Computer network: Unit - I

Defin:

A comp network is a set of nodes connected by communication links.

A node can be a computer, printer or any device capable of sending / receiving data generated by other nodes in the network.

nodes: computer, printer, camera, switches, routers etc

A communication link can be a wired (computers) or wireless (phones) link.

The link carries the info.

Characteristics:

→ fault tolerance (ability to continue working despite failures, ensure no loss of service)
→ scalability (growth based on needs, good perfo^{m} after growth)
→ QoS (quality of service) (set priorities, manage data traffic to reduce data loss, delay etc)
→ Security. (ability to prevent 1. unauthorized access
2. misuse
3. forgery
ability to provide 1. confidentiality
2. integrity
3. availability)
→ performance (can be measured in many ways including transit time and response time. Transit time is the amt of time required for a message to travel from one device to another, response time is the time elapsed b/w an inquiry and a response. It is often evaluated by 2 networking metrics: 1. throughput
2. delay)
→ reliability (measured by 1. freq of failure
2. time taken to recover from freq of failure.
3. network's robustness in a catastrophe)

Data Communication:

→ data comm^{n} is the exchange of data b/w two nodes via some form of link (transmission medium) such as cable.

→ data flow : ① Simplex
② Half Duplex.

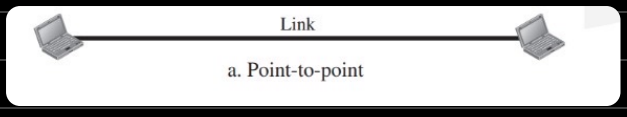
→ Simplex : Comm^{n} is always unidirectional.
One device can transmit & other device can receive.
ex: keyboards, Traditional monitors //

→ Half duplex : Comm^{n} is both directions but not at the same time.
If one is sending, other can only receive and vice versa.
ex: walkie talkies

→ Full Duplex : ~~~ both directions.
device can send & receive data at the same time.
ex: Telephone line //

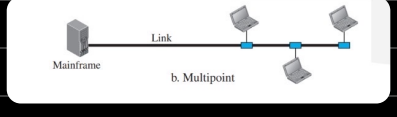
Types of Connection: Based on phy structure:

1. Point-to-point Connection:



→ provides a dedicated link b/w two devices.
→ the capacity of the link is reserved for transmission b/w the 2 devices only.
→ most of them use an actual length cable/wire to connect the 2 ends, but other options such as microwave / satellite links are also possible.

2. Multipoint connection:



→ more than 2 devices share a single link.
→ the capacity of the channel is shared b/w all devices.
→ if several devices can access the link at the same time, it is a spacially shared connection
→ if devices take turns to access then it is a time shared connection.

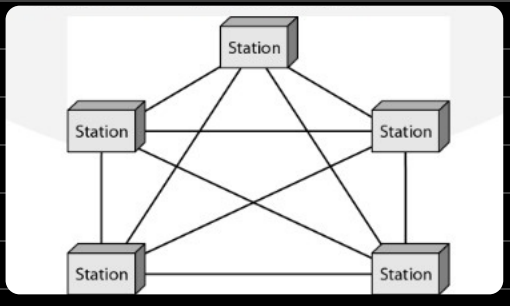
Network Topology:

→ Physical topology is when the network is laid out physically
→ When 2 or more devices are connected, it forms a link.
→ When 2 or more links are connected, it forms a topology / network topology.
→ It is a geometric representation of the relationship of all the links and linking devices (nodes) to one another.

1. Mesh Topology:

→ Every device has a dedicated point to point link to every other device.
→ We need $n(n-1)$ phy links to connect n nodes if the links are simplex mode.
→ In duplex mode, we need $n(n-1)/2$ links.
→ To accomodate that many devices we need $n-1$ I/O ports in the network to be connected to the other $n-1$ stations.

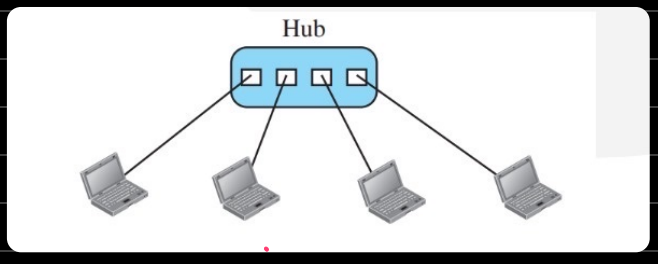
Advantages:
→ Each connection can carry its own load. So, traffic b/w the devices is not shared.
→ Robust.
→ Higher privacy / security.
→ Ease of fault identification.



Disadvantages:
→ Large amt of cabling & I/o ports.
→ Difficult to install / reinstall.
→ Expensive.

2. Star Topology:
→ Each device has a point to point link with only the central controller a.k.a the hub.
→ Devices are not directly connected to one another.
→ If one device has to send data to another it sends it to the controller and the hub relays the data to another connected node.

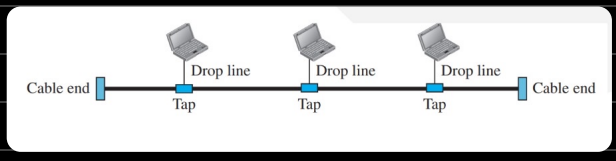
→ Advantages:
Less cabling, less I/o ports.
Less expensive.
Robust.
Ease of fault identification



Disadvantage:
the hub is a single point of failure.

3. Bus Topology:

→ It is a multipoint connection.
→ one long cable acts as a backbone to link all the devices in a network.
→ nodes are connected to the bus cable by Droplines and Taps.
→ a dropline is a connection running b/w the device and the main cable.

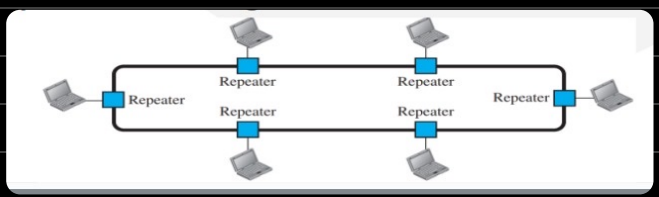


Advantage:
→ uses less cabling / I/o port.
→ easy installation.

Disadvantage:
→ difficult reconnection/fault identification.
→ difficult to add new devices.
→ difficult modification / replacing the backbone.
→ limited length / no. of nodes.
→ break in bus cable stops all transmission.

4. Ring Topology:

→ each device has a dedicated point to point connection with only the two devices on either side of it.
→ a signal is passed along the ring in one direction from device to device until it reaches its destination.
→ each device in the ring incorporates a repeater.
→ when a device receives a signal intended for another device, its repeater regenerates the bits & passes them along.



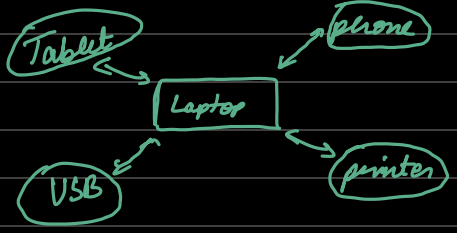
Advantages:
→ easy to install.
→ reconfigure / add / delete.
→ simple fault detection.

Disadvantages:
→ limited length/no. of nodes.
→ break in the ring can bring the entire network down.

Classification: (5)

① PAN:

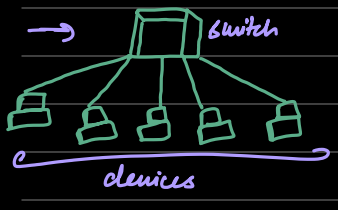
→ personal area network.
→ Most Basic type of comp network.
→ restricted to a single person / individual workspace.
→ network range: 1 - 100 meters b/w user & device.
→ Transmission speed is very high & easy maintenance, low cost.
ex: USB, printers, home network.



→ Types ⇒ wired (cables, USB), wireless (WiFi, Bluetooth connt b/w devices)

① LAN:

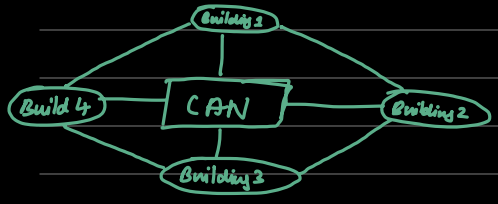
→ local area network is a comp n/w that interconnects computers within a limited area.
→ local: small area. ⇒ 2kms range.
→ restricted to a limited area.
→ like a school, residence, uni etc.
→ high speed
→ easy maintenance, low cost.



→ all the comps can communicate with each other internally.
→ LAN devices: ① wired LAN ⇒ cables ⇒ Ethernet (used to connect 2 or more nodes)
ex: Hub, switch
⇒ comps are connected to the switch with the help of a ethernet cable
② wireless LAN: ⇒ no cables.
ex: wi-fi

② CAN: (Campus area network)

→ Bigger than L.A.N, smaller than M.A.N.
→ used in colleges & schools.
→ network spreads over several Buildings.
→ uses ethernet technology mainly. (1km - 5km)
→ high speed, moderate cost.



② MAN: ex: area to area in a city, large area with many Buildings.

→ metropolitan area network.

→ comp network that interconnects users with comp resources in a geographic region of the size of a metropolitan area (city). (small)

→ MAN devices: ① Switches / hubs
② routers / bridges

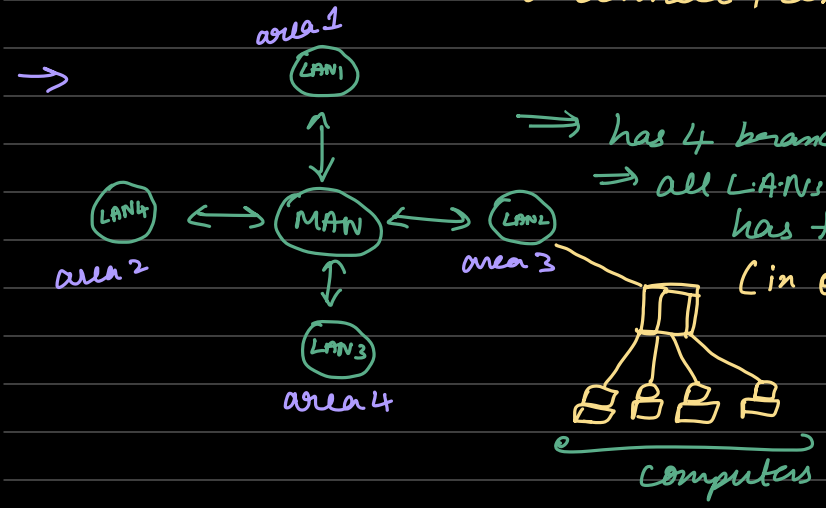
ex: there are 2 branches (LANs) in a city & these 2 branches are somehow connected to each other.

→ 5 km – 50 km

→ any speed, high cost.

To establish a L·A·N

to connect / connec b/w 2 branches



→ has 4 branches / L·A·Ns.

→ all L·A·Ns communicate with each other and has to be connected with each other. (in every L·A·N there are computers & these comps can communicate with other comps in a diff L·A·N)

③ WAN: ex: state to state in a country, (>50km)

→ wide area network.

→ telecommunication network that extends over a large geographical area mainly for comp networking.

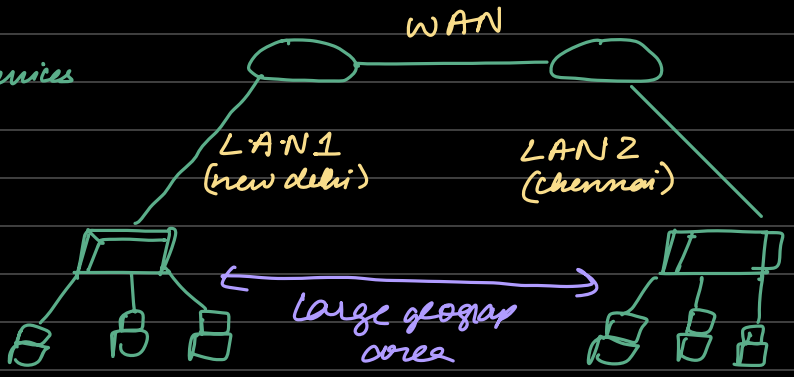
→ any communication at a distant is it as telecommunication.

→ 2 or more L·A·Ns can communicate with each other but in a larger area.

→ WAN devices: ① end devices
② intermediary devices

→ low speed, high cost.

→ most common WAN ⇒ Internet.



④ Internet:

→ a large WAN, public W.A.N.

→ country to country, connects all comps together worldwide.

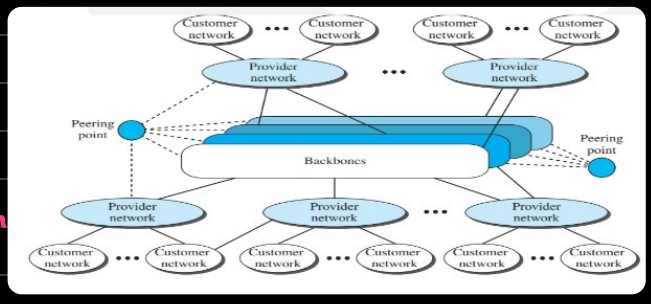
→ many routers & protocols.

→ wireless technology

→ cloud computing, IOT used to deliver services.
(data storage servers) (exchange data over internet)

The Internet:

→ internet with lowercase 'i' is two or more networks that can communicate with each other.
→ Internet with uppercase 'I' is thousands of interconnected networks.
→ The internet has several backbones over wider networks and customer networks.
→ Backbones → Top level → owned by some com companies. Backbone networks are connected through some complex switching system called peering points.
→ second level → smaller networks called provider networks that use the services of the backbones for a fee.



→ the provider networks are connected to the backbones and other provider networks.
→ customer networks are edge of internet that use the services of internet.
→ They pay fee to the provider networks for receiving services.
→ Backbone and provider networks are a.k.a ISPs ⇒ Internet service Providers
→ Backbones are referred as International ISPs.
→ Provider networks are referred as national / regional ISPs.

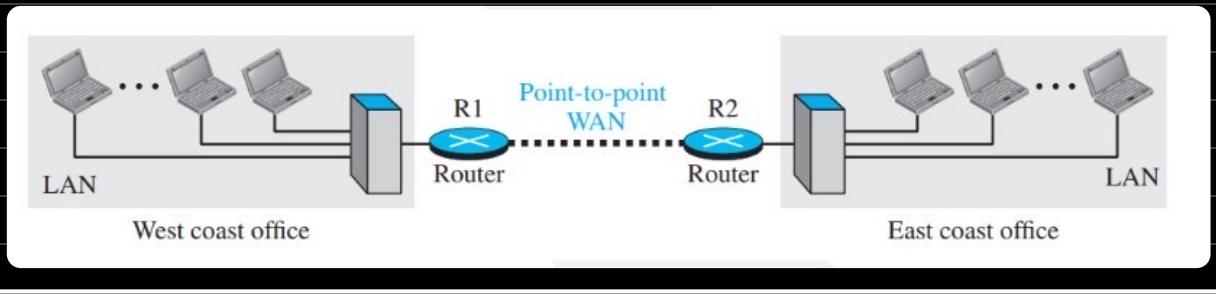
How to access the internet?

1. Telephone networks:
→ most telephone networks are already connected to the internet.
→ One option for the residences & small businesses is to connect to the internet by changing the voice-line b/w the residence or business & the telephone center to a point to point WAN.
2. Cable networks:
→ cable companies have been upgrading their cable networks and connecting to the internet.
→ A residence / small business can connect to the int by this service.
3. wireless networks:
→ ~ ~ ~ by a wireless WAN that is growing rapidly.
4. Direct connection to the internet:
→ large organization / corporation can itself become a local ISP. and be connected to the internet.
→ Org^n / Corp can lease a high-speed WAN from a carrier service provider and connects itself to a regional ISP.
Eg: Large university with many campuses can create its own internet work & connect to the internet.

Internetwork:

→ It is rare to see LAN & WAN in isolation in todays world.
→ they are connected to one another.
→ when 2 or more networks are connected to one another they make an internetwork/internet.

Ex: Assume an orgn has 2 offices. One in east coast and one in west coast. Each office has a LAN through which the employees can communicate with each other. To make the comm b/w employees at diff offices possible, the management leases a point to point dedicated WAN from a service provider such as a telephone company and connects two LANS.
Now the company has a private network/internetwork.



Network models:

O.S.I:

→ provides comm b/w diff hosts.
→ developed by Iso: International orgn of standardization
→ each layer is expected to provide services to the layer above it.

- Application Layer
- Presentation layer
- Session layer
- Transport layer
- internet layer
- Data-link layer
- Physical layer

→ Two comps are conn' to each other with LAN cables and are sharing data with a NIC (network interface card) forms a network. But if one comp has windows and the other has MAC, how will they communicate?

In order to provide efficient comm b/w comps and network for diff architectures we have OSI model.

① Application layer:

→ used by network applications that use internet ex: web browsers.
→ allows diff ways to enable any user to access the network with ease.
(↔ same as TCP ~~)
→ Protocols: (used for functioning of this layer)

1. HTTP: hypertext transfer protocol.
   client server protocol.
   comm b/w web browsers & web servers.
2. HTTPS: ~~~~ secure.
   extension of HTTP.
   uses encryption to secure comm over a comp network.
3. DNS: domain name system aka phonebook of internet.
   provides naming sys for devices, services.
   Translates the domain name (selected by user) into the corresponding IP addresses.
   Ex: www.something.com ⇒ 192.36.10.8
4. FTP: file transfer protocol.
   helps to transfer diff files from 1 device to another.
   reliable, efficient data transfer, remote devices.
   Built on client server model ⇒ transf comp files from a comp network to a client.
5. SMTP: simple mail transfer protocol.
   to transfer mail from one user to another with ease.

② Presentation layer:

→ receives data from application layer in characters. This layer converts the characters to Binary form for machine to understand.
→ aka data translator for the network
→ maintains proper syntax of data and encryption too.
Func: ① Translation: as above
② Data compression: reduces no. of bits of data for easy flow.
③ encryption: provides protection of data by protocols
Ex: SSL ⇒ secure socket layer ⇒ encrypts the link b/w browser & server
⇒ provides security to data being transferred b/w web. browser & the server and the data is private.

③ session layer:

→ manages sessions b/w applications/ safe connection is maintained
→ helps in managing connections.
→ avoids data loss and maintains synchronization, terminates session etc
Ex: APIs ⇒ helpers of this layer
NETBIOS ⇒ input/output system.

funcy

① → authentication } performed by this layer
② → authorization }
③ → tracks data packets ⇒ session management (makes sure it reaches the right dest)

ex: while opening the comp

ex: when we open image/text file the particular data packet for image & text is provided.

④ Transport Layer:

→ provides end to end comm for applications.
→ establishes, maintains, terminates connections b/w devices
→ correct sequence, error detection, flow control.

ex: TCP, UDP, SCTP.

→ funcy: ① segmentation (data recd from session layer is div into segments. (small data units)
② flow control (amt of data being transmitted to prevent loss of data)
③ error control (checks for errors in the message coming from application layer & checks for integrity of data)

⇒ each data unit consists a port no. & a sequence no.
⇒ port no. directs each segment to its application (ex: w measure browser etc)
⇒ seq no. ⇒ correct sequence of data is maintained.

⇒ protocols: transfer control protocol ↑ user datagram protocol. ↑

T.C.P	U.D.P
→ reliable, connec are verified	→ unreliable, connec are un-verified.
→ connection oriented protocol	→ connectionless protocol.
→ speed is lower	→ speed is higher.
→ data packets are arranged in order.	→ data packets are unordered.
→ transfers segments.	→ transfers datagrams.
→ provides feedback / ack.	→ no feedback / ack.
→ has to establish a connec b4 data transfer.	→ detects errors but does not specify which error.
→ used for web browsing, mailing, file transfers.	→ no need to establish a connec b4 data transfer.
→ used for reliable but slow transmission.	→ used in games, video conferencing.
	→ used for only its speed, reliability does not matter.

→ datagrams carry less info since they dont need to have a response message from the destination. (contains enough info to be routed from source to dest.

③ Network layer:

→ manages logical addressing.

→ logical addressing is done by network layer where the packets (contain IP address of sender & receiver) is assigned to each segment, entirely called as data packet.

→ tracks location of devices in the network.

→ manages routing of data packets across multiple networks.

→ determines best physical path for data transfer from source → dest.

ex: IPv4, IPv6, ICMP etc

func → ① packetizing: (dividing the data from upper layer / encapsulating the data)

② routing: (moving data from one device to another device. The network layer finds the best route)

③ forwarding: (applied by each router when a packet arrives, when a router receives a packet from an attached network it forwards it to another attached network)

note: divides data at source ⇒ encapsulating.
reassranging data at receiver end ⇒ decapsulating.

* protocols:

① IP ⇒ int protocol ⇒ delivers packets from source to destination with the help of IP addresses.
ex ⇒ IPv4 (efficient)
IPv6
⇒ it is a unique address to identify the packet and for routing the packet.

② ICMP ⇒ int control message protocol.
⇒ used to check & sends notifications back to sender regarding datagram problems for any probs faced by receiver.
⇒ a datagram travels router to router & if a router is unable or device is damaged, the datagram is not delivered & ICMP protocol is used to inform the sender.

③ ARP ⇒ address resolution protocol.
⇒ used to find the physical address from the IP address.

④ RARP ⇒ used to find IP address from physical address
(reverse)

Transport Layer	Network Layer
Responsible to send entire message from a host to a destination	Responsible to send packets from a host to a destination
It's process-to-process communication or port-to-port communication	It's host-to-host communication
Used inside of same network and different networks as well	Used when the hosts are in different networks
Uses the port address to ensure the communication	Uses logical address ensure for the communication
~~Implemented on host machine~~	~~Implemented on networking devices such as routers and switches~~
Provide better flow control and error control	Flow control and error control is not as good as the transport layer

① Data link layer: receives packets from network layer.

→ manages physical addressing.

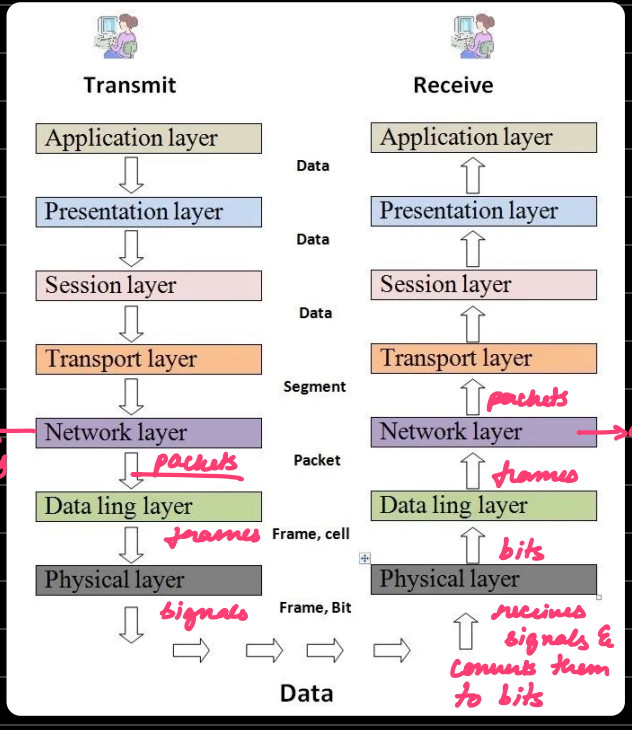
→ In physical addressing (by data link layer), MAC address of the sender & receiver are assigned to each data packet, entirely known as a frame.

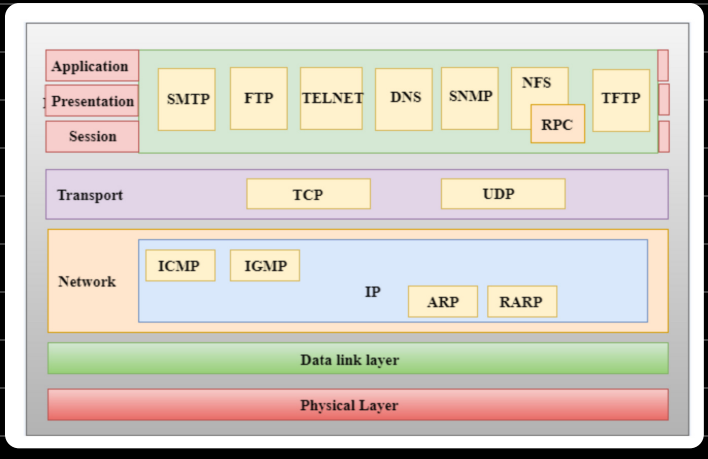
→ provides reliable data transfer across the phy network.
→ handles error detection & correction
→ The unit of data-link layer is the frame sequence of binary 0's & 1's)
→ func: framing (explained further)

① Physical layer.

→ lowest layer.
→ Till now the data from the application layer has been segmented by the Transport layer, placed into packets by Network layer and framed by data-link layer which is a sequence of Binary 0's & 1s. The physical layer converts this sequence into Signals and transmits over a local media.
→ consists of data connection b/w a device generating data & the network.
→ phy conn: b/w devices.
→ defines the hardware elements involved ⇒ cables, switches etc
ex: ethernet, hubs etc.

finally at the receiver: phy layer receives signals, converts it to bits and pass it to data link layer at the frame ⇒ application layer. the app layer makes the sent's message visible in the application in the receiver's comp screen.





T.C.P/I/P model:

→ Transmission control protocol.
→ used on computers today.
→ advantages:
* helps to establish a connⁿ b/w 2 diff computers/diff O.Sys.
* uses client – server architecture.
* lightweight.
→ disadvantages:
* complicated to setup.
* ~~~~ to manage.
* Transport layer does not gaurantee packet delivery.

L5 Application layer

L4 Transport layer

L3 network layer

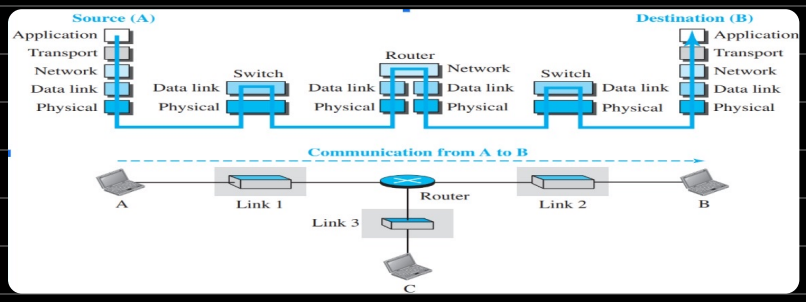
network access layer } data link layer L2, phy layer L1

→ in this model the network layer is the combination of physical and data link layer and the app layer is combined with presentation and session layer.

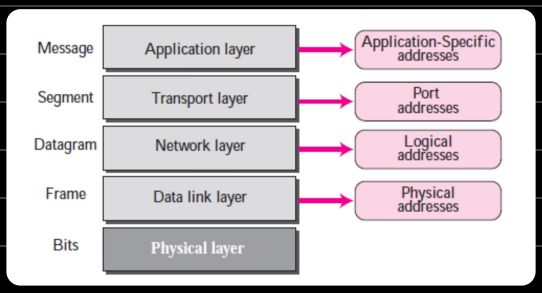
OSI	TCP/IP
→ Open sys interconnection. → reference model. → developed by ISO. → vertical approach. → reliable. → 7 layers (sess + presentation layers are seperate) → protocol independent standard. → supports both connⁿ oriented & connⁿless comm. → Transport layer provides assurance of delivery of packets.	→ Transmission control protocol. → implementation of OSI model. → developed by ARPANET. → horizontal approach. → less reliable/ reference tool. → 4 layers. → protocol dependent standard → supports only connⁿless comm. → Transport layer does not provide assurance of delivery.

① Network Access Layer: * Lowest layer

* comb of phy & data link layer.
* defines how data should be sent physically through the network. (b/w 2 devices in same network.
* controls the hardware devices & media that make up the network.
* protocols: PPP, Frame Relay, Ethernet.
  ↳ eg: Wi-Fi, ethernet.



Path of data



addressing

Addressing:

1. physical address (link / MAC): to identify devices on same local network.
2. logical address (IP): to identify device globally.
3. port address: assigned uniquely to identify conn^ec endpoint.
4. application-specific address: user friendly add designed for specific add. eg: email addrs

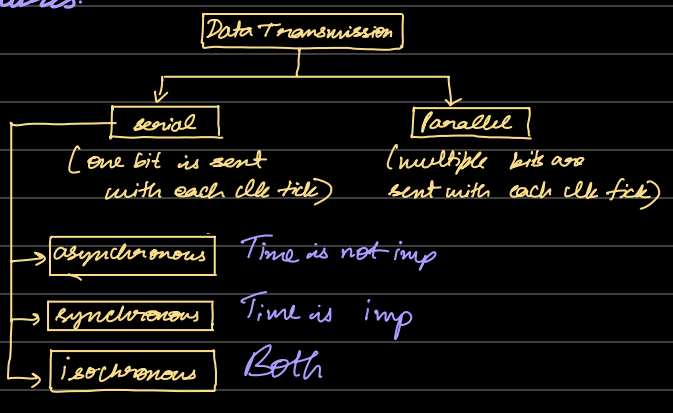
MAC ADDRESS	IP ADDRESS
Layer 2 address	Layer 3 address
Identifies network devices on a local scale	Controls how devices on the internet communicate on a global scale
12 digits, grouped into six pairs, separated by hyphens Example: 00-00-00-00-00-00	For IPv4: 32 bits, grouped into four decimal numbers Example: 000.000.000.000 For IPv6: 128 bits, grouped into eight sets of four digits Example: FEDC:BA98:7654:3210:0123:4567:89AB:CDEF
Can't be changed	Can be changed at any time
Sometimes called physical address	Sometimes called logical address
Hardcoded into the device at manufacturing	Assigned to device through software configurations

phy add is defined by its LAN or WAN. It is included in the frame used by D.L.L

Transmission Modes:

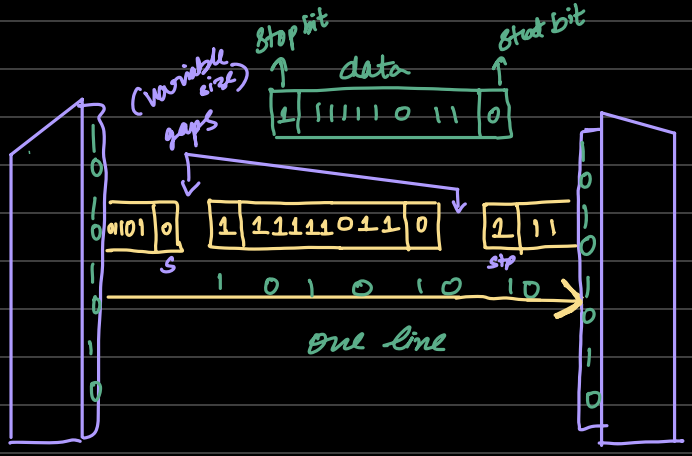
⇒ Transmission modes are used for sending streams of data containing 0s & 1s. from one device to another over wire / wires.

⇒ ① Serial mode
② Parallel mode.



Serial

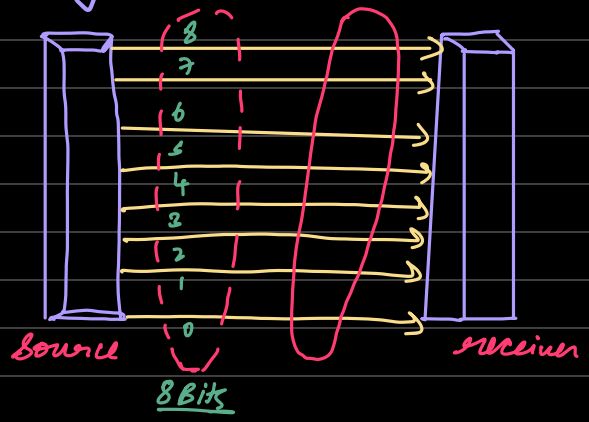
⇒ one bit flows at one clk pulse.
⇒ 8 bits are transferred at a time having a start & a stop bit.
⇒ only one comm channel is req.



⇒ suitable for long distances.
⇒ disad: low speed.
⇒ ad: cost.
⇒ slower than ||el transmission
⇒ ex: comm through telephone line.

Parallel

⇒ multiple bits flows at one clk pulse.
⇒ Binary data is dvn. into groups of n bits each | 8 bits each and sent at a time.
⇒ requires n/8 wires.
⇒ many comm channels are req



⇒ suitable for short distances
⇒ disad: ① cost of comm lines / wires. ② limited to short dists
⇒ ad: speed of transfer.
ex: comm from comp to printer.

---

① Asynchronous:
Time is not imp
sends every chara^c / byte with START & STOP bit
⇒ 0 ⇒ start bit.
⇒ alerts receiver that data is arriving.
⇒ 1 ⇒ stop bit.
⇒ alerts receiver that byte is finished.
⇒ gaps b/w data bytes i.e gap b/w comm. Therefore it is asynchronous
⇒ gaps are variable time interval b/w data units.
⇒ used for low speed transmission / communication.
⇒ Characteristics:
1. Timing: each byte has its own start/stop. Gaps: slower transmission. No continuous stream of data/ irregular
2. Synchronization: receiver synchronizes with the sender at start of each byte.
3. Overhead: high overhead due to additional start/stop bits for each byte ⇒ slower response.
4. data integrity: each byte is correctly framed with start & stop bits

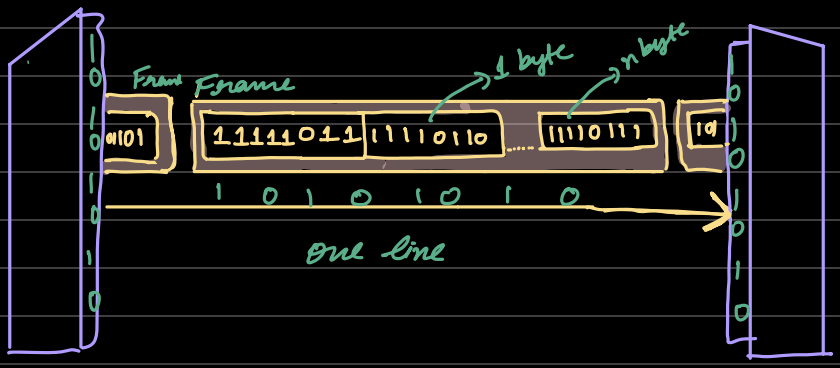
8 bits = 1 byte

ex: * keyboards
* slow speed voice channel.
* sensors.
* RS-232 //

② Synchronous:

⇒ time is imp.

⇒ high speed transmission.



⇒ no gaps b/w bytes.

⇒ Characteristics:

① Timing: data is sent in continuous stream, synchronized by a shared clock signal b/w sender & receiver

② Synchronization: Both sender & receiver use the clk signal to stay in sync, ensuring precise timing.

③ Data integrity: Synchronization allows efficient / reliable data transfer.

④ Overhead: less overhead as no start & stop bits are needed.

⇒ applications:

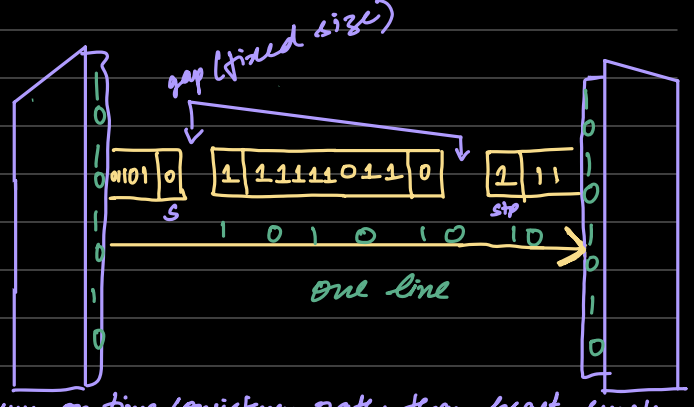
* used in high speed communication systems where continuous data flow is req.
* ethernet
* fibre optics
* telecommunication protocols.

③ Isochronous:

⇒ has characteristics of both sync & async transmission

⇒ The gaps are of fixed size & not variable.

⇒ Chars:



① Timing: data is sent at regular intervals / consistent timing.

② Synchronization: combines aspects of both sync & async, focuses on time consistency rather than exact synchro-nization.

③ Data integrity: ensures data is delivered at fixed rate / consistent time intervals.

④ Overhead: similar to sync but designed for time sensitive data.

⇒ applications:

* used in real time comm systems where time-sensitive info is delivered.
* audio & video streaming.
* industrial control systems.

Asynchronous	Synchronous
1. Data is transmitted character-by-character.	1. Data is transmitted block-by-block or word-by-word.
2. Data is not saved before transmitting.	2. Data is saved before transmitting.
3. Slower data transmission.	3. Faster data transmission.
4. Few characters can be transmitted at a time.	4. A large volume of data can be transmitted at a time.
5. There may be gaps between characters.	5. There are no gaps between blocks.
6. It uses start and stop bits to control data transmission.	6. It uses clock signals to control data transmission.
7. It is used in large organizations.	7. It is useful in small offices.
8. It is cheap.	8. It is expensive.

* Data Link Layer:

⇒ Data link Control: (DLC)

→ receives packets from network layer.
→ manages physical addressing.
→ logical addressing is done by network layer.
where the packets (contain IP address of sender & receiver) is assigned to each segment, entirely called as data packet. [IP1 | IP2 | segment]
→ In physical addressing (by data link layer), MAC address of the sender & receiver is assigned to each data packet, entirely known as a frame.
⇒ provides reliable data transfer across the phy network.
→ The unit of data-link layer is the frame. This layer is embedded as a software in network interface card (NIC) of the computer and provides means to transfer data from one comp to another via a local media.
→ local media includes copper wire, optical fibre or AIR etc. (media)

⇒ data link layer is responsible for moving frames from one node to another node.

⇒ Data Link Services:

1. framing
2. Phy addressing
3. flow control
4. error control
5. access control

1. Framing:

⇒ data unit of data link layer ⇒ frame
⇒ the data link layer needs to pack bits to frames so that each frame is distinguishable from the other.
⇒ it separates a message from one source to dest or from other messages to other destinations
⇒ The frame that is to be transmitted consists of data fields, header field.
⇒ framing is done by adding a sender address, dest address (for delivery), (for acknowledgement)

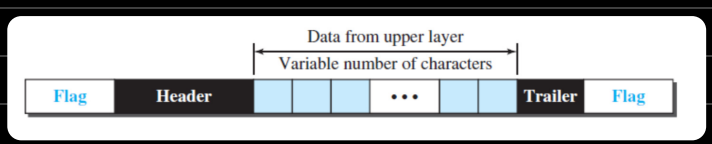
⇒ At the sender's side,
the DLL receives packets from network layer & divides them into small frames and sends each frame bit-by-bit to the phy layer. It also adds extra bits for error control etc at header & end of the frame

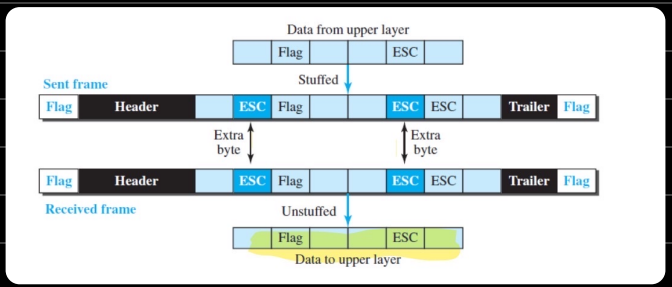
⇒ At receivers side,
the DLL takes bits from the phy layer & organizes them into the frame & sends them to the network layer

⇒ framing can be fixed-size or variable size.
⇒ in fixed-size framing,
* frame sizes are fixed and the frame size also acts as the delimiter of the frame ex: after receiving a frame of 100 bits the receiver can understand that frame 1 has ended and next 100 bits is frame 2.
* it does not require start & stop bits i.e boundary bits to identify when a frame ends & starts.

ex: asynchronous transfer mode.
⇒ in variable size framing:
* size of each frame to be transmitted is different/length is not uniform.
* start & stop boundary bits are required.
* has 2 protocols or approaches : Bit oriented protocols, Chara' (Byte) oriented protocols.

① Chari-oriented approach/Byte oriented :
* data is transmitted as a sequence of bytes i.e 8 bits in ASCII form or multimedia form.
* Parts of frame:
  (i) Frame header : contains source & dest addresses of frame.
  (ii) Trailer: contains bytes for error detection & error correction.
  (iii) flags : frame delimiters signalling the start & end of frame.
  (iv) payload field : message (data) to be delivered is present here
* this protocol is suitable for transmission of texts.
* There are chances that the pattern of the flag is present in the data field too. How will the receiver recognise it then?
→ Here, byte stuffing mechanism is used so that the receiver does not interpret the pattern as the end of the frame. (aka chara' stuffing)
→ A special byte called escape byte (ESC) is stuffed before every byte in the message with same pattern as the flag bytes aka ESC sequence.
→ If the ESC sequence is present in the message byte (again), another ESC byte is stuffed b4 it.
→ drawback: large overhead on the message, increase the total size of frame. or it leads to error.





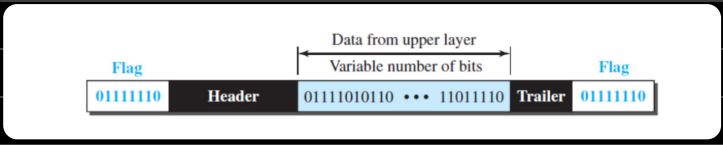
① Bit - oriented approach:

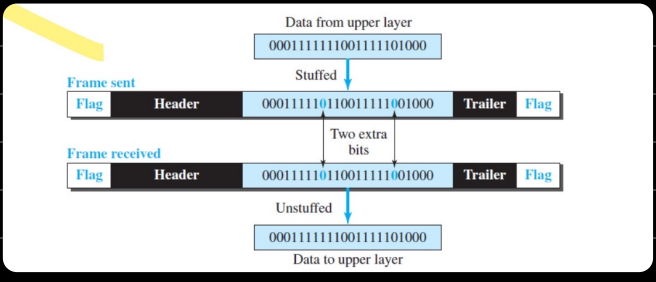
* data is transmitted as a sequence of bits that can be interpreted as text as well as multimedia.
* This protocol uses a special 8-bit pattern. 01111110 as flag.
* parts:

① header : contains source & dest addresses of frame.
② Payload field: message (data) to be delivered is present here
③ Trailer: contains bytes for error detection & error correction.
④ flags: start & stop flags. It will be an 8 bit sequence with 6 or more consecutive 1s. Most protocols use the 8 bit pattern 01111110 as flag.

* Suitable of transmitting sequence of bits.
* There are chances that the pattern of flag is present in the message/data.

→ For the receiver not to consider this as the end of frame, bit-stuffing is done.
→ whenever a 0 bit is followed by 5 consecutive 1s, an extra '0' bit is stuffed at the end of 5 1s.
→ when the receiver receives data, it removes the stuffed 0s after each sequence of five 1s.
→ The unstuffed message is then sent to the upper layers.





Q) assuming a framing protocol that uses bit stuffing, show the bit sequence transmitted over the link when a frame contains the following:
1 1 0 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0
mark the stuffed bits

---

② Flow Control:

⇒ There is limited amt of packet buffering capacity on each side of the link
⇒ receiving nodes receives the frames faster than that can process the frames.
⇒ as a result of this, without any flow control, the buffer on the receiving end will overflow and some of the frames can get lost.
⇒ In order to solve this, the DLL uses flow control mechanisms which prevents the sending node from overwhelming the receiving node with the frames on other end.
⇒ The DLL synchronizes the sender's & receiver's speeds & establish flow control b/w them.

③ Error control:

⇒ errors can be introduced very easily while transmitting the frame from one node to another.
⇒ These errors are introduced either by noise / signal attenuation.
⇒ the receiving node can incorrectly recognise a 0 as 1, vice versa.
⇒ In this case, there is no need to send datagrams that have errors.
⇒ To resolve this issue, the DLL provides error detection that is done by adding error control bits at the transmitting node. The receiver performs error check on it.
⇒ The receiving node not only determine if the error has been introduced or not in the receiving frame but also determine where exactly in the frame the error has been introduced & performs error correction. Uses protocols like ATM etc.
⇒ CRC ⇒ Cyclic redundancy check is added to the frame header and it is checked by the receiver. (by the sender)
⇒ Two things can happen:
1. if frame is corrupted it is silently discarded, if not it is delivered to the network layer. used in ethernet, wired LANs.
2. if frame is corrupted it is silently discarded, if not, receiver sends an ack to the sender.

---

④ Access Control:

⇒ When multiple devices use the same comm channel, there is a high prob of collision
⇒ DLL checks which device has control over the channel & CSMA/CD and CSMA/CA can be used to avoid collisions & loss of frames.

---

⑤ Phy addressing:

⇒ MAC address is the unique hardware address that is assigned to the device while manufacturing.
⇒ (explained in the DLC in the beginning)

⇒ D·L·L protocols:

1. Connectionless Protocol:

→ frames are sent from one node to another without any relationship b/w the frames.
→ each frame is independent.
→ connectionless does not mean phy conns; it means they are related to one another.
→ frames are not numbered.
→ no sense of ordering
Ex: LANs.

2. Connection oriented protocols:

→ logical conn' must be established first b/w 2 nodes (setup phase).
→ frames related to each other are transmitted (transfer phase)
→ logical connⁿ is terminated. (teardown phase).
→ frames are numbered.
→ frames are sent in order.
Ex: PPP, WANs, wireless LANs.

⇒ Types of DLL protocols: ③

→ we use FSM (finite state machine) to represent the DLL protocols. It contains finite no. of states and the machine is always in one of the states until the event occurs.

1. Simple Protocol:

→ neither flow / error control.
→ we assume that the receiver can immediately handle any frame it receives.
→ DLL gets a packet from Network Layer, makes a frame out of it, sends it to the receiver and
→ at receiver end, the DLL receives frame, extracts packet, delivers it to network layer.

FSM:

Sender should not send a frame until its network layer sends data.
receiver cannot deliver until a frame arrives.
each FSM has only 1 state ⇒ ready state.

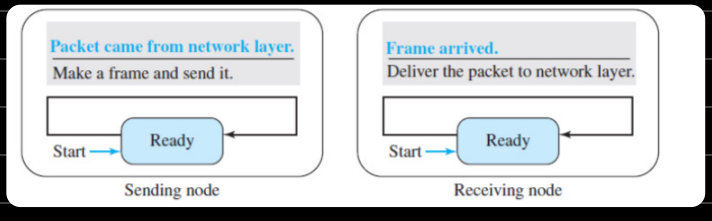
Sender machine:

remains in ready state until message arrives from upper layer (network layer)
once it arrives, sender encapsulates the message in a frame & sends it to receiver.

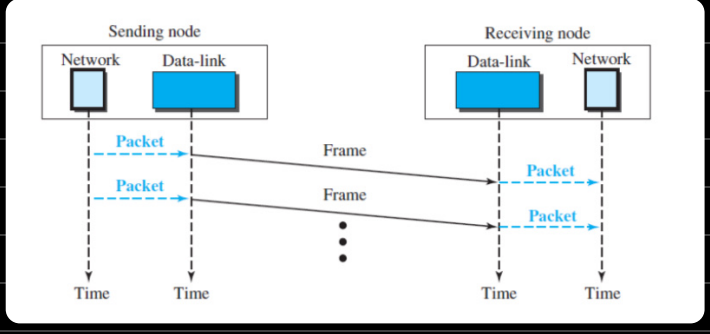
Receiver machine:

remains in ready state until frame arrives.
once it arrives, it decapsulates the message out of the frame & sends it to upper layer.

FSM



Flow diagram



2. Stop and wait protocol:

→ uses both flow, error control

→ sender sends one frame at a time & waits for ack. For error detection, we add a CRC to each packet of data. If CRC is incorrect, data packet is silently discarded (error). If not, it sends an ack. The silence from the receiver helps the sender to figure out that the data was corrupted.

→ So, everytime the sender sends a frame, it starts a timer. If an ack arrives b4 timer expires, the sender discards the copy of the frame and sends the next frame. If timer expires with no ack, the sender restarts the timer and resends the saved copy of the frame.

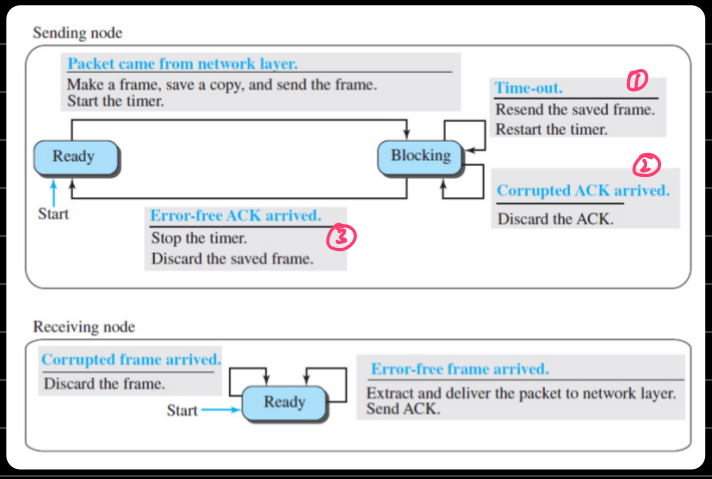
FSM:

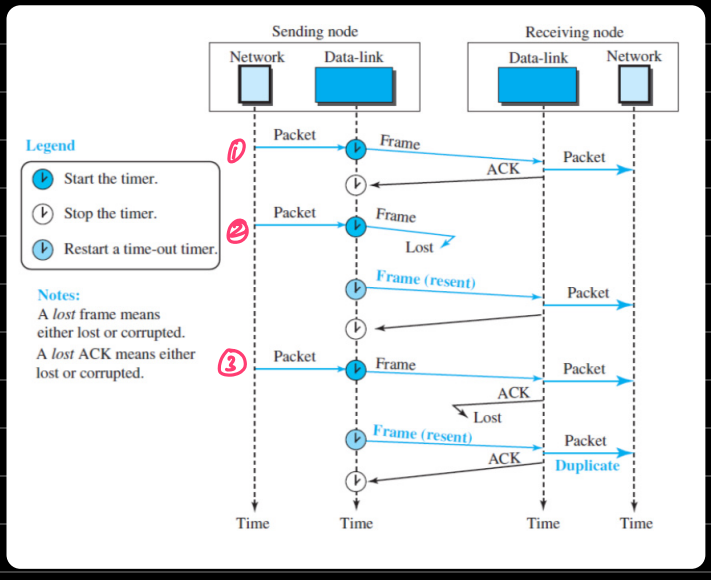
Sender:

1. ready state: waiting for packet from network layer.
   Once it arrives, makes a frame, saves a copy of frame, starts timer. sends frame to receiver
2. Blocking state: three events may occur:
   * corrupted ACK arrives, discards ack.
   * error-free ACK arrives, stops timer, discards saved copy, sends next pkt.
   * time-out occurs. restarts timer, resends saved copy of frame.

Receiver:

Ready state: two events may occur:
* error-free frame arrives, message in frame is delivered to upper lay.
* Corrupted frame arrives, frame is discarded.





Disadvantages:

→ low efficiency (has to wait for ack)

→ limited throughput (due to stop & wait)

→ high error rates.

→ overhead due to ACK.

→ not suitable for large data sizes as it is slow.

→ complex re-transmissions

Advantages:

→ sequence control by seq number.

→ error detection/error recovery.

→ flow control

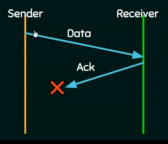
→ reliable data transfer.

→ useful for simple networks.

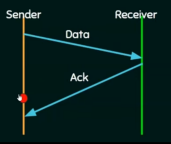
Problems that may occur: (3)

1. Problems due to lost data.
   - ★ Sender waits for ack for an infinite amount of time.
   - ★ Receiver waits for data an infinite amount of time.

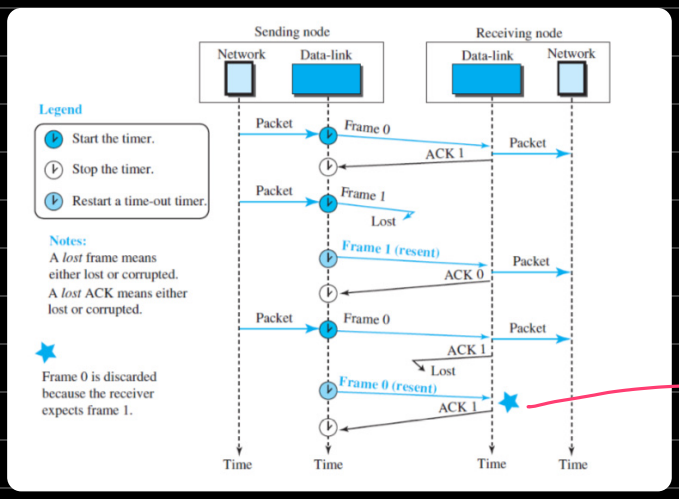
2. Problems due to lost ACK.
   - ★ Sender waits for an infinite amount of time for ack.



3. Problems due to delayed ACK/data.
   - ★ After timeout on sender side, a delayed ack might be wrongly considered as ack of some other data packet.



→ These problems can be avoided by sequence numbering and ACK numbering.
→ seq numbers starting with 0, 1, 0, 1...
→ ack numbers starting with 1, 0, 1, 0...
→ this can prevent duplicates too.

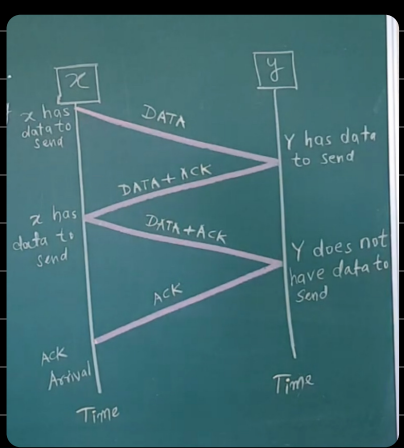


→ duplicate is not created as the recive discards the frame as it is waiting for frame 1 and not 0.

---

3. Piggybacking:

→ these two were unidirectional protocols. (data flows in one direc, ack in the other direc).
→ piggybacking is a bi-directional protocol.
→ makes comm more efficient.
→ data in one direc is piggybacked with the ack in the other direc.
→ when A is sending data to B, node A also ack the data received from B.
→ not a common practice as it makes data transfer more complicated



HDLC: aka SDLC (synchronous) (uses stop & wait protocol)

⇒ high level data link control (HDLC).
⇒ It is a bit oriented synchronous data link layer protocol.
⇒ It is an implementation of stop & wait protocol.

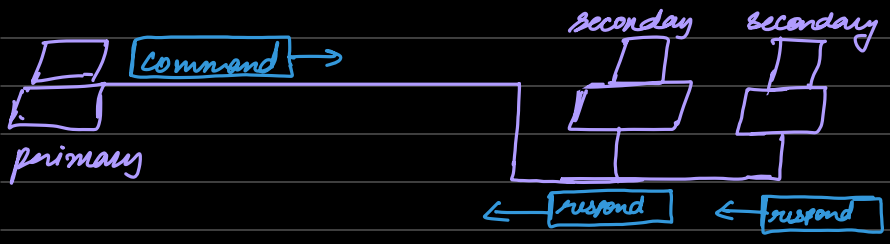
HDLC configuration / transfer modes:

① NRM → normal response mode.
② ABM → asynchronous balanced mode.

⇒ HDLC has 3 stations:
① Pri ⇒ sending data, controls flow.
② Sec ⇒ receiving data
③ Combined ⇒ Both features

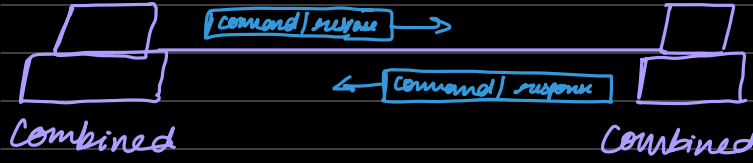
① NRM: ⇒ station config is unbalanced.
⇒ consists of a primary station and a secondary station.
primary station: sends commands; secondary station: responds to the received commands.
⇒ used for both PPP and multipoint comms

(i) PPP comm



(ii) Multipoint comm

② ABM: Station config is balanced.
⇒ each station can both send & respond to commands.
⇒ only used for PPP comm.



HDLC framing: ③

① I-frames ⇒ used for transport of user data & related control info. (information frame)
⇒ in the control field, 1st bit is 0.
② S-frames ⇒ used for transporting only control info. (supervisory frame)
⇒ 1st 2 bits are 10
③ U-frames ⇒ used for carrying info managing the link itself (un-numbered frame)
⇒ 1st 2 bits are 11. and sys management.

⇒ generally,

start	header	data	crc	stop
8 bits	16 bits		16 bits	8 bits

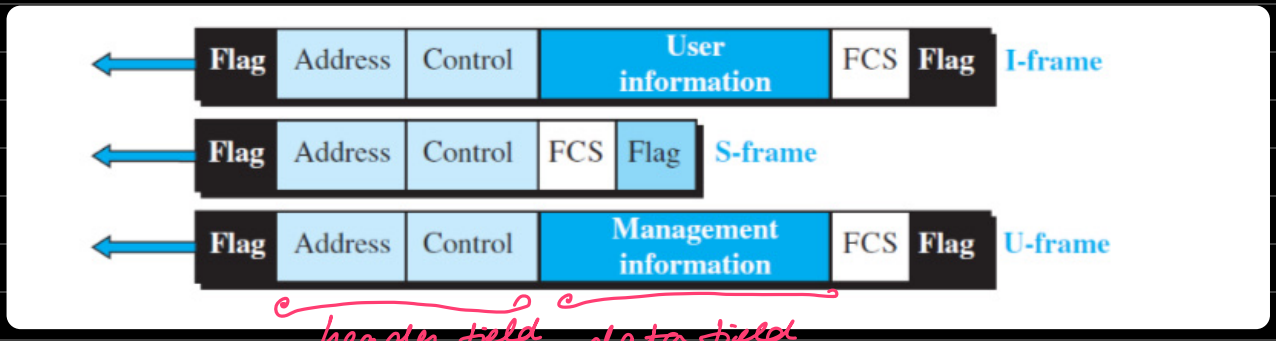
* flag ⇒ 8 bit sequence ⇒ 01111110
  ⇒ to identify the start & end of the frame.
  ⇒ serves as synchronization pattern for the receiver.

(if sender, it contains to add on
if recier, it has from add)

* Header ⇒ consists the address field & control field.
address field ⇒ address field contains address of the secondary station.
control field ⇒ control field is a 1/2 byte segment of the frame used
⇒ used for flow & error control

* Body ⇒ Payload (variable size), user info or management info.

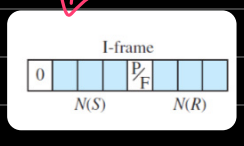
* FCS ⇒ frame check sequence ⇒ HDLC error detection field can
⇒ consits 2 - 4 byte CRC.



① I-Frame:

⇒ carry user info from network layer.
⇒ include flow and error control bits (piggybacked on data)
⇒ first bit is always 0 ⇒ to identify the frame as I frame.

I frame



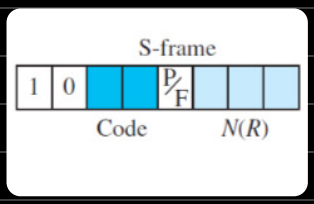
Control field of I frame

* N(S) ⇒ 3 bits ⇒ defines sequence no. of frame
* N(R) ⇒ 3 bits ⇒ defines acknowledgment no. [ack no. = next frame no. i.e next seq no.)
* P/F ⇒ 1 bit ⇒ poll or final

↳ if poll=1 ⇒ frame is going from primary to secondary.
if poll=0 ⇒ ~~ sec ~~ primary ↲

② S-frame:

⇒ no info field.
⇒ first two bits are 10 always

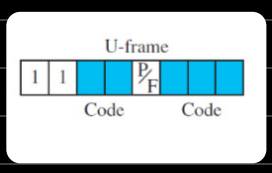


* N(R): 3 bits ⇒ acknowledgement no.
* Code: 2 bits ⇒ 4 types of S-frames are possible:

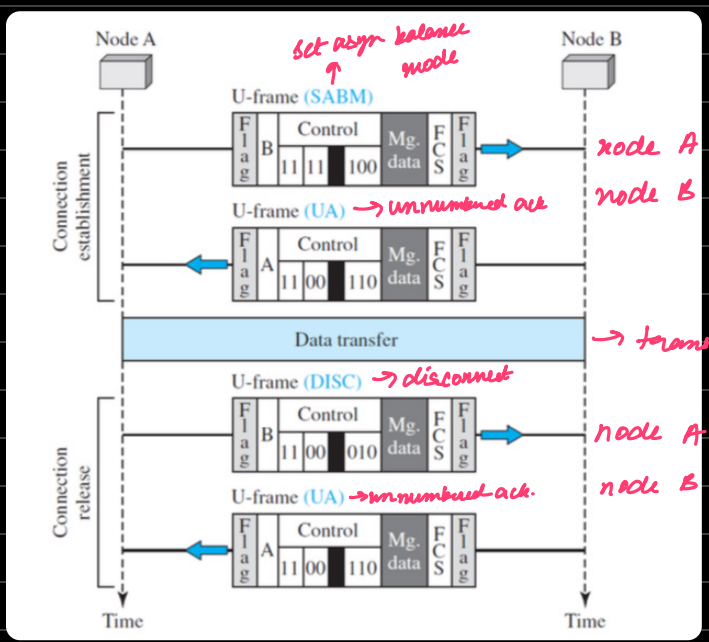
00 ⇒ RR ⇒ receive ready. receiver is ready to accept frames.
01 ⇒ REJ ⇒ reject frames. informs sender that frame is corrupted / damaged bf sender's timer expires. (improves efficiency)
10 ⇒ RNR ⇒ receive not ready. receiver buffer is full & it can't accept frames at the moment.
11 ⇒ SREJ ⇒ selective reject. selects a frame to reject from the buffer to accomodate new frame.

③ The U-frame:

⇒ first two bits are 1 1.
⇒ five bits are code. 2 bits prefix P/F & 3 bits suffix P/F.



⇒ 5 bits of code can be used to create 32 types of U-frames.
⇒ U-frames can be used for connec establishment and connec release.

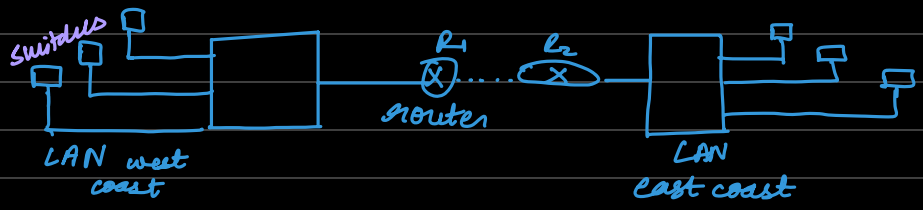


P.P.P: point to point protocol.

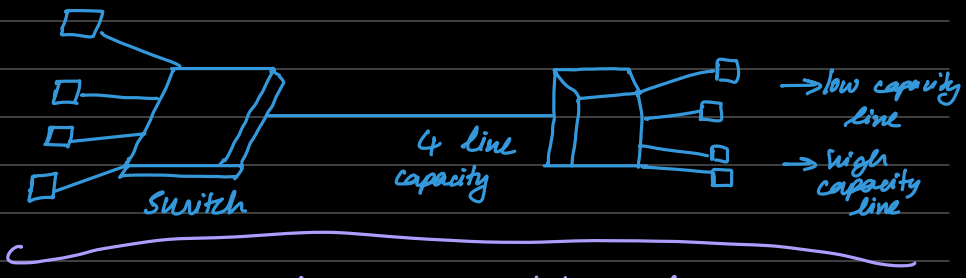
⇒ Char oriented framing. HDLC is bit oriented framing.
⇒ used for point to point access.
Ex: telephone comm.
⇒ most common.

Framing in P.P.P:

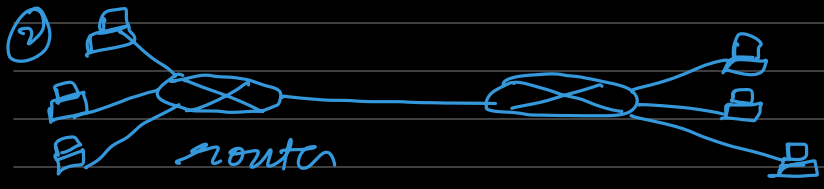
Internetwork:



① Circuit Switched
② Packet Switched
} 2 Types



circuit switch network

② 

packet switch

⇒ stores packet in buffer.
⇒ forwards it to the next system.
} Store & forward method
⇒ router : makes a decision through which line the packet should be sent.
i.e routing decision.
outgoing line
↳ not the case in circuit swi

Internet:

⇒ Dial → Service.
⇒ Cable network
⇒ wireless network.

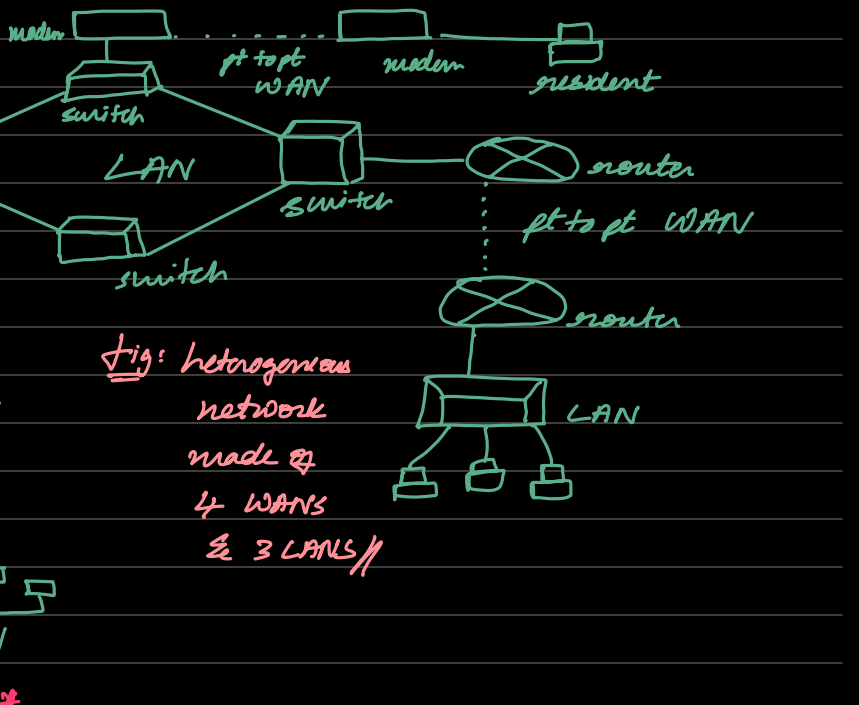


fig: heterogeneous network made of 4 WANs & 3 LANs

↓

* Traditional method.
* already existing lines.
* Eg: Telephone service.
* line is identified by a code i.e digital no.(data)
* line gets active when phone is lifted.

→ *

flow control
TCP
UDP
SCTP

Source A

L5 → HTTP (any website)
SMTP (mail transfer)
FTP (file transfer)
TELNET (to connect to network)
SNMP
DNS (domain name sys)

L4

ICMP ← L3
IGMP
DHCP
ARP

L2

L1

five layers

switch

router (decision maker)

Source B

five layers

another switch