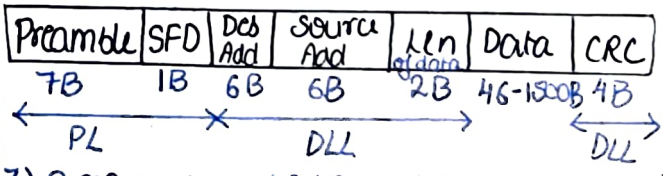
1) Topology : Bus
2) Switch looks like star but it's bus
3) Access control : CSMA/CD
4) Encoding : Manchester

NOTE: LAN works at DLL.

5) Data rate / Bandwidth

Normal	Fast	Gigabit
10Mbps	100Mbps	1Gbps

6) Ethernet frame structure



7) Preamble : 1010 ... 10 (alerts host)

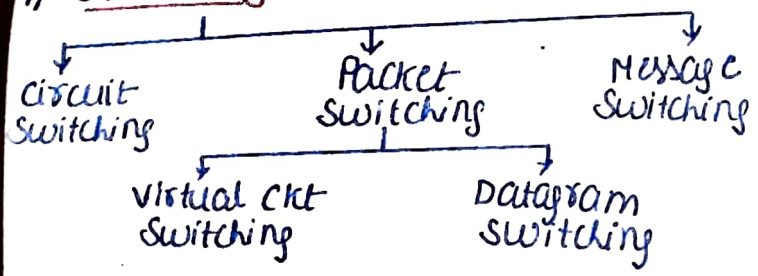
NOTE: Sour. addr is always unicast address. Dest. addr can be uni / multi or broadcast addr

8) Every station has a unicast MAC addr (LSB bit of 1st byte is 0)
9) Monopoly : when a host gets access to link, it keeps sending data.
10) Min frame size = $2T_p * B$ = 64B
11) In general, frame does not include preamble & SFD.

11)

Payload	Min	Max
Data	46B	1500B
Frame	64B	1518B

Switching:



1) Switching is done at Network layer.
2) Routing is optional.
3) Total time in circuit-switched n/w
= Setup time + T_t + T_p + teardown time
= Setup time + $\frac{L}{B}$ + $H * \frac{d}{V}$ + teardown time
H → no. of hops
d → length of each hop
4) Ckt switching is implemented at PL.
- No header needed.
- Data reaches inorder.

Packet switching:

1) Total time decreases on increasing no. of pkts but only upto a certain limit then increases.
2) Total time for 1 pkt = $H * \frac{L}{B} + H * \frac{d}{V}$ (no of hop = H)
3) In pipelining, 1st pkt will have H transmission, remaining pkts 1 transmission.
4) Optimal pkt size = $\sqrt{\frac{\text{Message len} * \text{Header len}}{(H-1)}}$ + Header len
5) Circuit sw better for large message packet sw better for small message
6) Pkt sw implemented at NL.

Virtual ckt switching:

1) Connection oriented service.
2) Only 1st pkt requires global header rest require local header.
3) ATM user it.
4) Implemented at DLL. (only ID is required 1st time)
5) 1st pkt reserves resources for subsequent pkts.
6) No pkt may get discarded at any intermediate router.
7) Costly

Datagram switching:

1) Connectionless service.
2) Data may appear out of order.
3) No resource reserved
4) All pkt requires global header.
5) IP n/w uses it. (whatsapp call uses it)
6) Implemented at NL.

NOTE: virtual ckt sw is charged for time, datagram sw charges for data transmitted.

Internet Protocol

Version 4bit	Hlen 4bit	service 8bit	Total length 16 bit	4B
Identification 16 bit		0	DF / MF	Frag offset 13bit	4B
Time to leave 8 bit	Protocol 8bit	Header checksum 16 bit		4B
Source IP (32 bits)				4B
Dest IP (32 bits)				4B
Options (0-40B)				
Data				

1) IP protocol used at NL.
2) Min header length = 20B
Max " " = 60B
3) Header len is divided by 4 & stored in Hlen.
4) If header len is not ÷4 then 0's are padded in option field to make it multiple of 4.
5) Service: first 3 bit → priority bit
4bit → service type (only 1 bit can be 1)
Last bit not used.
6) Total len = Header len + Data
7) Min len of datagram = 20B (0B data)
Max " " = $2^{16}-1$ B = 65535B
8) Min data len = 0B
Max " = 65515B
9) All fragments of some datagram will have same ident. number.
10) Iden. no. is unique in a connection.
11) DF: Do not fragment
12) MF: More fragment
Go for last fragment

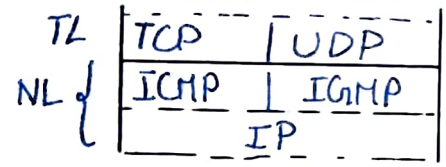
13) Frag. offset : NO. of data bytes ahead.
↳ scaled by 8.

14) While dividing data into fragments make sure that data in each frag. (except last) is multiple of 8.

15) Time to live : Max NO. of hops.
↳ decrement by 1 at each intermediate router & at destination.
↳ checked at NL.

16) When TTL becomes 0, pkt is discarded & ICMP message send to sender.
↳ used to avoid infinite loop.

17) The order in which router eliminates datagram from buffer is
ICMP > IGMP > UDP > TCP
(01) (02) (17) (06)



18) Header is divided into grps of 16 bits, added & its 1's comp is header checksum.

19) IP protocol is not reliable → no ACK

20) Checksum field is calculated at each router, as TTL is changed at each router

21) Fields that may change at router:
Option, HLen, TLen, MF, Frag. Offset, TTL, H checksum

22)

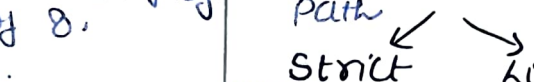
NID	HID	
✓	✓	valid IP
✓	0	NID
✓	1	DBA
1	1	LBA
127	✓	Loop back addr.
0	0	default purpose (DHCP)
1	0	subnet mask / nw mask
0	✓	host within n/w

NOTE: IP is connectionless, not reliable & host to host protocol

23) Option field is used for record routing, source routing, padding, timestamp, no operation, End of option.
↳ gap blw 2 options

24) Record routing : record IP addr of all routers in path.
↳ can have max 9 routers addr.
↳ in option 4B is used for option type.

25) Source routing : used to specify path



Strict: specify all routers onway
Loose: specify few routers (Other routers can also be visited.

26) Every router records incoming & outgoing time in option.

Segmentation & Fragmentation
↳ at TL ↳ at NL

1) Fragmentation is done at router not at source.

2) At source segmentation is done so that there's no need of fragmentation.

3) MTU → Max data size at DLL

4) Fragments reassemble at NL of receiver.

5) Amount of data send in 1 fragment such that
- (data + header) < MTU
- data size multiple of 8. (except last frag)

6) Offset of 1st fragment = offset of parent fragment.

7) MF of parent fragment = MF of parent fragment.

8) Overhead = (Total no. of frag − 1) * Header len

9) Efficiency = $\frac{\text{useful bytes (data)}}{\text{Total Bytes transferred}}$

10) Bandwidth utilization or throughput = Bandwidth * Efficiency

11) Initially, when pkt is not fragmented, MF & offset is 0.

12) 1st frag → offset = 0
Last frag → MF = 0

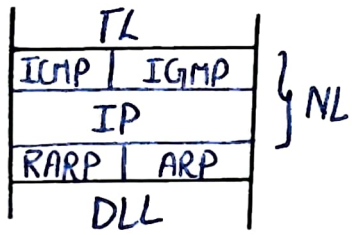
13)

MF	Offset	
1	0	1st frag
1	!=0	intermediate frag
0	!=0	Last frag
0	0	NO fragmentation

Protocols at NL:

(a) Address Resolution Protocol (ARP)

1) IP addr → MAC addr. (Media access control)



2) ARP is used to find MAC addr of host in same network.
3) ARP request is broadcasted. ARP reply is unicasted.
4) Every host & router maintains ARP table.

(b) Loop back address : 127.x.x.x

used for
- testing self connectivity
- client server testing.

(c) Reverse ARP (RARP) (obsolete - not used)

1) MAC addr $\xrightarrow[\text{server}]{\text{RARP}}$ IP addr.
2) MAC addr is present in ROM, IP addr is present in RAM.
3) RARP request → limited Broadcasted
   RARP reply → unicasted
4) Every nw has RARP server.
5) Static mapped RARP server table so wastage of IP.

(d) BOOTP (Bootstrap protocol)

1) Used to find IP addr from MAC.
2) Works at AL whereas RARP works at NL. (similar to RARP)
3) There's only 1 BOOTP server.
4) Nws which don't have server has Relay agent.
5) Static mapped table

(e) DHCP (Dynamic host conf. protocol)

1) Used to find IP addr from MAC addr.
2) Same as BOOTP, table has dynamic & static part.
3) Operated at AL.
4) No. of IP addr used = No. of host online
5) Backword compatible with BOOTP

NOTE: Port no. for BOOTP & DHCP → client: 67 server: 68

(f) ICMP (Internet Message control protocol)

- NL protocol, not reliable (ICMP sits in IP pkt)

ICMP

Error handling & Feedback Messaging
- TTL exceed
- Parameter problm
- Source Quench
- Source Redirect
- Destination unreachable

Request & Reply
- Echo request & reply
- Time stamp R&R
- Address mask R&R
- Router solicitation & advertisement

1) When ICMP pkt is lost, no ICMP pkt is generated
2) We can trace route using TTL.

* Error Handling:

- TTL exceed: ICMP pkt is sent to source when pkt is discarded becos TTL=0
- Source quench: say stop to source
- Parameter prblm: strict source routing & router in way is down.
- Destination unreachable
  - Dest host unreachable - dest down
  - Dest port unreachable - invalid port
- Source Redirect: when better path available, router warns source

* IP header along with port no. from TCP/UDP is cut & placed in ICMP pkt

* Request & Reply:

- Echo R&R: used to test NL of dest & intermediate routers
- PING (Pkt internet groper) command is used for it.

- Router solicitation: to choose default gateway
- Router advertisement: when a new router is connected, it send. Its one way message, no reply.
- N/w mask R&R: to get nw mask, default gateway provides.

NOTE: Traceroute is unreliable, uses unreliable ICMP protocol AL NL

- ICMP can be used to find path MTU. Its unreliable.

Routing:

- Static (Not in syllabus)
- Dynamic
  - Distance vector Routing (DVR)
  - Link state Routing (LSR)

- In dynamic, routing table changes with topology & traffic.

Distance vector Routing:

1) Repeat proadure for (n-1) times as max shortest distance may be (n-1) (n → no. of nodes)
2) check if there's better path between AB then this link is unused.
3) Don't use existing value for updation even if its smaller. Update according to distance vectors only.
4) Count to infinity problem.
   Good news spread fast, bad spread slow
   router added — router removed
5) Sol^n of count to ∞ is split horizon along with DV, next hop is also shared.

- In LSR, each router floods its table, then each router apply dijkstra algo.
- Problems in LSR:
  - Traffic
  - Duplicate pkt
  - Seq no corrupted
  - Transient problem

DVR	LSR
1) Bellmon Ford algo	1) Dijkstra algo
2) comparatively slower	2) converges faster.
3) count to ∞ prblm	3) X
4) Less traffic	4) More traffic
5) Less Bandwidth, we sent only DV	5) More bandwidth, we sent entire pkt.
6) Local knowledge	6) Global knowledge
7) ∞ Loop possible	7) Transient loop possible
8) Low CPU utilization.	8) Intensive utilization
9) Periodic update	9) Update when something gets wrong.
10) Implementation is RIP	10) OSPF

NOTE: In RIP, max hop count is 15. 16 is considered as ∞ (destn unreachable)

- RIP uses UDP transport protocol with port no 520.

Transport layer:

- can be connectionless or connection oriented.
  - Transmission control protocol
  - User datagram protocol

TCP:

Source port (16 bit)	Des port (16 bit)	4B
Seq no (32 bit)		4B
ACK no (32 bit)		4B
Hlen 4bit / Reserved bits(6) / URG ACK PSH RST SYN FIN	Window size (16 bit) (Advert window)	4B
Checksum (16 bit)	Urgent pointer (16 bit)	4B
Option (0-40B)		
Data		

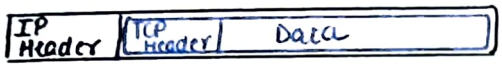
1) IPaddr + Port addr = Socket addr
   32 bit + 16 bit = 48 bit
2) TCP → Byte stream protocol each byte is given unique no.
   NL → Pkt stream protocol each pkt is given unique no.
3) TCP is connection oriented, end to end, Byte stream protocol.
4)

Port no.	Name
0-1023	well known
1024-49151	Reserved for future
49152-65535	General

5) TCP connection is uniquely identified by socket number.
6) Seq no field contains seq no. of 1st data Byte.
7) Ack no tells next expected data byte from sender.
8) Wrap around time: time taken to use all 2^{32} seq nos.
   $WAT = \frac{\text{Total sequence no's}}{\text{Bandwidth (B/sec)}} = \frac{2^{32}}{B}$
9) Life time of TCP segment ≈ 3min
10) WAT ≫ LT, generally
11) If WAT < LT, increase WAT by taking extra bits in option for seq no.
12) Min no. of bits required in seq no. field to avoid WAT within LT
    $= \log_2$ [Life time * Bandwidth]
13) Length of TCP header = 20B - 60B
    Scaling factor = 4

14) Range of Hlen field = 5 - 15

15) TCP pkt is in IP pkt.



16) TCP is reliable, process to process delivery of entire message.

17) TCP connections are full duplex.

18) TCP connection establishment has 3 phases
- connection establishment
- Data transmission
- connection termination

19) Random initial seq no. is used to establish connection.

20) conn. establishment is a 3 way handshake process (Request, Reply, ACK)

21) window size can be increased by using extra bits from option.

22) SYN=1 → consumes 1 seq no.
ACK=1 → " 0 " "
FIN=1 → " 1 " "
1 Data Byte → " 1 " "

23) Pure Ack don't take seq no.
Piggybacking takes " "

24)

SYN	ACK	Meaning
1	0	request
1	1	reply
0	1	Pure ACK or data
0	0	NOT possible

25) In TCP, except 1st request segment, all segment will have ACK.

26) TCP uses cumulative ack.

27) FIN: Finish flag used to terminate connection.

28) ACK is small size, can be send without connection (reserving buffer)

29) connection can be closed with min 3 pkt exchange. (FIN, FIN+ACK, ACK)

30) PSH: (Push) immediately send data, don't wait for buffer to get full.

31) URG: (Urgent) to give data highest priority (1)

32) Urgent pointer points to end of urgent data byte.
it's evaluated only if URG bit is 1.

33) No. of Urg Bytes = Urg pointer + 1

34) End of Urg Byte = Seq no. of 1st byte in segment + urgent pointer

35) RST (Reset) used to reset TCP connection. (immediately terminate connection)

36) Adv window contains size of the receiving window of sender empty.

37) when sender is waiting, after persistent timer sender should send 1B data & check if adv window is empty

38) checksum is calculated on TCP segment & some part of IP header which does not change like IP addr

39) Option field:
- Timestamp - to increase seq no.
- window size extension
- Parameter negotiation
- Padding - to make header len * 4

40) Each connection is associated with 4 windows.

41) Data may arrive out of order, but no out of order data is delivered to the process.

42) Sender knows that TCP segment is lost when
- Time out timer expires
- or it receives 3 duplicate ack.

43) For 3 duplicate ack, min 5 segments need to be transmitted.

#Congestion control:

1) TCP has congestion control.
Cong. control is also at NL but not supported by IP, not in syllabus.

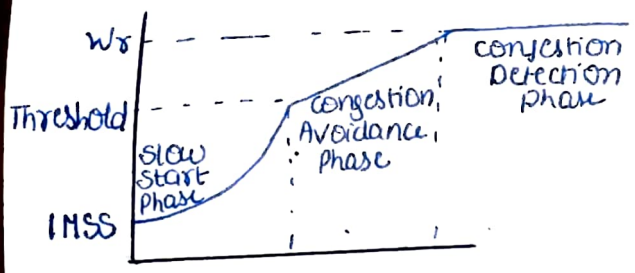
2) $W_S = \min(W_C, W_R)$ (Receiver)
Sender → congestion window size
By default starts with 1 segment
W_S, W_C, W_R are in terms of no. of segments.

3) Threshold = $\frac{\text{Max no. of TCP segments receiver window can accomodate}}{2}$

4) Till threshold, W_C will grow exponentially (double) then grow linearly.

5)

6) congestion control phases



Slow start	congestion Avoidance
◦ If ACK arrives $W_c = W_c + 1$	◦ If ACK arrives $W_c = W_c + \frac{1}{W_c}$
◦ After 1 RTT $W_c = 2 * W_c$	◦ After 1 RTT $W_c = W_c + 1$

7) congestion Detection phase:
◦ Time out (severe congestion)
Resume slow start phase with W_c=1MSS & threshold is set to half of current congestion window size.
◦ 3 ouplicate Ack (Mild congestion)
Resume congestion avoidance phase with W_c = new threshold.

Time out Timer in TCP:

* **Basic Algo:-**
◦ TOT = 2 * RTT
◦ Next RTT = α IRTT$_{initial}$ + $(1-\alpha)$ ARTT$_{Actual}$
α → smoothing factor $0 \le \alpha \le 1$

* **Jacobson's Algo:**
◦ TOT = 4 * ID + IRTT
↳ initial deviation (guessed)
◦ NRTT = α(IRTT) + $(1-\alpha)$(ARTT)
◦ ND = α(ID) + $(1-\alpha)$(AD) $0 \le \alpha \le 1$
↳ Next deviation
◦ Actual deviation, AD = |IRTT − ARTT|

* Karn's Modification - if ack is not received within timer, retransmit with double TOT.

---

* Silly window syndrome:
- Sender sending small small segment
→ sol: Nagle's algo (buffer while waiting for ack)
- Receiver accepting small small bytes.
→ sol: clark's algo (wait then advertise)

* Traffic shaping: congestion control method.
- Leaky bucket - Token bucket
* Token bucket:-
capacity of token bucket = C tokens
token enter bucket at rate of r token/sec.
Max no. of pkts that can enter n/w in t sec = (C + rt)
Max avg rate, M = $\frac{C + rt}{t}$ pkts/sec
Token arrival time, t = $\frac{C}{M - r}$

User Datagram Protocol (UDP):

1) Connection less protocol
2) unreliable, fast, stateless protocol
3) Not guarantee inorder delivery.

source port no 16 bit	Dest port no 16 bit
Length of Data + header 16 bit	checksum 16 bit

4) Header is of fixed size 8 Byte.
5) checksum is calculated similar to TCP but it's not mandatory.
If checksum not used, fill with 0.
6) Does not provide flow, error or congestion control.

* **Need:**
i) Application that need 1 request one reply like DNS, BOOTP/DHCP.
ii) Broadcasting, Multicasting
iii) App which require speed rather than reliability.
iv) that require constant data flow, multimedia data transfer
7) There is no seq no., no ack.
8) Protocols using TCP: HTTP, FTP, SMTP, POP
Protocols using UDP: DNS, BOOTP/DHCP, SNMP, TFTP, DHCP, all real time protocol.

Application Layer protocol :

	DNS (DN→IP)	HTTP (Web pages)	SMTP (Email)	FTP (hu)	POP (Email retrieving)	IMAP
State	Stateless	Stateless	Stateless	Stateful	Stateful	Stateful
TL protocol used	UDP	TCP	TCP	TCP	TCP	TCP
Connection	-less	-less	-oriented	-oriented	-oriented	-oriented
Persistent	X	HTTP 1.0 x HTTP 1.1 ✓	✓	control connection ✓ Data conn. X	✓	✓
Port no.	53	80	25	20 : data conn. 21 : control conn.	110	143
Inband/out of band	Inband	Inband	Inband	out of band	Inband	Inband

ⓐ DNS : (Domain name service)

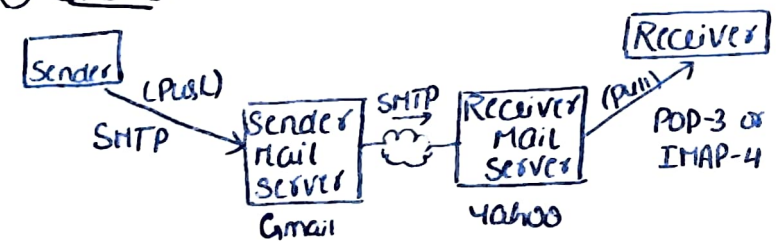
1) Domain name space → Generic (3 char) .com
   → country (2 char) .in
   → inverse (IP → DN)
2) DNS can contact server in 2 ways : Iterative way, recursive way.

ⓑ FTP : (File transfer protocol)

NOTE: Persistent means connection is forever
non-persistent means con- is closed after file trans
- out of band means control & data info flow in different connection.

1) FTP solves all prolm of different file name convention, data representation & different directory structure

ⓒ EMail :



ⓓ SMTP : (Simple Mail Transfer Protocol)

1) Push protocol
2) has 2 component:
- user agent prepares message
- Mail transfer agent transfers it.
3) SMTP can handle only 7 bit ASCII text in message
4) Persistent, can send multiple emails at once.

* MIME (Multipurpose internet mail Extension protocol) deals with non text content & can have unlimited message length.

ⓔ POP3 (Post office protocol)

1) Message access protocol
2) Pull protocol

Drawback: can't organize mail on server; can't partially see content before downloading.

ⓕ IMAP-4 (Internet mail access protocol)

1) eliminates drawbacks of POP3
2) Mail can be accessed from multiple device
3) It is bidirectional i.e all changes made on server or device are made on the other side too.

NOTE: Telnet : 23
DHCP : 67 server
68 client

ⓖ HTTP (Hyper text transfer protocol)

Status code of response:
- 1xx : request getting processed
- 2xx : request accepted
- 3xx : action needed from client
- 4xx : error at client side
- 5xx : error at server side.