20/2/24.

Internet - heterogenous devices.

CSP - cloud service providers.

DHCP - dynamic host configuration protocol → is a server.
↳ works on the process DORA - discover, offer, request, acknowledge.
by broadcast (10.0.0.1 - 255.0.0.255) ↳ lease (1 hr / 1 day etc).

ip config /release

ip config /renew

ping - packet internet groper

Data - is a raw fact (volume)
↓
info - which has meaning.
↳ knowledge → wisdom.

3 types of data → structure, unstructure, semistructure

Structure:
- stored in a desired format.
- RDBMS (SQL)
- similar to excel.
- CRUD - create, read, update, delete.
- 20% of data is structured.

Unstructure:
is in form of object.
(NOSQL)
Generally 80% data is unstr.
Ex: Images, videos, audios etc.

Semistructure:
- mixture of both.
Eg - html

- Data life cycle - depends on its value.

1/3/24

Types of network

1) Local area n/w (LAN) - ethernet, wifi, FDDI
   → covers limited geographical area. → same building
2) Wide " " (WAN) - T_1 lines, leased line, DSL → Blr — ISP — Mys → Internet
   (has infrastructure)
3) campus " " (CAN) → diff buildings ← n/w b/w - same location.
4) Metropolitan " " (MAN) - diff locations
5) Personal " " (PAN) - bluetooth, NFC, ethernet.
   (NFC → near field comm.)

- N/w is determined by its size, location, ownership.

Network Model (Layered task)

OSI model - open system interconnect
↳ reference model.

Practically implemented n/w → TCP/IP.

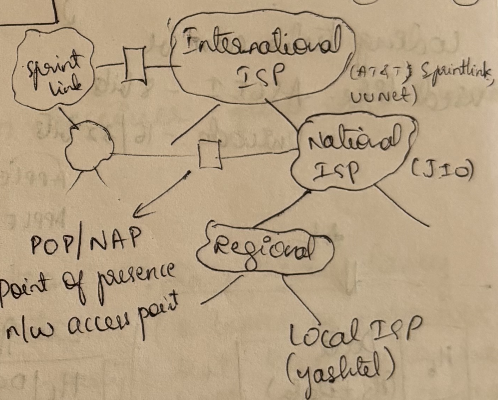
7. Application layer
6. Presentation "
5. Session "
4. Transport "
3. Network "
2. Data link "
1. Physical "

(7, 6, 5 → S/w layers; 4 → Intermediate layer; 3, 2, 1 → H/w & S/w layers, router, cables)

7) AL → provides user interface.
http, ftp, telnet, ssh (remote accessing), dns, dhcp, snmp (n/w management protocol), smtp (mail service)
→ all these are protocols - consists of set of rules!

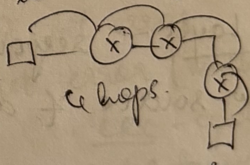
- Data / msg gets added with additional info - encapsulation.
↓
decapsulated at receiver side
↳ so that receiver can understand what client is asking.

Internet - n/w of n/w, interconn. of various n/w.

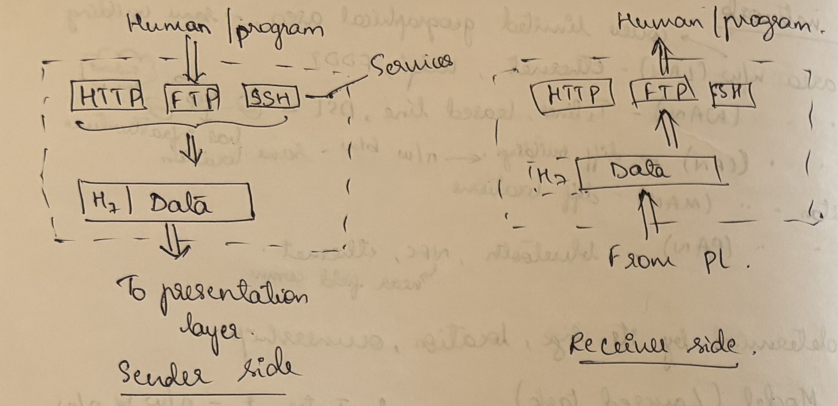


POP/NAP
- Point of presence
- n/w access point

- Distance is measured in hop count in n/w.
- will be < 8 hops.

Ex:


- Discovered by Arpa in 1960's.



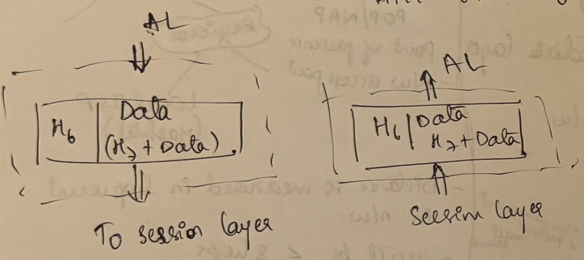
6) PL – 3 works → translation, encryption, compression.

coding techs can be used like ASCII - 8 bits
Unicode - 16/32 bits

securing of data by transposition, substitution
APPLE → EAPPL
APPLE → USSMD
A P L E
U S M D

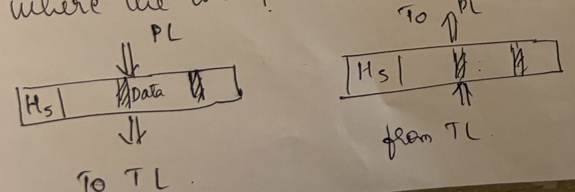
reduces data size.

http - 80
DNS - 53
Telnet - 22
SSH - 23
} port add.



5) Session Layer – responsible to create, maintain, terminate the session b/w source & destination.

a) Dialog control
b) Add checkpoint – if connection gets disconnected while da file transferring then once the " is reestablished, it'll start from where the checkpoint is assigned. (100bytes – 100 CP. 550 —— starts from 500.)



4) TL: – It is resp. to deliver every segment from sender process to receiver process (or) end-to-end delivery.

a. Port address (or) service/specific address
b. Segmentation & reassembly.
c. Flow control.
d. Error "
e. Connection " – connection oriented.

well known
1 - 1023
why you use it.

– msg is divided into equal parts
is segmentation.
unknown is given sequence no.

3 types of port address (16 bits) → i) well known → 0 - 1023
ii) Registered → 1024 - 49153
iii) Unregistered → 49154 - 65535.

Msgs ÷ing into small management parts – segmentation.
At receiver side, these parts are combined – reassemble.
Sequence no. should be given.

–When the sender rate is higher than receiver – flow control works.

S (100bps) —— R (10bps)

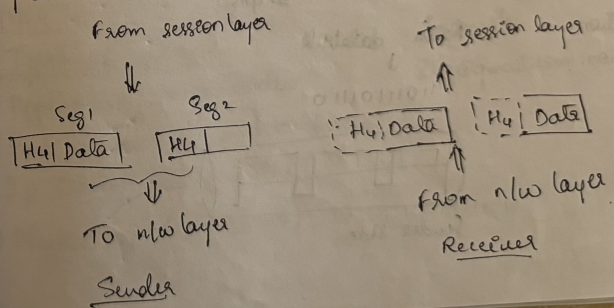
Error handling – 2 types – error detection – data redundancy. (2 copies of msg are sent. Receiver compares both & detect errors if any).
– error correction.

checksum.
– Haffman
– Shanon Fanon.

01101110 1 parity
Checks 8 bits & compares with parity.

• parity → no. of ones is odd then =1.
– Single bit error
– Burst error (multi bits error).
• CRC – cyclic redundancy check.

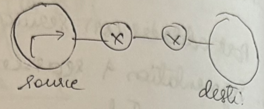
–All these (TL) mechanisms are implemented in protocols TCP & UDP.



3) Network layer - ~~responsible~~ responsible to send a packet from source to destination.

i) Logical address / Ip address (32 bits)
↓
has 2 info < n/w info, host "

ii) Routing.
↓
RIP
EIGRP } BGP
OSPF Protocols.



32 bits → grouped in octate → ∴ called doted decimal.

00000001 → 1.1.1.1

Ex: $\boxed{10.24.1.}$ 1 — n/w
1 → rep n/w
0 → " host.

10.24.1.1
10.24.1.2 } same address.

Subnet mask : 255.255.255.0
↓
used to diff. the host & next n/w.
host — changes; n/w — fixed.

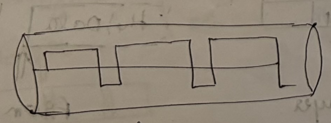
2) Data link Layer. - responsible to transfer frames from one hop to another.

i) Framing.
ii) Physical address (48 bits) - S & d in same location. (MAC address).
iii) Flow control
iv) Error "
v) Access " < Point to point link, Broadcast link.

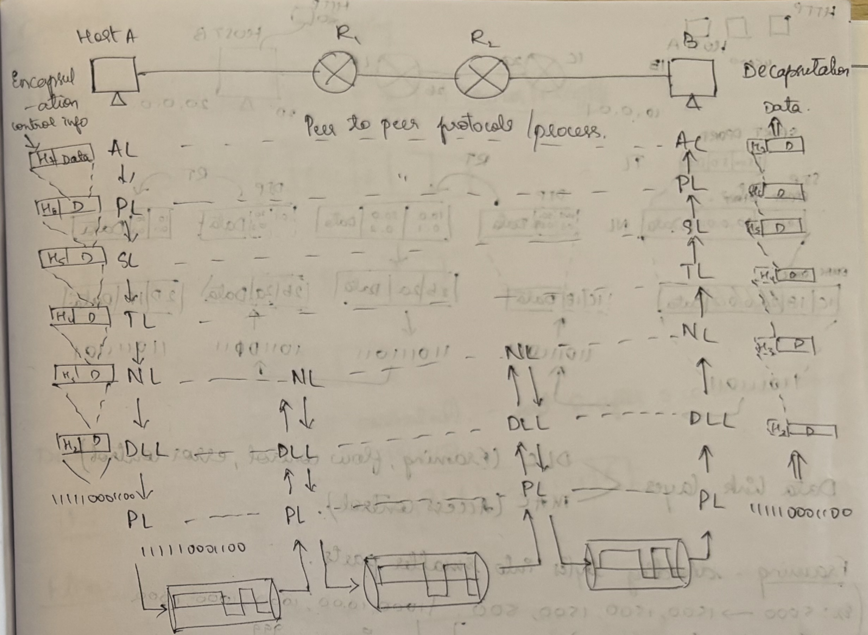
1) Physical layer - responsible to transfer every bits from one hop to another.

i) Type of medium < wired - coaxial, ethernet, OF; wireless - air, 802.11 | 2.4GHz, 5.GHz.
ii) Topology
iii) Data representation
iv) Line configuration.

From datalink
↓
1101110111 0



Sender side.



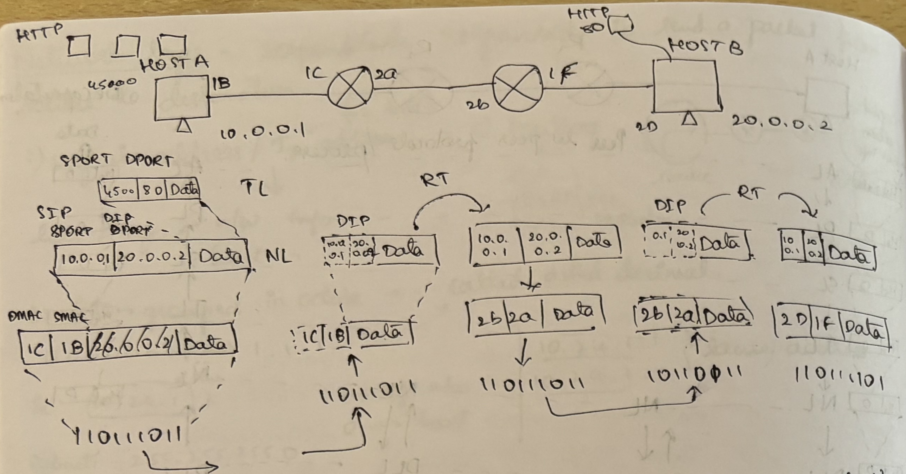
OSI model | TCP/IP

AL, PL, SL } → AL — HTTP, FTP, SSH, Telnet, SNMP, SMTP, DNS, DHCP
TL → TL — TCP, UDP, SCTP
NL → N/w / internet layer — ARP, RARP, IP, ICMP, IGMP
DLL, PL } → Host to n/w — IEEE 802.3, 802.15.4, 802.11, 802.16

13/3/24.

Types of address — Example

1) Application specific address — xyz@gmail.com
2) Port address (16 bit) 0 - 65535
3) Ip address (32/128 bits) - 10.1.1.1
4) Physical address (48 bits) - 1a:1b:1c:1d:1e:1f
(mac)



Data link layer < DLC (Framing, flow control, error control) / MAC (Access control).

Framing - dividing bytes into smaller parts.

Ex: 5000 → 1500, 1500, 1500, 500. | 1000, 1000, 1000, 1000, 1000

Ethernet → (MTU ~ 1500 max)

2 types — fixed of size framing / variable ,,

Disadv - efficiency

| F | 1500 | F | | F | 1500 | F | | F | 1500 | F | | F | 500 | F |

start of frame, end

01111110

byte / char, bit, esc (extra byte when F is in the data), - neglects the esc & takes next byte.

| F | | esc | F | | | F |

Character stuffing & destuffing.

| Flag | Header | | | Esc | | F | | | | Trailer | Flag |

1 byte

| | | ESC | ESC | | | ESC | F | | |

| | | ESC | ESC | | ESC | F | | | |

| | | ESC | | F | | | |

Data - taken in bits. (bit oriented). - Add 0 after 5 consecutive ones

F = 01111110

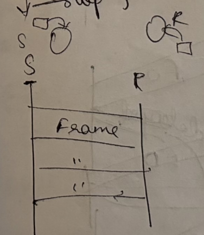
F | 0111110111110111011111010111110011111 | F

Flow control

Protocol

- Noiseless channel → S — R (no data duplicate, " data loss, " error)
  - Simplest
  - Stop & wait
- Noisy channel
  - Stop & wait ARQ (Automatic repeat request)
  - Go back N (GBN) ARQ
  - Selective repeat ARQ

~~Stop & wait.~~

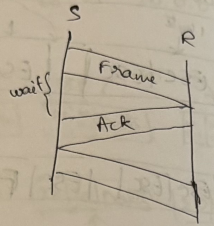



```
while (true)
{
  if (Event (Request to send()))
  {
    Getdata();
    Make frame();
    Send frame();
  }
}
```

```
while (true)
{
  if (event (arrival notify))
  {
    receive frame();
    extract data();
    deliver data();
  }
}
```

Stop & wait

After sending a frame to reciver, sender should wait (for some time) till the ack from receiver is obtained.



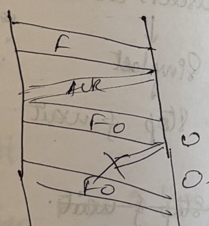

```
while (true)
cansend = true;
{
if (event (request to send () && cansend)
{
  getdata ();
  make frame ();
  send frame ();
  cansend = false;
}
```

Stop & wait (noisy)

Sender waits for ack from receiver & receiver waits for frame from sender.

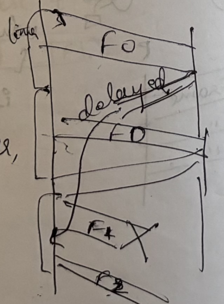
∴ use timer

(say, set timer = 10 sec. If ack is not received within 10 sec, the frame is resent).



- If the ack fails, sender resends the frame — this leads to data duplication.
  ∴ Give seq. no to frames.

- If the acknowledgement is delayed, sender resends the frame & receiver the ack. Further, when the frame1 is lost & the delayed ack is obtained ~~the~~ to sender, it thinks the F1 is sent & received — this leads to - data missing.
  ∴ give lables to acks.



ARQ - automatically requests for f & ack.

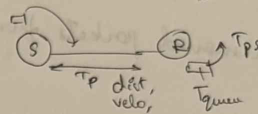
Delays.

$T_t = (T_{fr}) = \frac{L}{B}$ — no. of bits / bandwidth

Ex: 1000 bits $T_t = \frac{1000}{1000}$.

$1 \text{kbps} = 10^3 = 1024$ $\frac{1024}{1000}$

T_p — propagation delay. $T_p = \frac{d}{V}$.

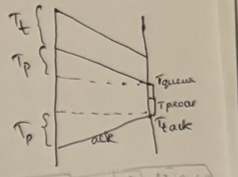


$\eta = \frac{\text{usefull time}}{\text{total}}$ ∴ → $T_{t\,data} + T_{p\,data} + T_q + T_{proces} + T_{t\,ack} + T_{p\,ack}$

depends on config. will be less bits ∴ doesn't take much time ≈0

$= T_t + T_p + T_p = T_t + 2 \times T_p$

$= \frac{T_t}{T_t + 2T_p} = \frac{1}{1 + 2\left(\frac{T_p}{T_t}\right)}$ for LAN



$T_t = 1 \text{msec}$ $T_p = 1.5 \text{msec}$

$\eta = \frac{1}{1 + 2\left(\frac{1.5}{1}\right)} = \frac{1}{4} = 25\%$.

$\frac{T_t}{T_t + 2T_p} \geq \frac{1}{2}$.

$2T_t \geq T_t + 2T_p$

$T_t \geq 2T_p$.

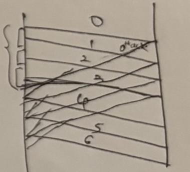
Eg: 10 packets needs to be transmitted, Every 4th packet gets lost.

1 2 3 4 5 6 7 8 9 10 = 13 packets are transmitted.

$\uparrow_4$ $\uparrow_7$ $\uparrow_{10}$

Pipelining. — ↑ses efficiency. ↓ sliding window mech.

buffer 6 5 4 3 2 1 0



In T_p, other packets are transmitted. Here by the time ~~1st~~ packet is rec'd, 0th ack is obtained, the window is slides.

$\eta = \frac{N \times T_t}{T_t + 2T_p}$

- GBN → 1. Sender buffer size → N > 1
- SR   2. Receiver " → 1
          3. Cumulative AR

GB4. - If 3 is lost then go back to last packet than sender tried & move N times & re te all 4. 6 5 4 3 2 1 0

1 2 3 4 rete. again.

1 2 3 4 5 6 7 8 9 10

4 5 6 7 8 9 10 = 18.

Every 4th is lost AF 4

- $T_t = 1\,msec$ $T_p = 49.5\,msec$. GB10 $\eta = ?$

$$\eta = \frac{N}{1+2a} = \frac{10}{1+2\cdot\frac{49.5}{1}} = \frac{10}{1+99} = \frac{10}{100} = 10\%.$$

1	0	3	2	1	0

Removed packets from buffer can be reused.