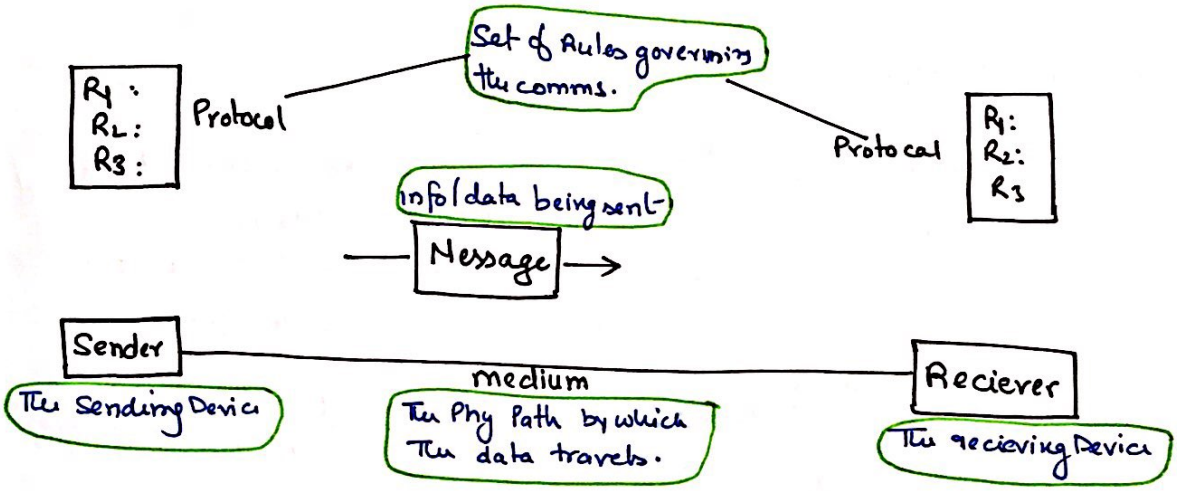
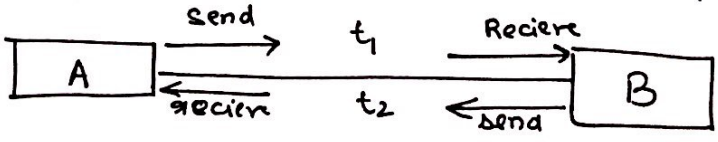
Computer Networks

- **Data Communication** : The exchange of data between two devices via some form of transmission medium. To facilitate this the devices must be connected to data comm system.
  - **Delivery** - system must deliver the data to intended device.
  - **Accuracy** - The data must be delivered without any alterations
  - **Timeliness** - The data must be delivered within an appropriate time.
  - **Jitter** - This is the difference/ variation in packet arrival.



- **Data Flow Models**
  - **Simplex** - The comm is unidirectional. only one device can Transmit.
    keyboards and Monitors are simplex mode.
    The benifit is the device can use the entire bandwidth.

  - **Half Duplex** - Both devices can transmit & reciere but not at the same time.
    Walkie Talkies use Half Duplex mode.
    It has the same benifit as simplex but allows 2 way comms



- **Full Duplex** - 2 sided sending and recieving can happen concurrently.
  This is used in networks that need 2 way comms like telephone
  The bandwidth gets divided here.

• Networks

A network is the interconnection of a set of devices capable of communication. A device can be a Host / connecting device etc. The devices are connected using some form of wired / wireless media.

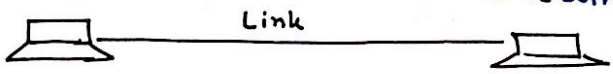
• Network Metrics

1. **Performance** - This can be the speed / transit time / response time / Throughput / delay.
   while the first are are complimentary the later two are contradictory.
2. **Reliability** - This is the frequency of failure of the network The time taken to recover from failure & robustness.
3. **Security** - Protection of data from unauthorised access / damage. Procedures to recover from breaches.

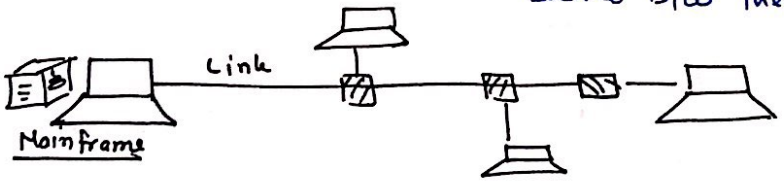
• Network Structure

• Connection Type

1. **Point-to-Point** : There is a dedicated channel between the two devices. The entire capacity is reserved for them.

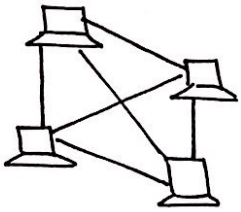


2. **Multipoint/Multidrop** : More than two devices share the channel. It can be spatially or temporally shared b/w them.



- **Topology** - Refers to the way the network is laid out. The devices are called nodes.

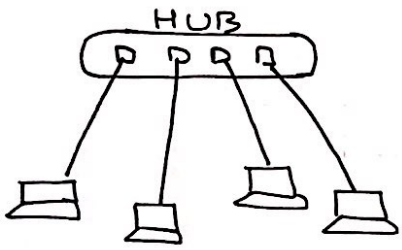
1. **Mesh Topology** : Every device has a point-to-point link with Every other device.
   $n(n-1)/2$ duplex connections are needed.



Advantages :
- No Traffic issues
- Robust, one link failing isn't a issue
- secure
- Easy to pin-point faults

Drawbacks
- Requires alot of cable.
- Requires an unreasonable amount of I/O ports.

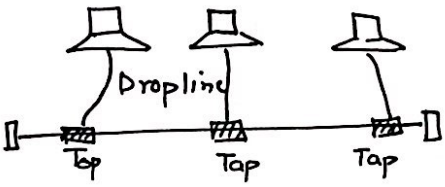
2. **Star Topology** : Every device has a p-2-p link with a central hub.
   The Hub relays the info b/w the devices.



Advantages :
- Robust, one link failing does'nt effect the net
- Less Expensive
- Easy to locate faults.

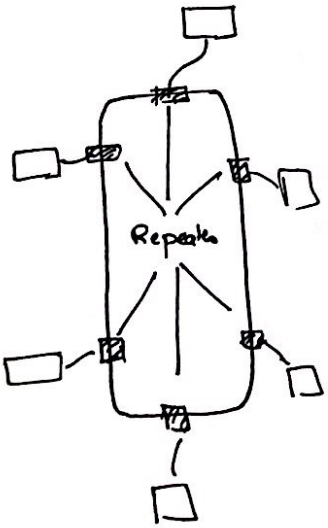
Drawbacks
- If the Hub fails, all the net goes down.

3. **Bus Topology** : This uses a multidrop connection the devices are connected via drop line and taps to the backbone line.



Advantages
- Easy Installation by optimizing the backbone length.

Disadvantages
- Diffult fault location & isolation
- Degradation in signal due to reflection @ taps
- Any break/fault in b.b stops all transmission.

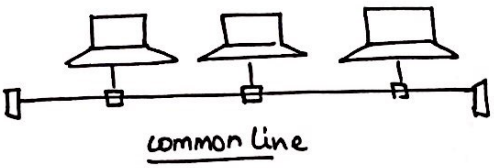


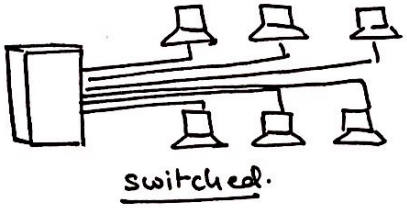
4. **Ring Topology** : Each device has a P-2-P link with its 2 neighbours The signal is passed device to device till it reaches the destination.

Advantages
- Easy to install & Reconfigure.
- Easy fault isolation

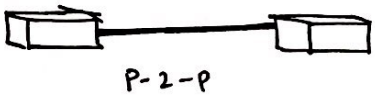
Drawbacks
- Unidectional flow will cause break in data if there a break in the ring.

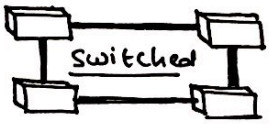
- **Network Types**
  - **LAN** (local area Network)
    - These are usually privately owned & connect a few Hosts.
    - They can be as simple as 2 pcs connected with a cable.
    - Each host has a unique identifying address and the packet sent has both the sender and reciever address.
    - These are either common cable or switched.





  - **WAN** (wide area network)
    - WAN covers a wider geographic area & connects connecting devices.
    - They're owned by companies.
    - WAN can be point to point b/w two devices
    - WAN can be on a switched Network for multiple devices.

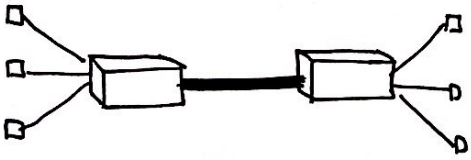




Switching

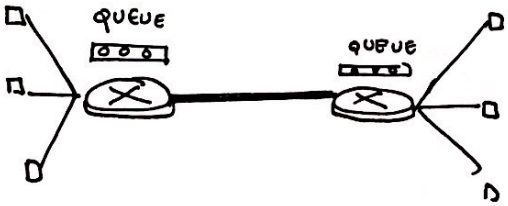
Circuit Switched

- Contigious stream
- uses a switching system
- No storage.



Packet Switched

- discrete
- uses as routing system
- maintains data on a queue



Network Models

- ## Protocol Layering

To accomodate communication that is complex. we break up the task into layers.
This is an example of using modularity to decompose a complex task into simpler ones.

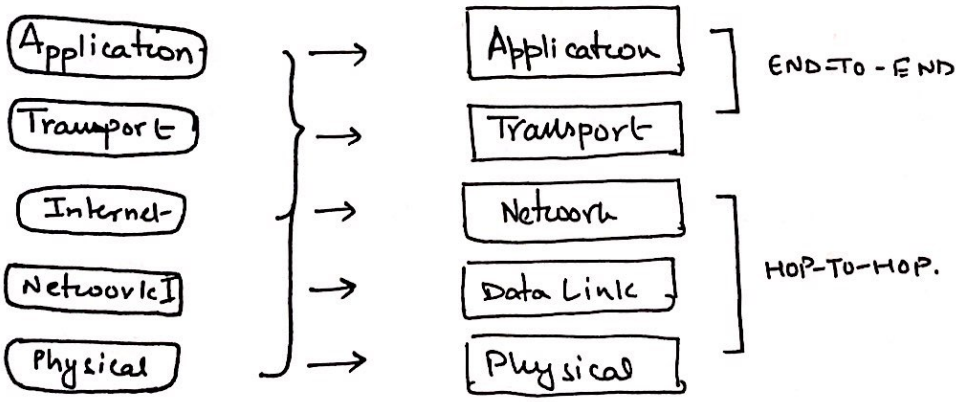
- ### Advantages
  - Seperates the services from the implementation. A layer recieves info from a lower layer & passes it to a higher layer & vice-versa.
  - Allows for simpler intermediate system which only need some layers to function.

- ### Principles
  1. The layers need to be bidirectional in communication.
  2. The layer at the same level should be identical on both sides of the line.

- There's a logical connection between the layers.

TCP/IP Protocol Suite

It is a hieraclical protocol made up of interactive modules, each of which provides a specific functionality. The upper levels are supported by the functionality of one or more lower levels.
The layers are shown joined with logical connections to make it easy to understand the duty of each layer.



• TCP-IP Layers

1. Physical Layers

- Carries individual bits in a frame across a link.
- At the link level the connection is still logical because there's also the physical medium in between.
- The logical unit exchanged is bits.
- The issues due to the medium are dealt with here.
- The bits are transformed to signals & V.V here.

2. Data Link Layer

- Responsible to taking the datagram & moving it across the link.
- The unit here is the frame.
- Takes care of the Hop-to-Hop transport.

3. Network layer

- Responsible for creating a link between the sender and reciever. i.e. Host-to-Host link.
- Controls the Packet routing.
- The unit here is a packet.
- Has other Protocols IP / ICMP / DHCP / ARP.
- Quality of service provider.

4. Transport Layer

- Takes the data from the application & splits it into datagrams / segments.
- This Has the main TCP, UDP protocols i.e Flow control error control and congestion control.

5. Application Layer

- The communication is b/w two processes.
- Provides user services
- Has the HTTP/ FTP/ SMTP / TELNET Protocols.
- equal to the combination of Presentation, session, application in the OSI model.

Addresses

Physical	Logical	Port	User
• Used by the DL	• Used by the NL	• Used by the Process	• Used by People
• 12 Hex digits	• 4 bytes	• 16 bit TCP/IP	• .com / IDs domains
• MAC Address	• IP Address		

The OSI Model

- This is a model developed by the ISO to understand & design a network architecture that is flexible, robust, & interoperable.
- This is a 7 layer framework each layer defines a part of the process of moving information.
- Apart from the layers in TCP/IP It also has.

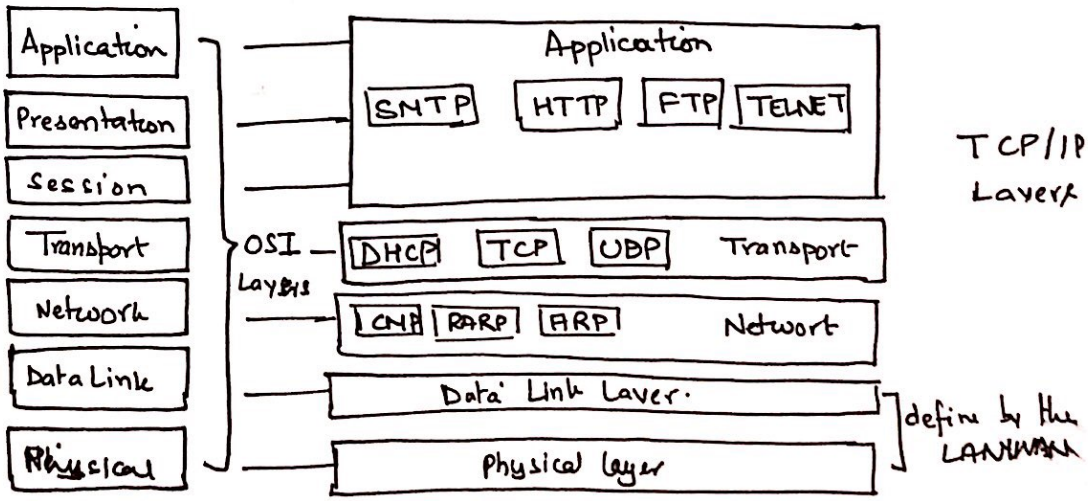
1. Session Layer
   - Establishes sessions b/w diff devices
   - logging, sync & token mgmt.
2. Presentation layer
   - The info being transmitted is symantic & syntactically checked.
   - Manages high level abstractions
   - controls the compression, translation & encryption.

- OSI vs TCP/IP

- TCP/IP does not have a session & presentation layer, this is left to the developer to implement
- OSI is NOT A PROTOCOL.
- The Transport layer can have multiple protocols in TCP/IP.

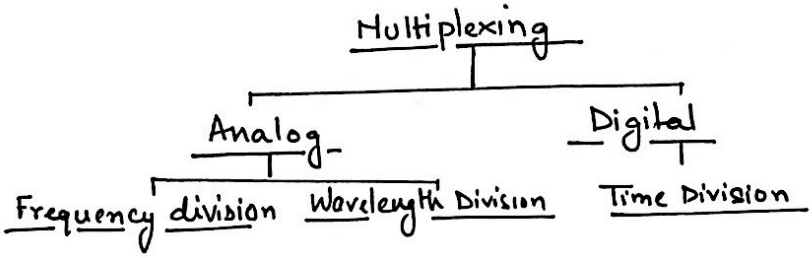
- WHY isn't ISO used?

- When ISO was introduced TCP/IP was already in use & the switch would've been expensive & time consuming.
- Some layers in OSI are not actually defined.
- The performance wasn't high enough to make a shift to it.



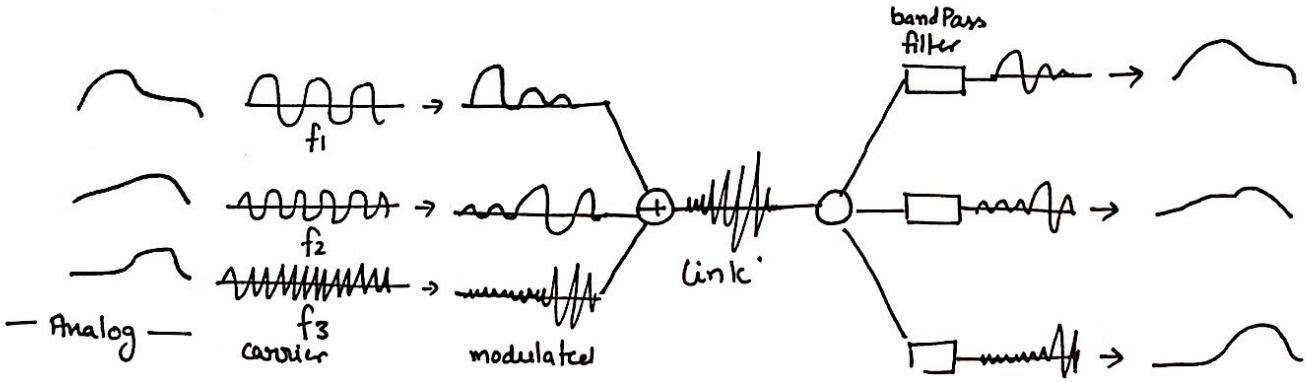
Multiplexing

- If the bandwidth of the link is greater than that needed by the devices, then it can be shared via Multiplexing.
- n lines share the bandwidth of one link. The MUX combines the signals into one signal and the demux filters out the individuals.



1. Frequency Division Multiplexing

- Applicable when the bandwidth of the link is greater than the combined bandwidths of the signals.
- The sender modulates different carrier frequencies, which are then combined to a single composite signal.
- The channels are seperated by guard bands to prevent overlapping.

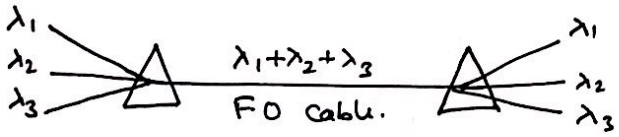


FDM process

- **Analog Coorier Systems**
  - The carrier/link signals can further be multiplexed into groups from groups ~~to~~ (KHz) to Jumbo Group (MHz).
  - The Jumbo groups can carry up to 3600 channels.
  - Used extensively in telephone lines.

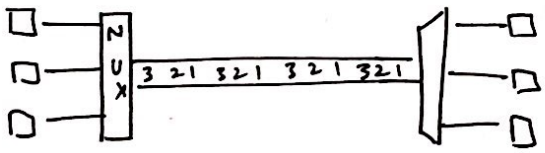
2. Wavelength Division Multiplexing

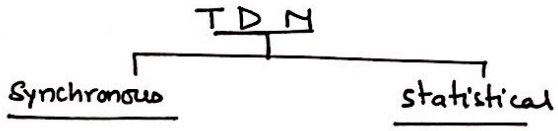
- Similar to frequency division multiplexing but designed for high rate links like fiber optics.
- The combination and splitting of light is done using prisms.
- Each channel is a specific wavelength of light / narrow band.
- This is used in SONET Networks.



3. Time Division Multiplexing

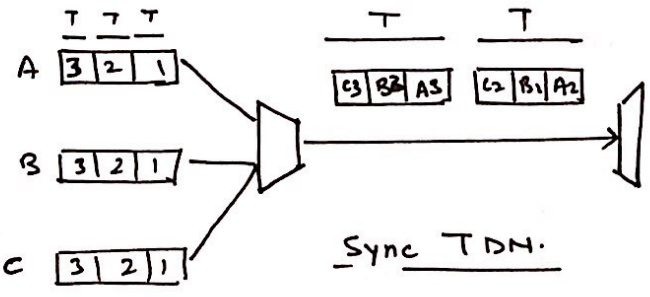
- A digital mux where each signal occupies the entire link for a time slot.





• Synchronous TDM

- Each input gets the same sized time slot. if the time slot is T and there n lines then each gets T/n time.
- all T/n are collected into a frame and sent on the link.
- The data rate of the link must be n times the rate of the inputs.



Sync TDM.

Drawbacks

- If the source does not have any data then the time slot for it goes blank. i.e wasted.
- If the rate of input is not as same as that of the link/others problems can happen to handle this we have
  - a) **MultiLevel MUX** : Two/More low rate lines are muxed to be in sync with others. The freq is a mulliple of others.
  - b) **MultiSlot** : Two/more ~~high~~ ~~rate~~ are made by demuxing a single high rate line.
  - c) **Pulse stuffing** : when the freq isn't a multiple, we shift the freq to be a mulliple of the others.
- Synchronization b/w MUX and DEMUX is a major issue implementationally
  - **Frame Synchronization**

    To sync MUX & DEMUX we add sync bits which allow them to be in sync.

Statistical TDM

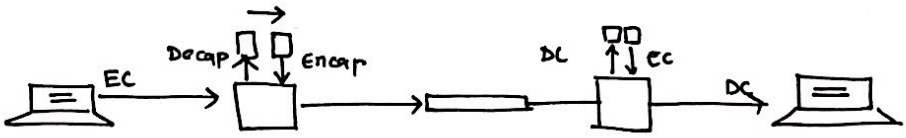
- To overcome the drawbacks of SyncTDM, the slots are allocated dynamically. The slot is only set when the sender has data.
- The number of slots is less than the input lines. as ~~so~~ all n devices wont communicate at once.
- The slots carry the address of the data as well to facilitate this dynamic system. The addressing can be as simple as the lineNo.
- The address: data slot size is in the bits : bytes range.
- There is no need to sync bits.
- The capacity of link is not necessarily equal to that of the sum of the inputs. This is defined based on the statistical load.

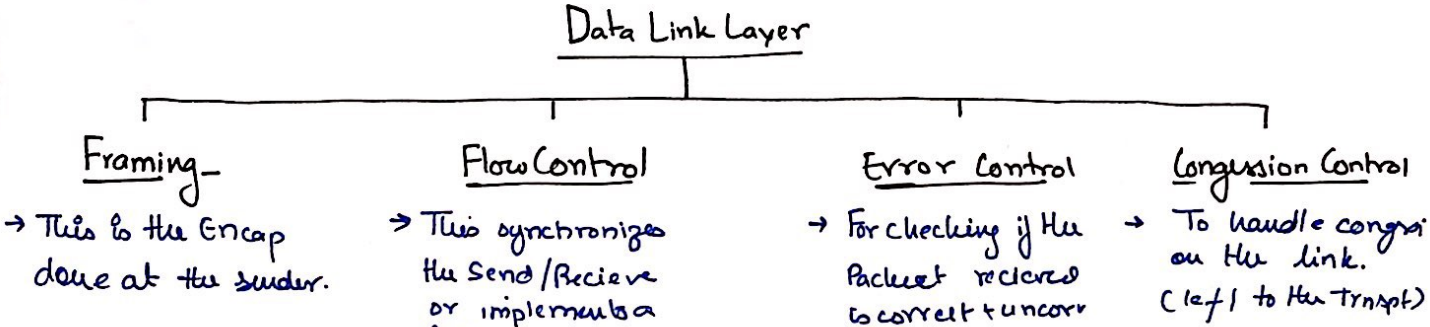
Data Link Layer

- The DLL is responsible for ~~link~~ node-to-~~link~~ node communication. The TCP/IP does not define any protocols for DLL, this is left to the Network.
- The hosts and routers are refered to as nodes
- The connection b/w nodes is called the link.

Services of DLL

- The DLL delivers the datagram to the next node in the path.
- The DLL encapsulates the datagram into a frame and decapsulates it at the reciever.
- Each intermediate node does the decap-encap too.





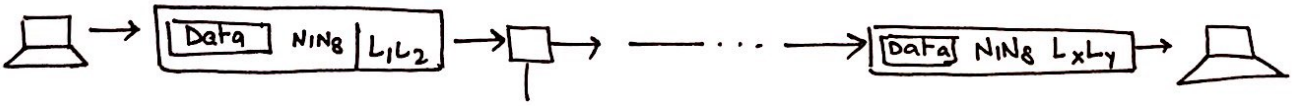
- The links can be point-to-point or broadcast.

Sublayering-

- The DLL is sub divided into the Data Link Control and Media Access Control. DLC controls both PtP and BC but MAC only does BC

Link Layer Addressing-

- This is called the Link/Physical/MAC address.
- The source when making the packet adds two address dest and sender address. These are changed at each node.



Address Types

Addresses

UNICAST

1:1 comms

Each node has a UC Address.

MultiCast

1:N comms

The juris diction is inside the link

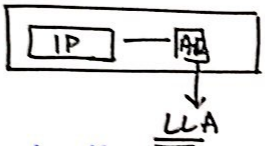
Broad Cast

1: ALL

The data is sent to all nodes.

Address Resolution Protocol

- The sender knows the IP of the default router and of the dest but to get Link layer addresses we need the ARP.
- The ARP accepts the IP address and returns the corresponding Link layer address.



- When a sender needs the LL addr it sends an ARP req with its MAC and IP as well as the dest IP. The IP of reciever reads this and in turns the reply with the LLA.
- Both the request and reply are broad cast.
- The ARP introduces caching of the LLA which greatly improves the efficiency.

Data Link Control

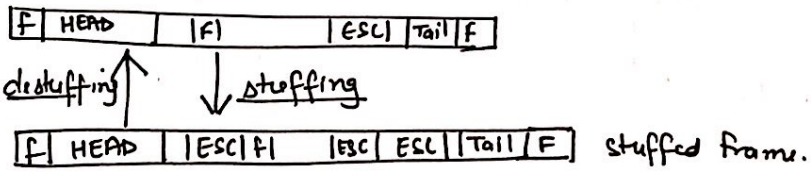
- Deals with procedures for communication between two adjacent nodes irrelevant of the link type.
- It performs 3 tasks

1> Framing -

- Seperates a message from one source to dest by adding a sender and destination address.
- The frame sizes can be fixed or variable, with fixed frame sizes we dont need any seperators
- with variable size we need to define the boundaries.

a> Character oriented Framing -

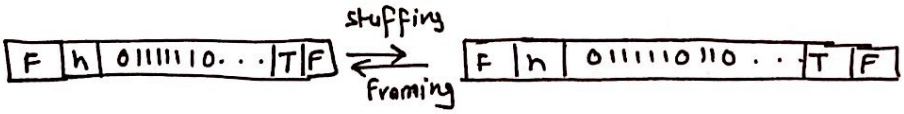
- Done for 8 bit ASCII characters. The Frame is compromis of a Header, the data, trailer,
- Two frames are seperated by flags.
- If the Fraame ~~data~~ data contained the same pattern as the flag, it would end the frame incorrectly thus we add escape sequences to prevent it. This is called Byte Stuffing.



- This is not compatible with the current unicode standers
  ∴ we use bit level stuffing.

b> Bit Oriented Stuffing -

- The flag Pattern is defined as 0111110
- whever 5 consecutive 1's occur in the data, we ~~pad~~ stuff a zero

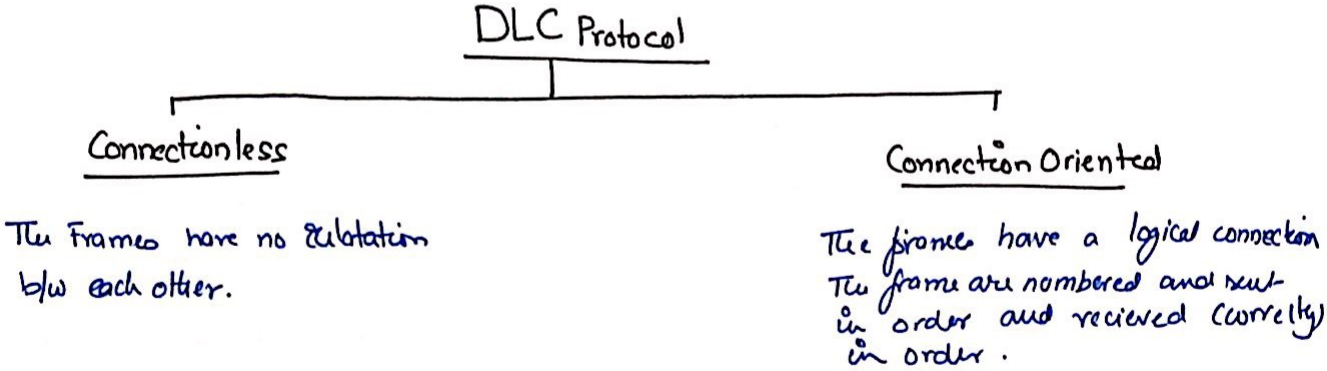


2> Flow Control

- The Flow is b/w 4 Entities. NL & DL of sender + NL & DL of Reciever

- Buffers

- maintain buffer at sender & reciever. while the buffer is empty for consumer send frames, otherwise buffer up frame in the sender.

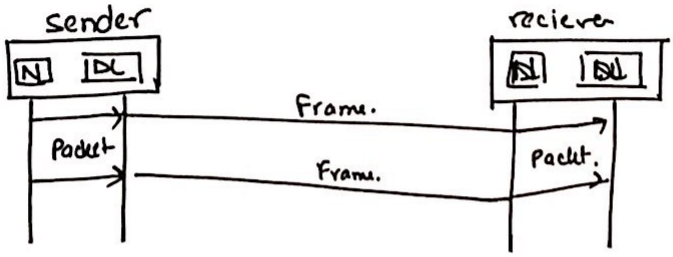


Flow Control Protocol

→ These behave like Finite State Machines and go from one state to another.

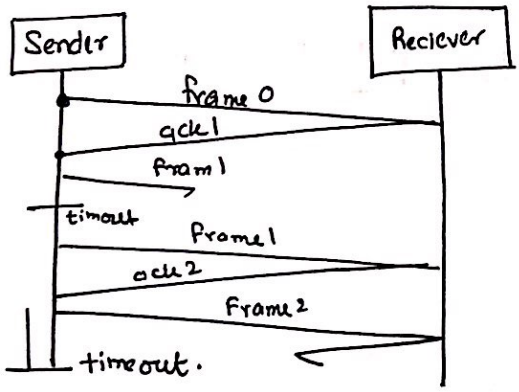
a> Simple Protocol

- Assume the reciever can immediately consume the frame. the reciever can never be over whelmed.



b> Stop and Wait

- The senders sends a frame and starts a timer if the reciever get the frame correctly it returns an ACK when the reciever get this ACK it sends the next
- if the ACK is not Recieved then the timer runs out and the sender assumes the packet was lost & retransmits it.
- To prevent duplicate copies we also keep seq no and ack no values.



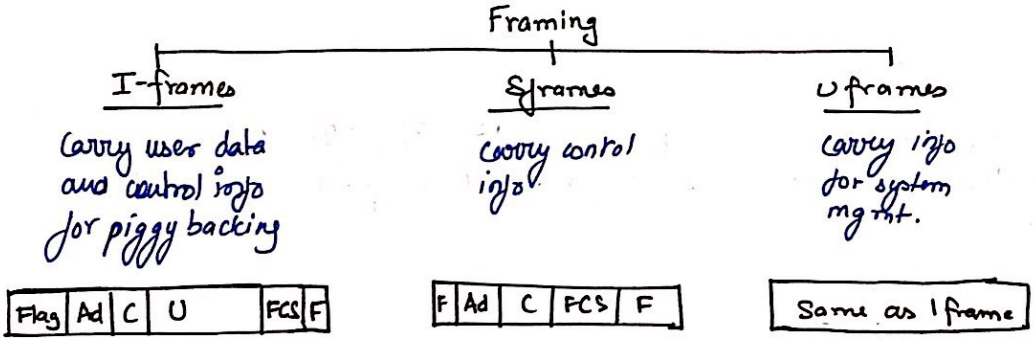
- High Level DLC

- Bit oriented protocol that uses the stop and wait protocol.
- There are two modes of transfer.

(NRM) : unbalanced confing, One primary multiple 2nd ary.
Prime can send 2nd can only respond.

(ABM) : This is for P2P work and each node is a peer.

• HDDC Framing

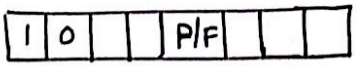


Flag : The sync pattern
Address : Contains the to address (primary) or a from address (2nd)
Control : 1 or 2 byte info for flow / Error check
User date : Contains the data
FCS : Frame Check Sequence. Error Detection.

o> Control of I frames

- Bit 1 = type, 0 => I frame.
- N(S) are the seq number of the frame.
- N(R) are the ACK field used in Piggy backing.
- P/F means Poll/ Final, when =1 means it was sent by Primary.
  when = 0 means it was sent by 2nd ary.

o> Control of S-frames



- Bit 1,2 mean this is a S frame.
- NCR) The ack number.
- Based on Bit 3,4 we have 4 states
  1) Reciever Ready :(00) fame recieved correctly N(k)= ACK No.
  2) Reciever Not Ready: (10) congession control.
  3) Reject (01): The NACK fram showing fram rejection
  4) Selective Reject : (11) The NAK for selective repeates.

o> Control of U-frames

- These are used by the system for manag ment.

Point-to-Point protocol

- defines the frame format and how the connection is established.
- This provides Network address configuration.
- PPP does not provide flow control.

Framing in PPP

Flag	Addr	control	Protocol	Payload	FCS	Flag
1by	1by	1by	1-2 by	var	1by	1by.

Flag: Signals frame boundary set to 01111110.

Address: set to (1) i.e as broad cast } due to absence of Flow Control.

Control: set to 00000011 i.e empty cell

Protocol: defines what is high carried.

Payload:

FCS: Frame Check Seq: 2/4 byte CRC.

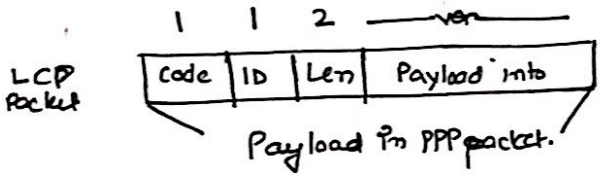
- The Frame stuffing esc seq is 01111101.

Multiplexing

- PPP can provide mechanism for Linking, authentication & network tasks. via mux.

0) Link Control Protocol

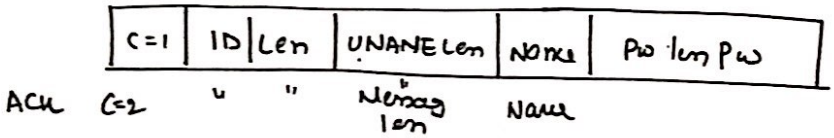
- Establishes, maintains, configuring and maintaing the link.
- The info for the is carried in the payload and the code is set to C021.
- There are 3 type of Packets [1–4] for establishing [5-6] termination [5-0] monitoring.
- ID holds the request-reply match.



Authentication Protocol

1) Password Authentication Protocol

1) Sender sends ID and Pw
2) System check validity and either ace or rej the req.



2) Challenge Handshake Authentication Protocol.

1) A sends req with challenge value.
2) B applies function on value & pw & replies the packet
3) A does the same and if the result are same then connection is made.

Network Control Protocol

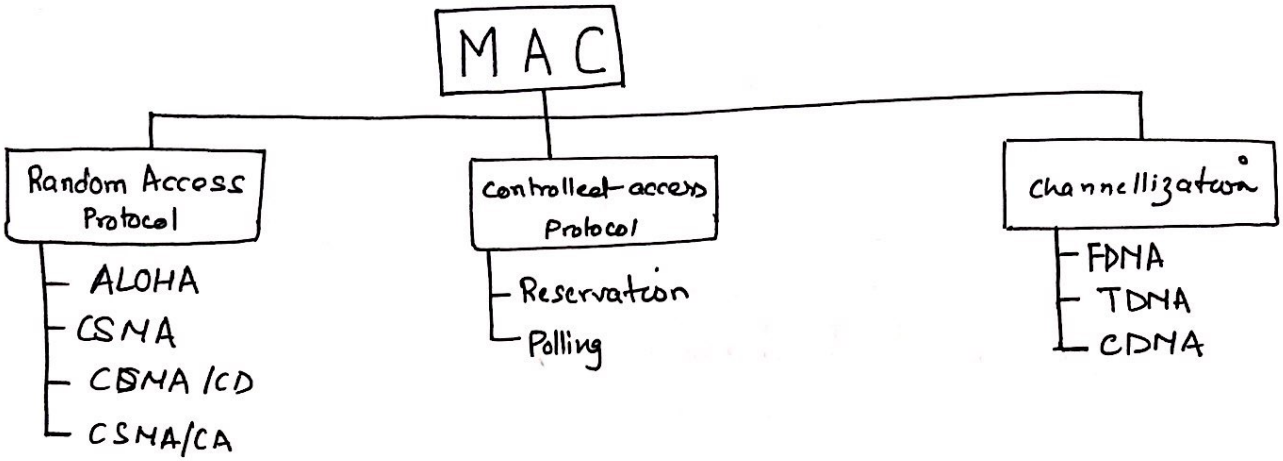
- There is one for each type of system. one of them is IPCP.

– IPCP

The protocol code is 8021 and the pack is same as multiplexed PPP.

- There's also Multilink PPP.

Media Access Control



ALOHA

- If the sender has a frame, send it. This can cause collisions
- To check for recieval it uses ack from reciever. within a time slot.
- If No ACK is recieved resend, there is a possibility that they will collide too.
- PURE ALOHA says wait for some T_B before resending.
- After K_{max} tries give up and try later.
- time-out = $2T_{prop}$
- one method to set T_B is binarye backoff.

> Volnerable Time

- lenght of time for which there can be collisions.

$$T_{Vul} = 2 \times T_{fr}$$

T_f - avg Fram transmission time

> Through Put

$$S = G \times e^{-2G}$$

G = avg frames generated

$S_{max} = 0.184 \quad (G = 1/2)$

Slotted ALOHA

- The time is in T_{fr} slots and transmission is only done at the start of the slot
- The vulnerable time is reduced to just T_{fr}
- The throughput also increases to 0.384 theoretical Max @ G=1.

Carrier Sensing Multiple Access

- Tries to minimize collision by sensing the channel.
- Collisions can still happen due to the propagation delay in medium

Vulnerable Time

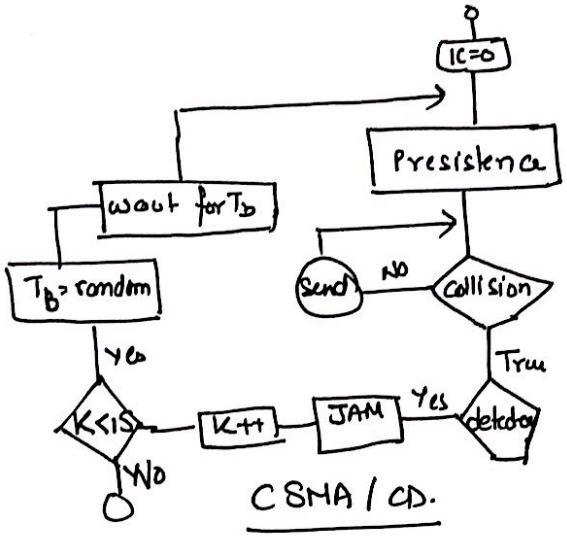
This is the same as the propagation Time.

Persis Methods

- 1-Persistent
  - if the channel is idle send immediatly
- Non Persistent
  - if the line is idle send immediatly
  - other wise wait for some random time befor trying again.
- P- Persistent
  - if the line is idle
    1. send fram with probability (P)
    2. with probability q wait for next time slot.
       if line is idle goto 1
       else backoff.

CSMA/CD

- This has methods to handles collesions.
- Monitor the channel to check for succes.
- The Frame size must be restricted to ~~be~~ be able to check for collision before the last bit is sent.



CSMA / CD.

Through Put

- The Max-throughput is greater than ALOHA and depends on G. for P presistant it is 50% but can be as high as 90% for non presistnet @ G = 3-8.

CSMA / CA

The collisions are avoided by using 3 strategies:

1) Interframe space (IFS) : The sender waits for some t before sending even if the channel is idle. This allows to avoid starvation.
2) Contention Windows : A ready station select a random amount of slots to wait for this is according to backoff. This is similar to p-presistnet and prioritizes the slots whoive waited the longest.
3) Acknowledgement : A positive recieved acle and timout is also kept.

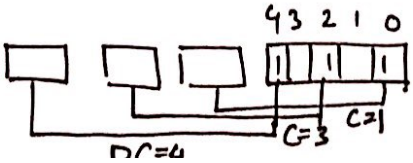
- This also has a Networ Allocator Vector to keep track of the busy channel.

Controlled Access

Controlled Access

Reservation

→ each station reserves a time slot and the Pack carries a reservation list



Polling

- works on topos with P & S systems. all data is from P.

Select : P has something to send. P tells all S to give ack for the tranomission.

Poll : when P is recieving it Polls the S if they have some item to send.

Intro to Network Layer

• Performance

1> Delay–

- Delay can be divided into 4.

$$Delay_{tr} = (Packet\ len) / (Transmission\ Rate)$$

$$Delay_{Prop} = (Dist) / (Prop\ Speed)$$

$$Delay_{Pro} = decision\ Time$$

$$Delay_{qu} = queuing\ Time$$

$$Delay_{Total} = (n+1)\left[\sum Delay\right]$$

2> Throughput

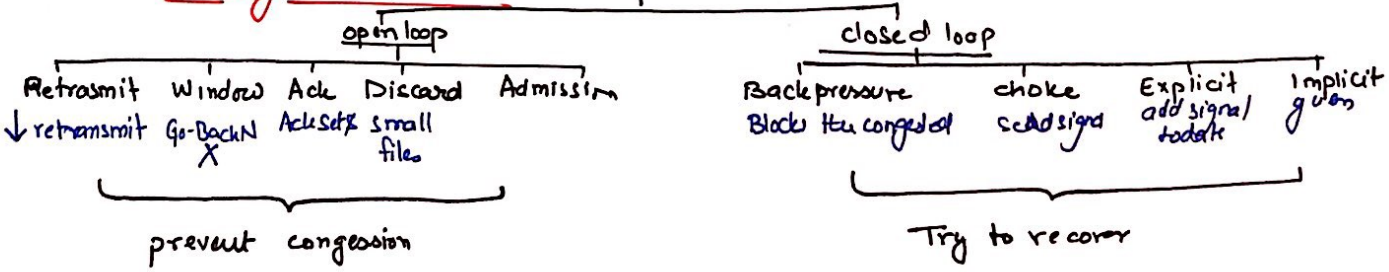
- The minimum transmission rate of any link in the net.

$$Thp = MIN(TR_1 \ldots TR_N)$$

3> Packet Loss

- The No of packets lost or corrupted or rejected in the network.

4> Congestion Control CC



The Physical layer

Max Bit Rate = $2 \times \text{bandwidth} \times \log_2 L$

L = No of signals.

Shannon Capacity = $\text{bandwidth} \times \log_2(1+SNR)$ SNR $\neq$ 3162

Network Security -

Goals

- **Confidentiality** - Keep it a secret
- **Integrity** - Keep it together
- **Availibility** - Keep it available.

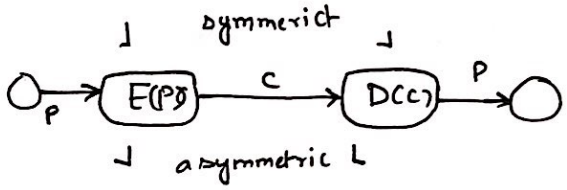
Attacks

on confidentiality.
+ Snooping
- Access/Intercept the data to get into
+ Traffic Analysis
- Track data to get info on Nature of communication

on integrity
+ Modification
- Intercept, access and modify data to make it harm.
+ Masquerading
- Pretending to be someone else.
+ Replaying
- Intercept and replay
+ Redudation
- Deny sending/recieving

on availibility
+ Denial of Service
- Buy a chromecast.

- Cryptography -



1. Traditional Ciphers

- Transposition
- Substitution
  - Mono.
  - Poly.

Substitution: Replace one letter with another. Eg A → S.

- Monoalphabetic: The letters sub^d are mapped to the same always i.e one to one.
  Eg A → S ∀ A in text.
  - additive/Shift cipher: Each key is mapped to a % 26 number, the key is also a mod 26 number.

  Done by Right / Left shift of the letter % 26.

Key = 15 (e = 7) e+15 = 22 = (w)

- Additive / Shift can easily be brute forced.
- Better way to agree on the mapping scheme.

Poly alphabetic : 1:m mapping. They hide the letter frequency.

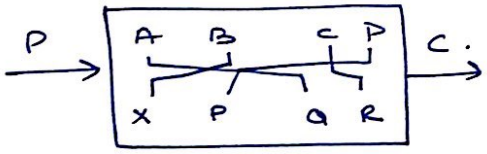
- depends on both the char and the position.
- key = stream of sub keys each sub key depends on position.

Autokey Cipher : K_0 is known and exchanged.
K_1 is value of 1st pt char.
K_2 is value of 2nd and so on.

E⇒ $C_i = (P_i + k_i) \% 26$ D⇒ $P_i = (C_i - k_i) \% 26$.

Transposition Cipher

- Reorders the symbols. works in blocks.
-

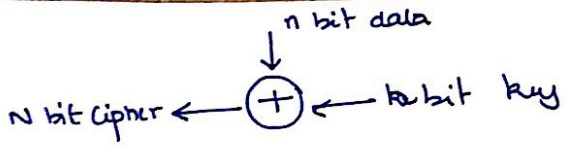


- Stream Ciphers : work on the char level.
- Block Ciphers : work on the Blocks of data,
- Combined → use both.

2) Modern Symmetric key-

- Done at bit / byte level.

- Block Ciphers
  - n-bit block. k-bit key. $n \in (64, 128, 256, 512.)$
- @ Block level they look like subⁿ.
  - Made out of transpose units P-boxes + S-box + XOR
  - P box : same as traditional Transpose.
    - Permutation
    - compression
    - expansion.
  - S box : bit level subⁿ cipher.



- Circ shift.
- Swap = circ shif by $n/2$.
- Split by $n/2$, combine.

- **DES** Data Encryption Std.

  64 bit Pt ⟶ 64 bit cipher. } 56 bit key.

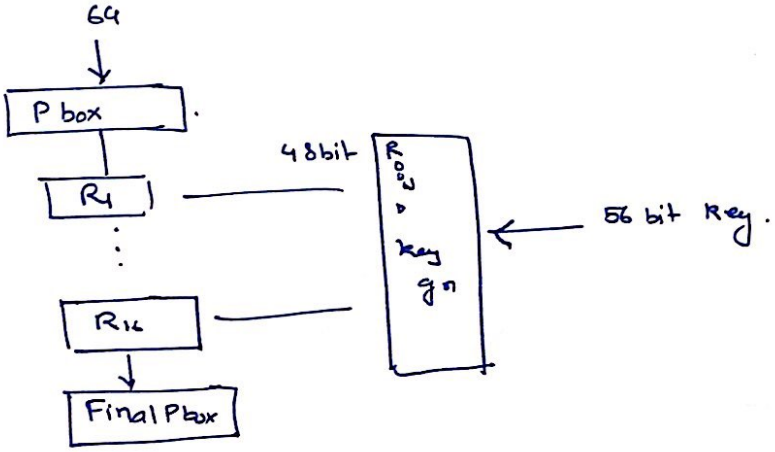
- Rounds: 16 rounds, each is an invertible Feistal transform.

  Round takes L_{i-1} and R_{i-1} from a round & does $L_i \leftarrow R_i$.

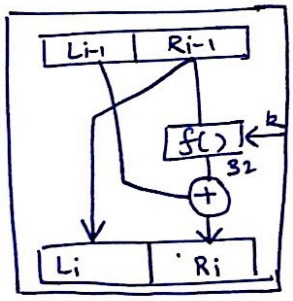
  each round has upto two {mixer / swapper}.

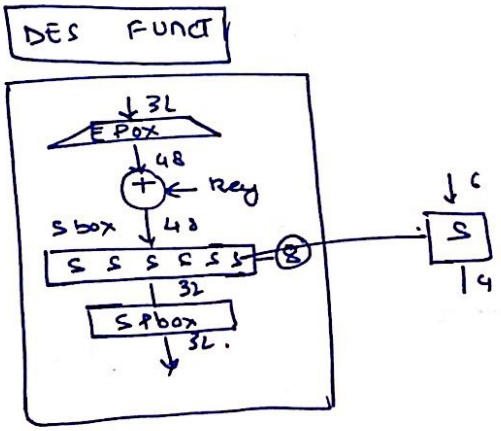
- DES Function: 48 bit key to Right most 32 bits → 32 bit out

DES

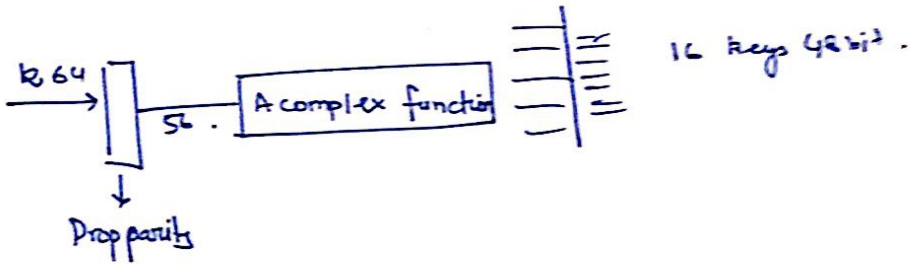


DES Round





- key gen : 64 bits → 8 for parity



- Triple DES

- Does DES 3 times

  uses 2 or 3 key i.e 112 or 168 bit.

E = P → Encrypt → Decrypt → Encrypt → C

k1 k2 k3

D : C → D → E → D → P

k1 k2 k3.

- AES Advanced Encryption Std.

	Key	Round
128 bit data	128	10
	192	12
	256	14

- Starts with an XOR and the k rounds (depends on key).

Sub Byte

128

↓

byte Sub^n

↓

Byte Perm

↓

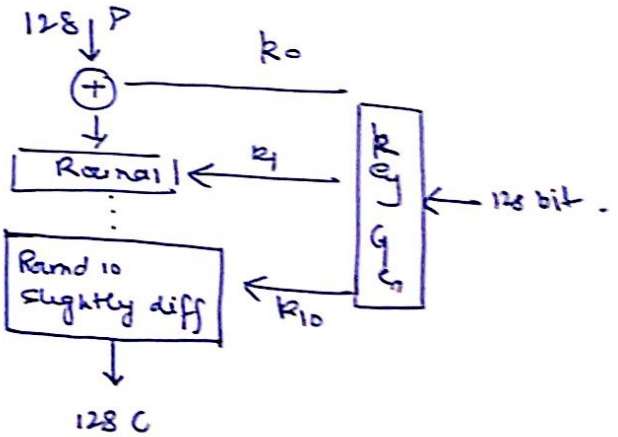
Complex Function

↓

⊕ ← ki

↓

128



Mode of operation

- ECB | CBC | CFB | OFB.

- Electronic Code Block:
  - Purely Block cipher.
  - Divide to N blocks
  - Same key ∴ blocks with same text give same cipher..
  - Independent i.e errors are not propagated.

- Cipher BlockChaining
  - includes the prev block.
  - Initial block takes some IV.
  - Blocks are dependent.
  - Same P → diff C.
  - Error is propagated.

- Cipher Feed Block.
  - ~~Just CBC but they instead use the prev key.~~
  - use r bits of data when R is diff.
- OFB.

  use the output of prev instead.
  - Errors dont prop.

→

Asymetric Key

- ## RSA

uses two nos e and d as pub and pvt.

Selecting P and d.

@: Bob:

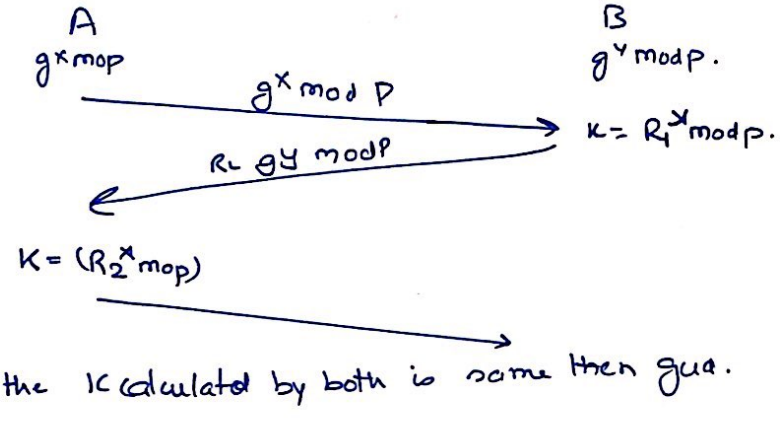
1. select some very large P and q
2. get $n = p \times q$
3. get $\phi = (p-1)(q-1)$
4. get some random e and get $d \times e = 1 \bmod \phi$
5. announce e and n to public, keep ϕ & d secret.

E => $C = P^e \bmod n$

D => $P = C^d \bmod n$ $(P < n)$.

Diffie Hellman KS

- Two parties choose P and g P is a large Prime.
- Each party chose some x any y as secret — A → x By.
- Alice calculates $R_1 = g^x \bmod p$
- Bob calculates $R_2 = g^y \bmod p$

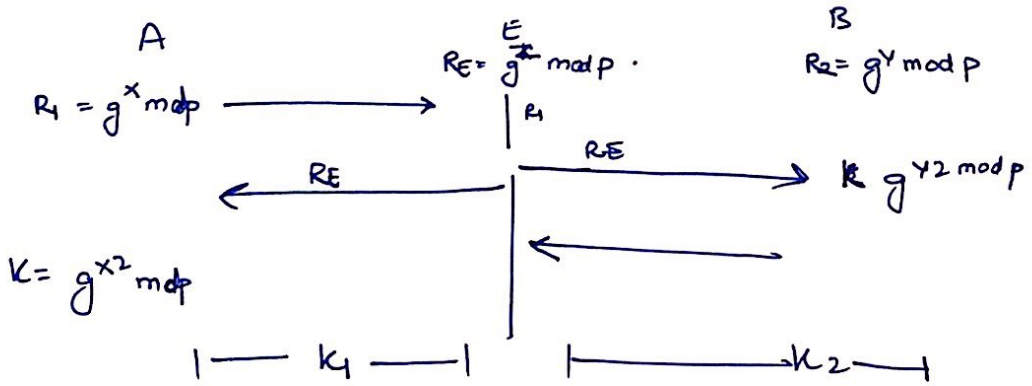


if the IC calculated by both is same then gud.

K is the session key.

$$K = g^{xy} \bmod p$$

- Man in the Middle Attack



=> GG WP

solution? Authenticatain

Security Services

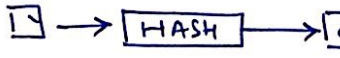
- Message
  - Confidentiality
  - Authen
  - Integrity
  - Nonrepud
- Entity
  - Auth.

- Confidentiality
  - Symmetric key.
    - Both S and R have the same secret
    - cannot be exchanged as plain text ∴ version to exchange it is needed.
    - The session key itself is a asymm key.
    - Very efficient
  - Asymmetric key
    - No key sharing, the pub key is announced.

(A) ↳ $E(pub_{Bob}, P) \rightarrow C$ ⟶ $D[C, PVT_{BOB}) \rightarrow P$ (B)

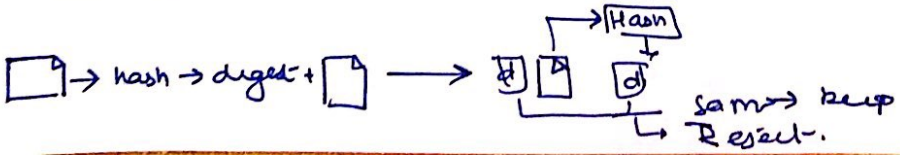
    - Mathy ∴ slow.
    - No authenticity of sender/ recipient..

- Integrity -
  - Sometimes Security may not be imp but rather the integrity
  - Document and finger print.
    ↳ Both needed.
    - This is done with a 'digest' of the doc.



    - The Doc & print are linked together.
    - The digest must remain secret..

creating & checking

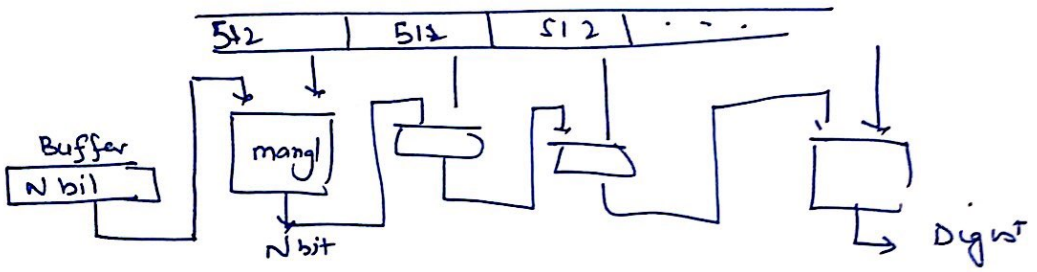


Hash function Criteria

- One Wayness
  - cannot back track i.e get the msg from digest
- Weak collision Register
  - msg cannot be easily forged.
  - None can create a doc that makes the same digest.
- Strong Collision Resist
  - No two msgs should make the same digest
  - i.e send two msgs & say one is wrong.

SHA-1

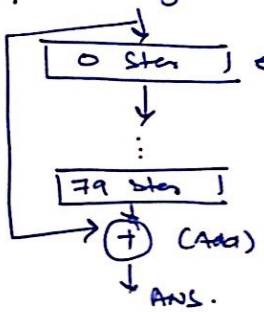
- consists of a mangler i.e mix two inputs to get the out



→ 160 bit digst 5 32 bit words.

- The 512 bit block only has 16 words we need 80 ∴ we expand the block.

→ mangler : 80 steps , ea gets 32 bit key + the inpt



Message Authentication

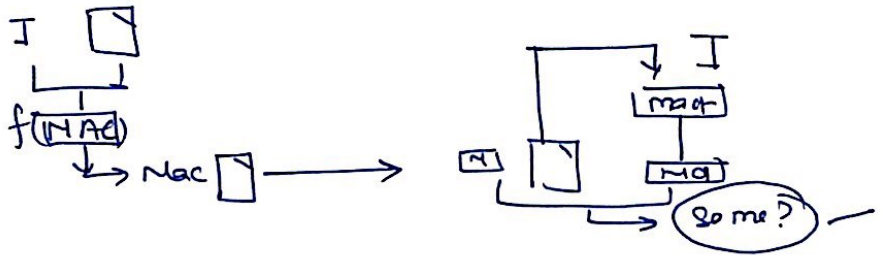
#ing cannot tell if the sender is legit.

(MDC)

- The Digest is called the Modification, Detection, Code.

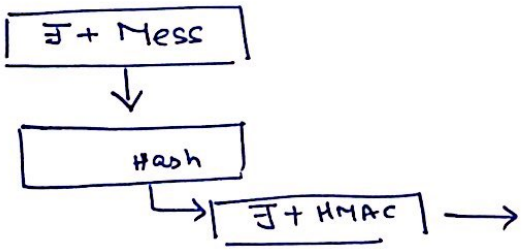
- MDC is keyless #
- To make a Message auth code we need a keyed #

  Key # = symmkey



- HMAC (#ed MAC)

  Nested MAC by applying keyless # to the con cat of message & J



Digital Signatures

→ HMAC Needs J keys

→ works exactly like a IRL signature.

IRL vs Digital

→ Digital is seperate from the Doc.

→ Reciever applies the verify on the doc+sign

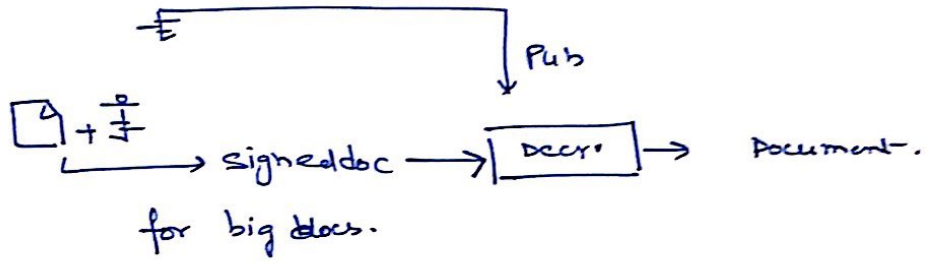
→ Digital is diff for all.

→ DS cannot be distinguished if copied.

- Use of key
  - The sender adds her pvt key to a signing algo & reciever uses the pub key of sender.

2 Way

→ Sign the whole doc.



for big docs.

→ Sign the digest.

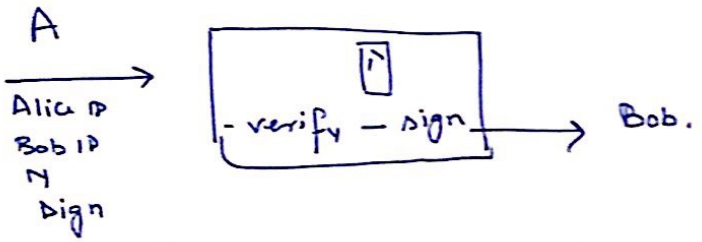
Why not use Symkey

→ only know b/w two, diff key for dif doc.

→ creating a key need & auth and oh boi. ∞ O

→ Misuse of key can be done.

- Services of Signature.
  - Integrity : yep, cannot get same sig if its changed.
  - Auth : yep.
  - Non Repu. : using a trusted party.
    - Using a trusted 3rd par to keep all keys.



The 3rd party can ~~say~~ give a message of auth.

Entity Auth

- Let one party give the ID of another.
  - claimant : Entity whose ID is to be proven
  - Verifier : the prover

- Happen in Real time.

- The claimant can use
  - Something known : only known only by your
  - possed : to prove
  - Inherent : sign, voice, prints

- Passwords
  - Something possed.
    → Fixed / OTP.

- Fixed
  ↳ Eavesdrop
  ↳ steal
  ↳ Access the password file
  ↳ Guess ⟩ store Hashed p/w.

Dictionary attack.

a list of all 6 digit numbers & check.

SALTING
- Append p/w with random string. → #
- make harder

- OTP
  → Make ED & Steal useless
  → complex

- Challenge Respons.

→ The challenge is a time varying value which is sent by verifier the claiment applies a function & sends it back.

- Symm Hash
  - Shared secret
  - Use Encrypt algo.

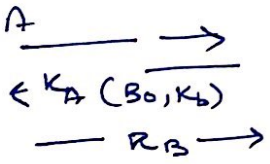
1. Ask for challenge
2. Bob gives a nonce R_B
3. Alice divers the Response

$$C = R_B$$
$$Resp = Enc(R_B, K_{AB})$$

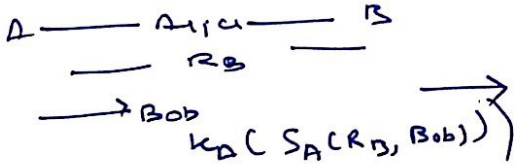
- The key must be secret
- Bob keep the key till Alice reponds
- Eve cannot send back Response key it's expired.

- Using # keyed.
  - Integrity kept
  - Some countries dont like Enc.

- Using ℱℰ keys.
  - Secret is pvt key
  - verifer uses pub key.



- Digital Sign



Key Management

→ How to get them keys?

→ Turn secret keys.

if N people want to talk we need $\sim N^2$ key.

1) Key Dist Centr (KDC)
   - 3rd Party

Process: Alice sends request to KDC for Lonna with Bob.

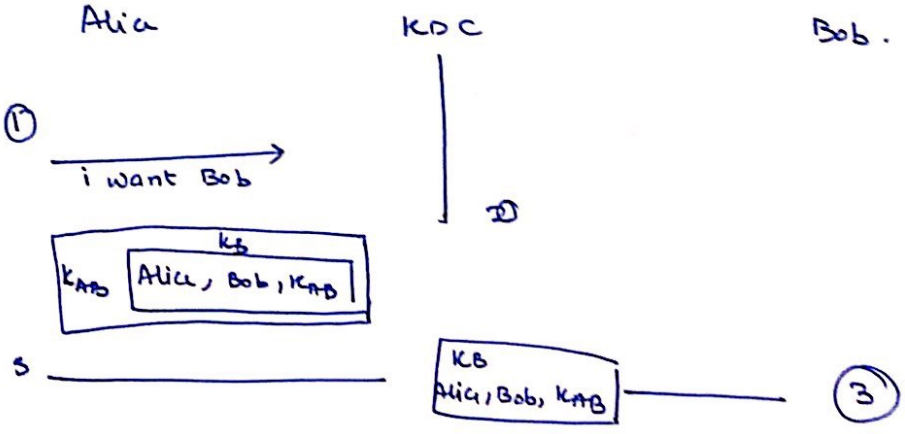
KDC tell Bob about alice.

Bob agrees for talking.

Session keys:

KDC makes keys for everyon (Secret & session).

1) Two Seckey b/w KDC & user.
2) Two session key b/w users.



1) Alice requests for Bob.
2) KDC makes a Ticket = $E([E(Alice, Bob, K_{AB}], K_B), K_{AB}])$
3) Alice sends info to Bob.

Kerberos Server

- auth protocol + KDC
- 2 servers : AUTH & TICKET Grant .

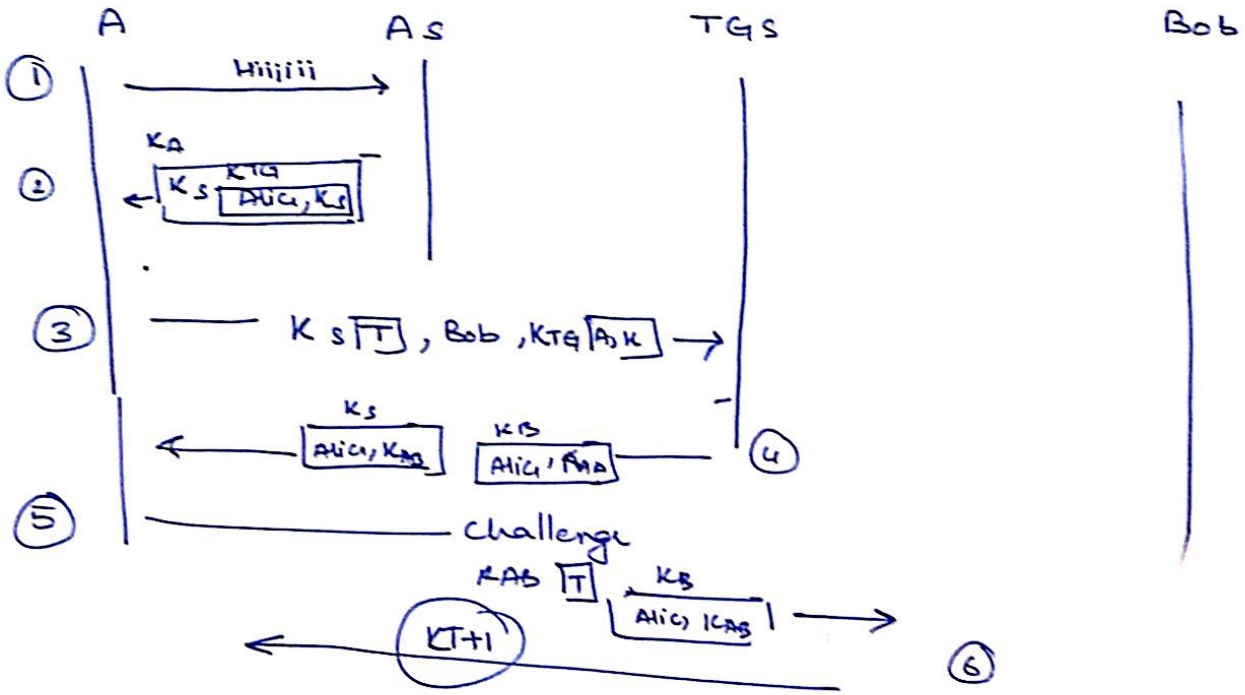
@ AS : Each user registers here & is given some ID & plw.
It auths user, issues a session b/w TGS & user.

@ TGS : issues ticket for Bob .

: A

- AS is checked once but TGS many times .

@ Real Server : Provides service to alice.



1, 2, 3 → KDC

- (4) 2 Tickets , first with K_S & one with K_B.
- (5) Alice sends a CRT to Bob & Bob's ticket
- (6) Bob Replies with CRT.

To get diff services, reuse K_S & do 3-6 again .

Realms

→ Alice asks local TGS for some other TGS.

Public Key distribution

→ Public Announcement
- naive
- easyly to forge by masquerading

→ Trusted center
- maintains dir of PubKey.
- Requires each user to register.

→ Controlled TC
- Announcement = (Announce + sign + timestamp) from a CA.
- Pub key is change prof.

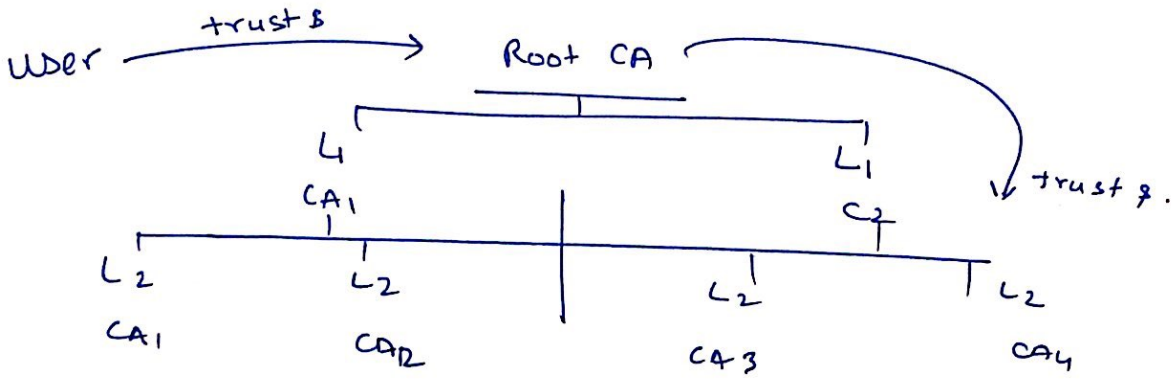
→ CA Certificate Authority.
- Bob wants 2 thing
  - to be famous.
  - NO fakers.
- CA is a federal / state org. The CA Pubkey is well known.

cert = { Bob's Pubkey
Time Stamp
Signature of CA.

- X.509
  ↳ format for certificate.

version
S.no
Sign
Issuer
Period of valid.
Sub : To whom
Sub Pub key
I U ID
Sub U ID
Extension
Encrypted: algo id, # of other, sign of th #.

. Keep a heirarchy of CAs



Elgamal Encryption

* $a^{-1} = a^{n-2} \pmod m$

1) Key generation

@ Alice

$G : Z_p^*$ (cyclic group G of size p)

g : generator of G.

1) select x from q at random

$x \in \text{random } \{1, 2, \ldots, q-1\}$

$h : g^x \bmod p$

→ Alice published (G, p, g, h). x is the pvt key.

(2) Encrypt

@ Bob chooses some y from Z_p^*

$c_1 = g^y \bmod p$

$k_{AB} = h^y = g^{xy}$

$c_2 = (k_{AB} * m) \,\%\, p.$

Bob sends (c_1, c_2) i.e $(g^y, m' \cdot g^{xy})$

(3) ~~dt~~ Decrypt

@ Alice $s : c_1^x \; (k_{AB})$.

$m = c_2 \cdot s^{-1}$

$pf = c_2 \cdot s^{-1} = m' \cdot h^y \cdot (g^{xy})^{-1} = m^{-1} g^{xy} \cdot g^{-xy} = m'$.

(Eg) B $m = 35$ $g = 5$ $p = 89$.

he @:A $x = 9$. $h = 5^9 \,\%\, 89. = 20$

(G, 89, 5, 20)

Bob: y = 4 $c_1 = 5^4 \bmod 89. = 2$

$c_2 = (67 \times 35) \,\%\, 89 = 31$

Send (2, 31)

@ Alic $s = 2^9 \bmod 89 = (67)$ $m' = 31 \cdot 67^{-1} = 3144 \; 124 = 35$

Elgamal Signature

(key)

$y = g^x \bmod p \quad (x \in 1 < x < p-2)$

$Y \rightarrow$ pub

$x \rightarrow$ pvt

Signature:

$k \in (1 < k < p-1) \quad \gcd(k, p-1) = 1$

$r = g^k \bmod p.$

$s = (H(m) - xr)\, k^{-1} \bmod (p-1)$

if $s = 0$ start over.

$H(m) = (xr + ks) \bmod (p-1)$

Verify

$g^{H(m)} = (y^r r^s \% p)$

RSA

P and q

1) calculate

$n = p \times q$

2) calculate

$\phi = (p-1)(q-1)$

(3) get some random e

get $d \times e = 1 \% \phi$

i.e mod inv

(5) announce P (e, n)

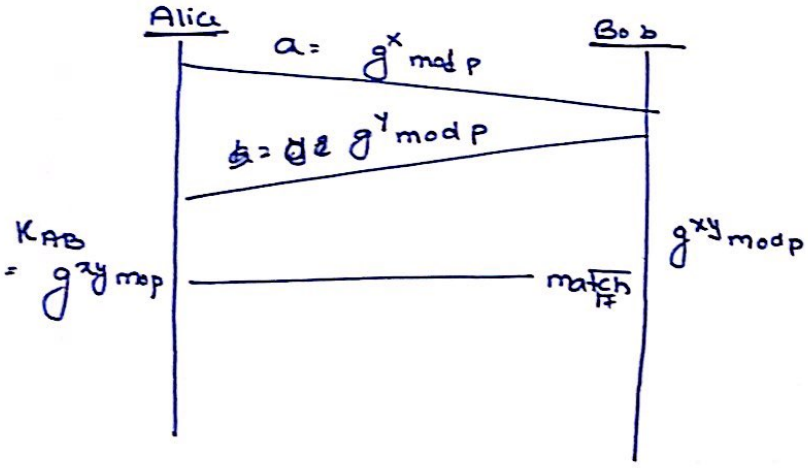
keep (d, ϕ)

$E = [P^e \bmod n]$

$O = [C^d \bmod n]$

DHKS

choose some p and g



Elgamal

Encpt

Z_p^*

choose somey from Z_p^*

$$c_1 = g^y \bmod p.$$

$c_1 \rightarrow$ public key

$y \rightarrow$ privat

send to ~~alice~~ ona.

Encrypt:

$$c_1 = g^y \bmod p$$

$$c_2 = ((Pub_{Alice})^y * m) \% p.$$

Send (c_1, c_2)

Decrypt:

$$s = c_1^x$$

$$m = c_2 s^{-1} \% p.$$

Signature

$$y = g^x \% p$$

$k \in 1, k, p-1$

$$r = g^k \bmod p.$$

$$s = (H(m) - x r)\, k^{-1} \bmod (p-1)$$

$$H(m) = (xr + ks) \bmod p-1).$$

$$g^{H(m)} = g^r r^s \% p$$

Bit rate = $Bw \times 2 \times \log_2 L$

Capacity = $bw \times (\log C + 1 + SNR)$

Throughput: $\frac{\text{avg frames/ sec}}{\text{frame size}}$

Latency = PT + TT + qt + Pd

PT = Dist / speed

TT = Message size / Blw

qt = ??

Net layer Delay = Packets/ Trate.

T .. Dist / Speed

Delay P

TCP Timers

$RTT_S = (1-\alpha)\ RTT_S + \alpha\ RTTM.$

$RTT_D = (1-\beta)\ RTT_D + \beta \times |RTT_S - RTTM|$

$RTO = RTT_S + 4 \times RTT_D$

Transport Layer protocols

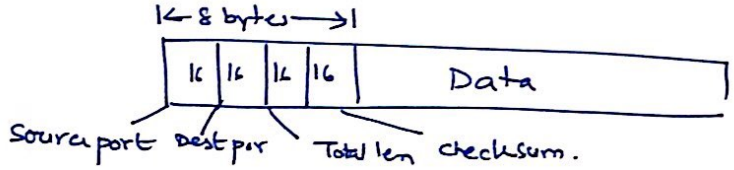
- For process-to-process delivery of data.
  - **UDP** : unreliable, connectionless, simple, efficient. can have error control.
  - TCP : reliable, connection oriented,

Important ports: 80: HTTP; 20,21 : FTP, 53 DNS, 67 DHCP.

Socket: IP + Port No

UDP

- connectionless
- unreliable
- doesn't add any service to IP.
- minimum overheads.
- Packets are the User datagrams:



UDP len = IP len - IP header len.

Pseudo Header to get checksum.

IP of Soure

IP of dest

| 0s | 8 bit Protocol | 16 bit len |

| Soura port | dest port |

| UDP total len | Checksum |

UDP Services

- P-to-Proc comms
  - uses sockets
- Connectionless
  - each data gram is independent

- Each DG take take a diff path.
- Doesnt allow streaming.
- Size limited to 65507 bytes.

Flow control

- No flow control.
- Reciever can overflow.
- Left to the process providing UDP.

Error Control

- No error control. just checksums.

Checksums

- included into the header.
- can be all 0s.

congestion control

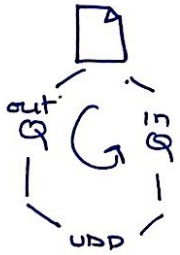
- Nope.
- assumes it cant create congestion.

Encapsan a Decap

- Yep

Queuing

- each port has an associated queue.
- When a process starts it- says a port.



Mux and Demux

- Yep.

Applications

- DNS.
- TFTP
- SNMP
- RIP
- interactive ,Real time apps.

TCP

- connection oriented
- Reliable
- has connection est.
- GBN & SR for Reliability.
- has check sums too :).

TCP service.

Proc-F Proc coms

- through ports.

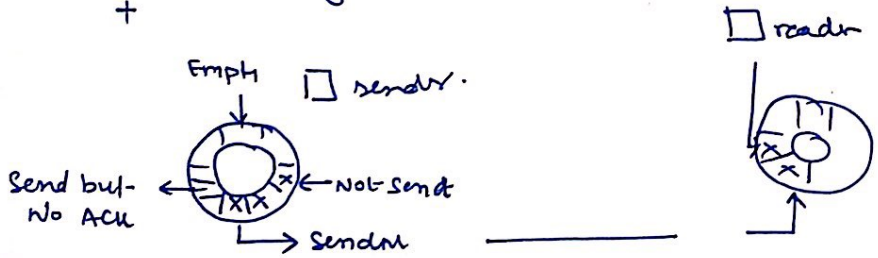
Stream delivery

- data is a stream of bytes.
- The connection makes a "tube" for data.

Send / Reciere buffers.

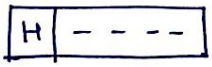
- Diff R/W speed
- cir arrays.

+



Segments.

- groups and number the data into segments, just in case.



Full Duplex Comm &

- 2 way comms

Mux and Demux

- Yep

Connection — Establish, talk, disconnect.

Reliability: uses acknowledgments.

TCP features

- Numbering system
  - Each segment is numbered. but this isn't in the header
    - Seq NO & Ack NO.

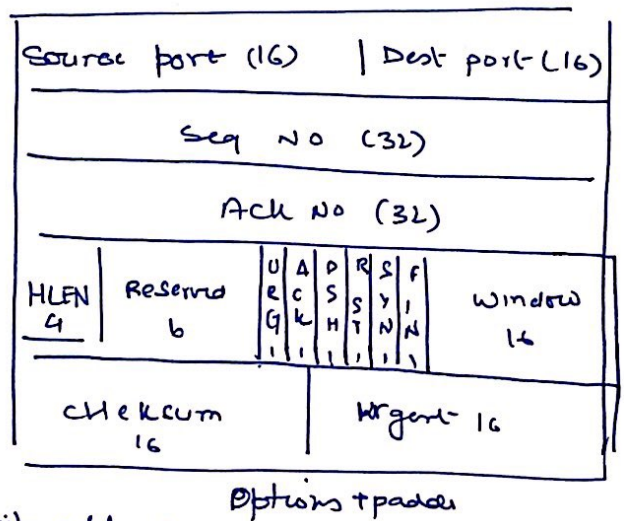
Byte NOs
→ All NOs are in Octets.
&
→ The start NO may be arbitary.

SeqNO
→ The seqNO of 1st Seg is ISN = Random
→ any other : Prev + its own size.
The seq NO is the value of first byte

Ack NO
→ The NO of next byte to be sent.
→ It is cumulative. (take no of last byte, add 1).

Segments
↳ The TCP Packet.

Format : 20–60 Bytes (20 minimum).



- source port : 16 bit add of port @ user sending
- dest port = " " " " " " " " receiver
- Sequence NO of packet. Seq. is ISN @ Random.
- Ack NO : No of next expected seq packet.

Header len : 4 bit field (Actual = (4x5 - 4x15))

Control : 6 different flags. for control flow, est, term etc.

control : URG : → urgent pointer is valid
ACK : → ACK no is valid
PSH : → Req for burst
RST : → Reset
SYN : → Sync seq nos
FIN : → Terminate

window size : The recieving window size of the packet.

check sum : 16 bit check sum. mandatory.

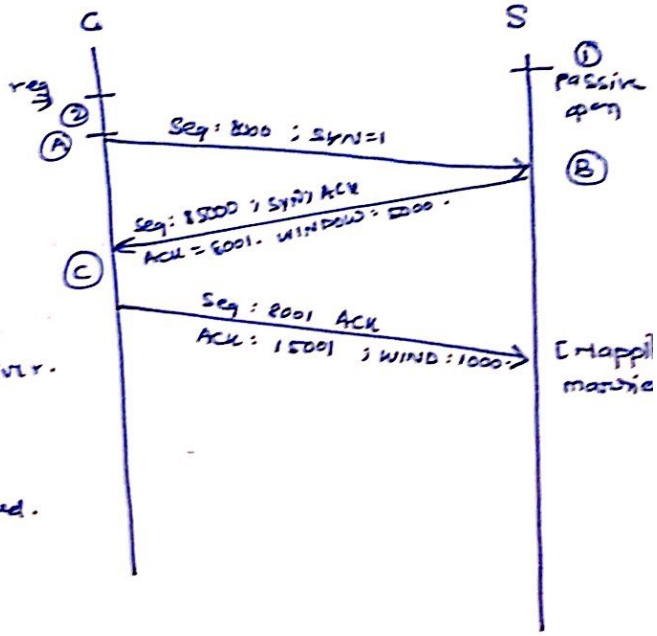
urgent pointer : 16 bit valid when flag is set.
The value to be add to seq no to get no of last urgent byte.

options : 40 bytes of control flow stuff.

TCP Handshaking (3way).

1) Server gets itself ready to connect : passive open.
2) client prog issues req : Active op.

→ Starts the handshake.



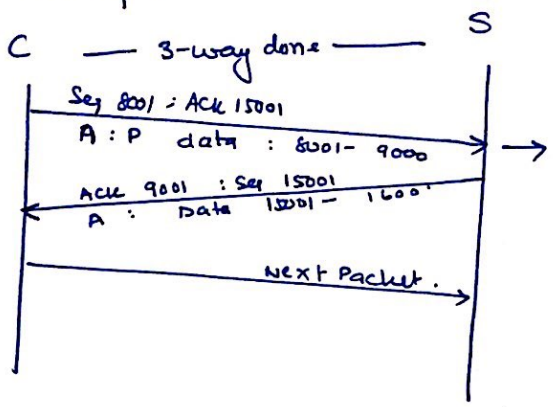
(A) Client sends a syn segment. with SYN = 1. To sync seq nos. the seq no is ISN.

(B) Server sends a syn+ACK it has its own seq no. this has the window size of receiving side server.

(C) The client acks the server with ACK flag. if the data is only ACK no seq is consumed.

— This can get SYN flooded by a client.
This is a dos attack.
sol^n : COOKIES !!!!

Data transfer

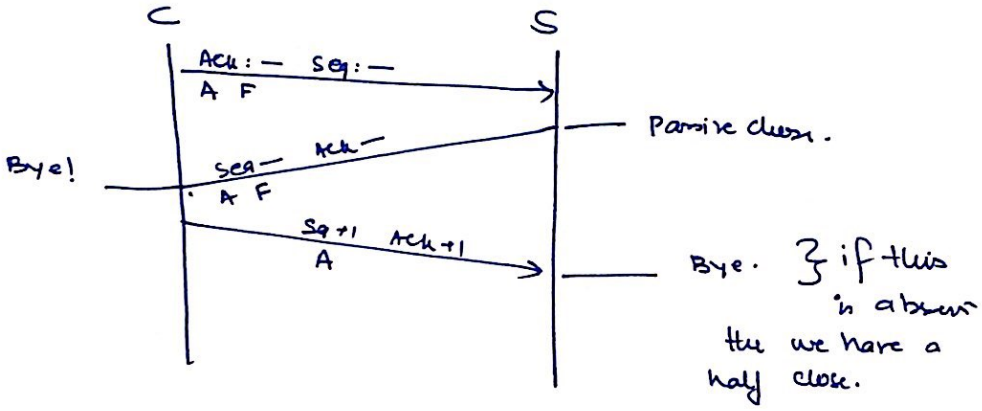


Push: Tells the TCP to push data to the prog immediately

Urgent: special someway.

Termination

→ can be a simple sent a F flag & get blank ack.
or 3-way.

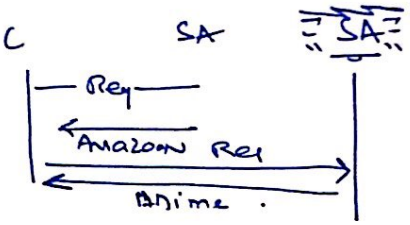


WWW

- A repo of info where web pages are dist all over the world.

  It is distributed and linked:

- THE linking is via Hypertext.



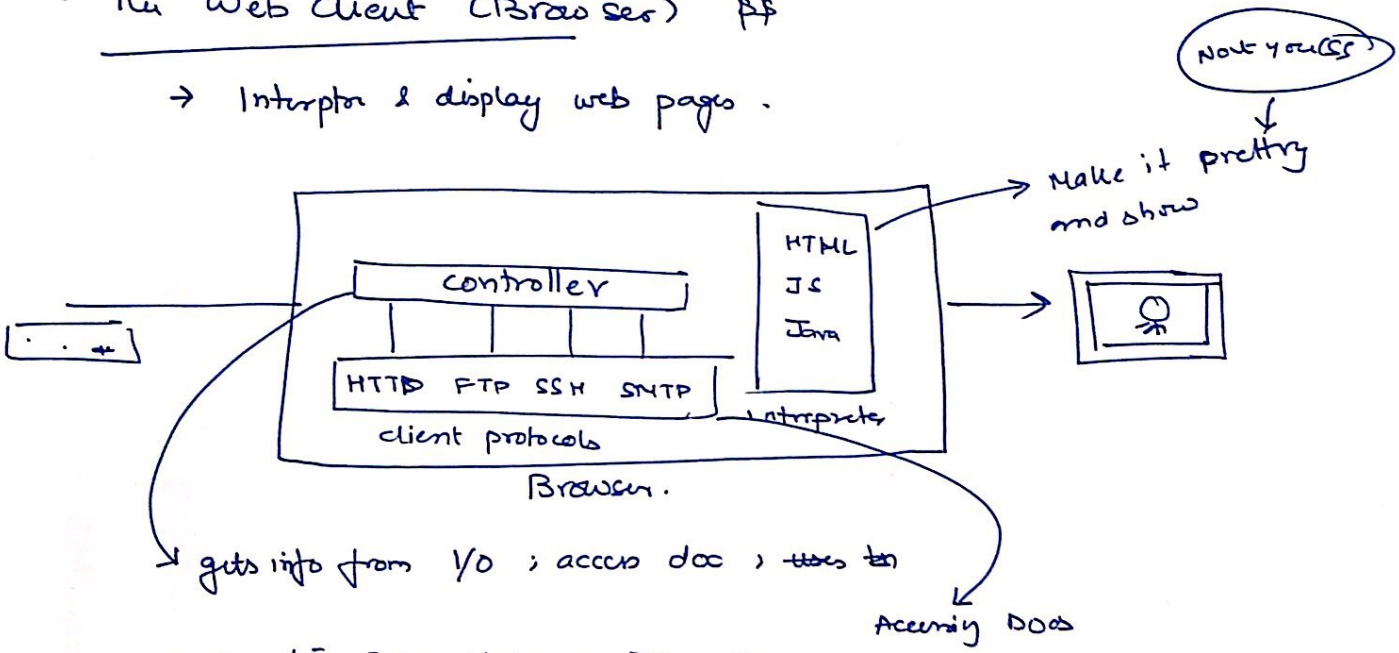
Architecture

→ It is a distributed client/server servic wher client with browser can access a servic using a server.

The servic is provided by -Sites.

- The Web Client (Browser) $$

→ Interpter & display web pages.



eg) IE, Edge, Netscape, Fifox. Cr.

- Web Server $$$$

→ They have the web pags, handle Req & then show the pages.

eg Apache ; MIIS.

Uniform Resource Locator (URL)

Protocol :// Host : Port /Path

Protocol : The type of protocol HTTP, FTP.

Host : IP or unique name of the server. IP can be X.X.X.X. or. Natural names. i.e Domain names

Port : The 16 bit app^n port. HTTP : 80

Path : The location of the file @ the server / site.

- Most common :

  protocol : // host / Path.

Web documents

- Static
- Dynamic
- actin

Static

- Fixed content.
- The copy of doc is sent when access.
- eg HTML, XML, XSL.

Dynamic

- created by the server and sent.
- The server runs the app, gets data and send.
- eg JSP, ASP, CGI.

Active

- The scripts!!!
- Progs that are continuously run.
- Applets / JS.

HTTP

Hypertext transfer protocol

↳ To access web page. (FTP + SMTP).

↳ TCP. @ port 80.

~~Persistent~~

non persistent:

→ each req / resp is a new connection
→ open TCP send req
→ get response & close.
→ read it EOL.

high overheads.

Persistent (HTTP 1.1)

- The connection is left open.
- closed after timeout or on req.

Message formats

→ Request / Response.

Req)

1) Req line. => 1st line.
   3 fields; method, URL, version
   ↓ Get, Head, URL, HTML vers.

2) 0 or more headers. have addtn info.

Req	Method | URL | Ver
Head	H name value crif
	:
Blank	crif
Body	═══

Resp)

→ Resp line. status:
   → version of HTTP
   → The status of request 101 / 404.
   → Phrase: status but in text.
→ Headder

version | status | phrase.

The 0 requests can be conditioned. eg last modified - only send if it was modified.

COOKIES (NO)

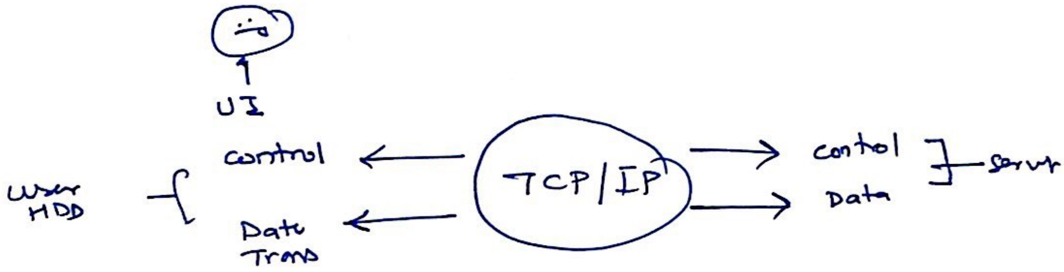
- The info stored about the session of user @ site.
- The cookie is in the rsp to client
- The cookie is stored by the browser.

when client sends req; Browser checks for cookies & if found adds it.

F.T.P

- TCP/IP for file transfer b/w comps. common to most data
- It uses two services, ∴ Two TCP connects

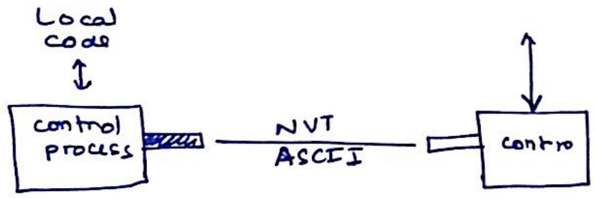
Port 21 for control
Port 21 for data.



control uses very simple rules. ; remains connect all the time.

data uses more complex rules. ; on need basis.

Control Connection



each code is terminated with \n & \o.

Every command gives on command. [### → Status, No → area. additional; text. Descr.

Data Connection

1) client issues passive open
2) client send port No to server
3) server issues a open Active.

→ FileType, DS, Mode, Transfer

Type : ASCII, EBCDIC, img
DS : file / record / Page structure.
file ↓ No structure; record → self exp; Page ↳ 4parts

Mode: stream / block / compressed
Transfer : Retrieve / Store / directory

SNMP

↳ Simple Network management.

↳ App^n level protocols.

↳ usable in Hetrogenous system.

→ Manager → runs SNMP.

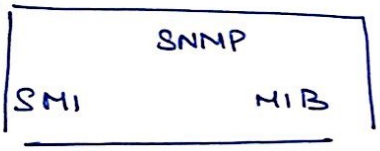
→ station ← managed.

Agent keep info on db wrt performance.

SNMP
- ↳ checks agents by req info
- ↳ forces an agent to do something
- ↳ agents contributes to manggement.

SNMP
- ↳ config → reconfig + document
- ↳ fault → reactive + proactive
- ↳ performance → capacity + Performance.
- ↳ accounting

Management



SNMP : gives the format of packet exchange b/w NMA.

It reads & changes the states of objects in SNMP Packet

SMI: gives the rules for naming & showing how to encode. Does not define the capacity

MIB: creates a collection of named objects, their types relationships.

Overview

↳ Manger wants to send a message to agent
- ↳ MIB finds the object that holds the info
  - ↳ SMI encodes the name of object.

· each object has a name, type & Encoding mode.

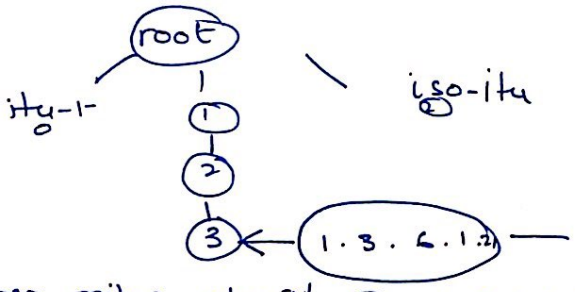
SMI

Structure of mgmt info.

Name

↳ each obj has unique name.
↳ uses object identifiers.
↳ Tree struct.

eg



SNMP uses mib-2 object id @ 1.3.6.1.2.1

Type

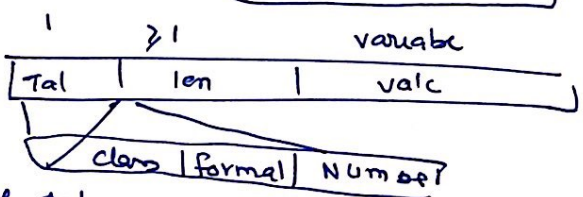
↳ simple
↳ structured

- IPAddr
- INT
- OCTSTR
- gauge
- BITS
- opaque

Sequence → heterogeneous Sq.
Sequence of → Homogeneous Sq.

Ecoding

↳ Basic Enc Rules.
↳ Triplet — Tag Len, Value



Tag : Typeof data

eg INT 02 IP 40 etc.

MIB

↳ MIB-2

↳ Each agent has its own MIB. The objects mgr can manage.

objects

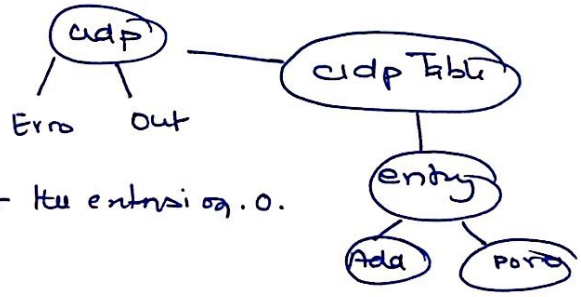
- ↳ sys — system : info of os
- ↳ if — interface info
- ↳ at — ARP Table
- ↳ ip — IP routing info
- ↳ icmp — ICMP info
- ↳ tcp — TCP info.

⋮

Accessing MIB vars:

simple var's : id of group followed by id of var

eg



The objects only define the vars.

To get instance info we must get the entry.0.

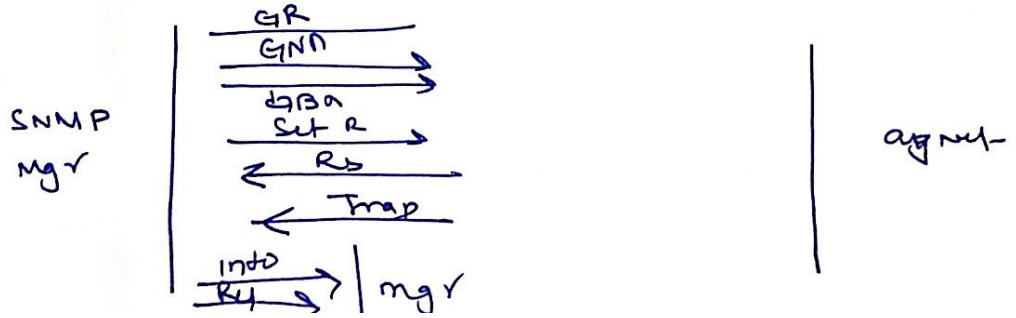
—Tables (.5)

SNMP

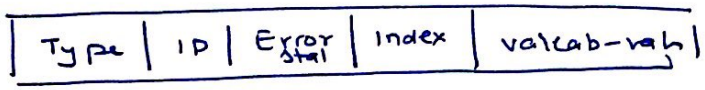
- Retrieve value of object
- store value to obj.
- send alarm messages.

V3 defines 8 types of protocols data units

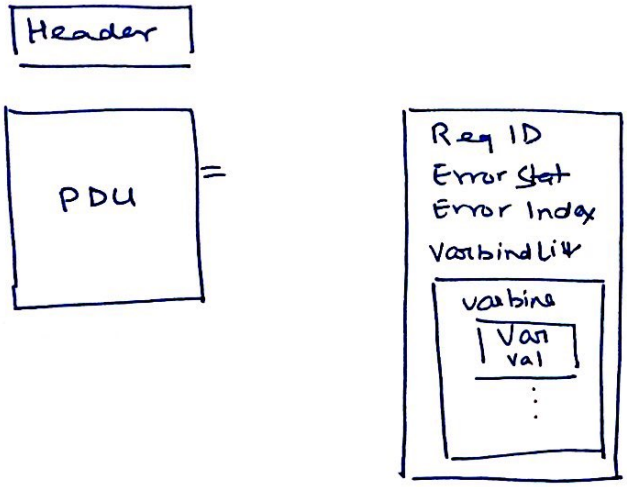
Get Req, GetNext Req, Get Bulk Req, Set Req, Resp, Trap, InformReq, Report.



format



Messages



UDP Port

↳ uses port 161 & 162.

Agent manager.

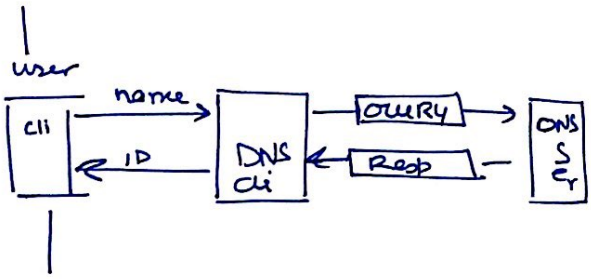
1) Passive opn

2) Active opn, exchang of info.

3) Trap.

DNS

- Dir that holds mapping from name to ID.



- Name space

- To be an amp the names must be carefully chosn from a namespace with complet control over binding.

- Flat namespa : name → address
  - Seq without structer.
  - any common set has no meaning.
  - cannot be used in larg e systems.

- Hierachical spaa : made of parts.
  - ~~not~~ The authoriti can be decentralized.

Domain namespace

→ To have a hierarchical spaa & The names are defined as inverted tree they comonly hav 128 levels.

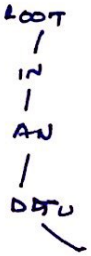
Label

: each node has a label. of 63 chars
: root is null.
: children have diff noms

Domain nam

: The seq of labels seprated by dots.
: if label is terminated with a dot => Fully qualified DN.

eg DTU. AC. IN.



Domain

: Subtree of DN space.

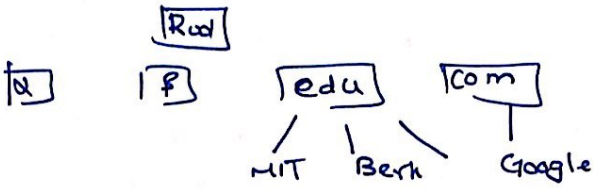
: name = name of node @ root.

Distribution of NS

The info in DNS must be stored. It is stored distributed.

Hierarchy

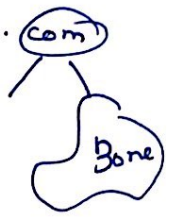
- DNS servers.



Zones

. The section of DNS with a name is its zone.

. Domain ≈ Zone unless there are subdomains.



Root

↳ Zone = whole tree.

→ has no info but delegates auth.

→ several roots each with complete duplicates.

Primary Server

→ loads all info from the disk file.

→ stores info as zone.

→ creates, maint & updates zone.

Secondary Server

→ loads all info from prime

→ does NO create or update.

DNS @ INet

→ Protocol for diff services.

3 sections → generaic, country, inverse.

generic
↳ have generic behavior .edu, .com, .gov.

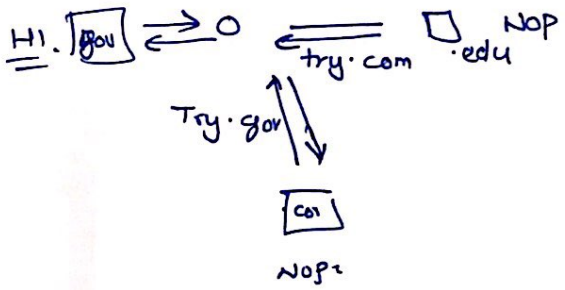
country
↳ two char county codes .in, .au.j.us.

Inverse
↳ RIP

Resolution
↳ mapping name to address.

Recursive
↳ sourc calls DNS to get ID,
calls another DNS
Root
calls the one than
sorvs
add

Iterative
↳ Sends the add of next server the host til.

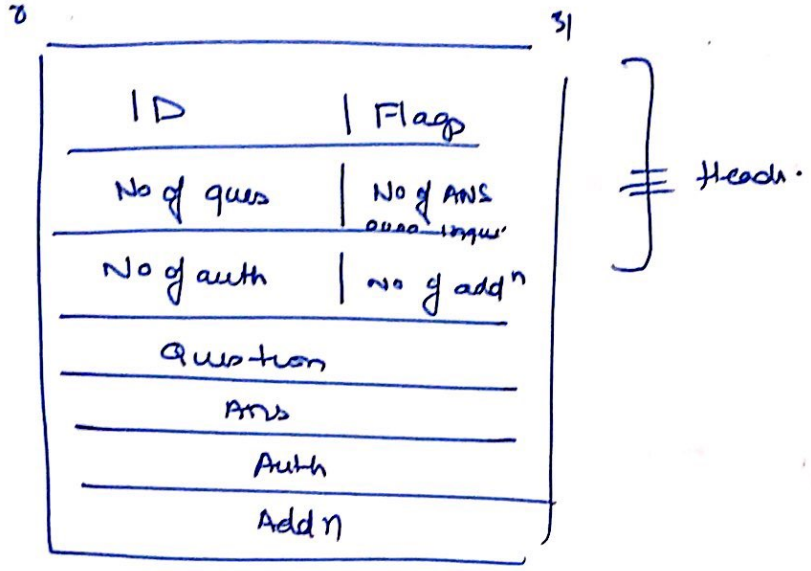


Caching -
↳ caches requests
↳ To prevent ancient replies a TTL is used.

DNS Messages

→ query & response

- same format → header
- header & quest



Registrars

→ They add new domain names.
→ accredated to ~~IANA~~ ICANN.
→ takes $ $

DDNS

Dynamic DNS.

→ The mapping is dynamic.
→ when the primary updates smt then 2nd are notified.

Encapsulation

→ uses UDP / TCP. @ well known port 53.
└ < 512 bytes.