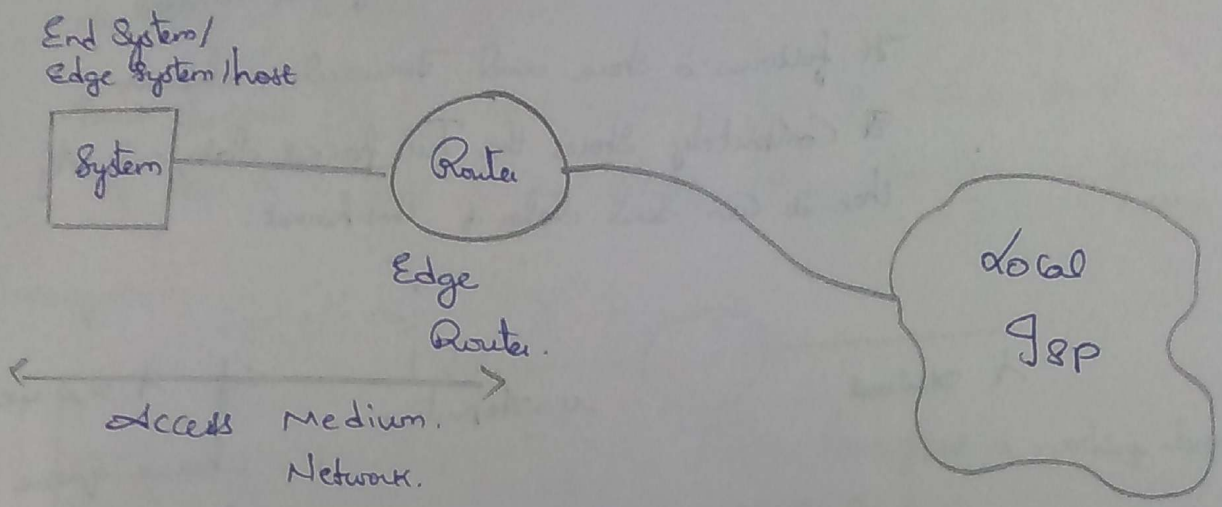
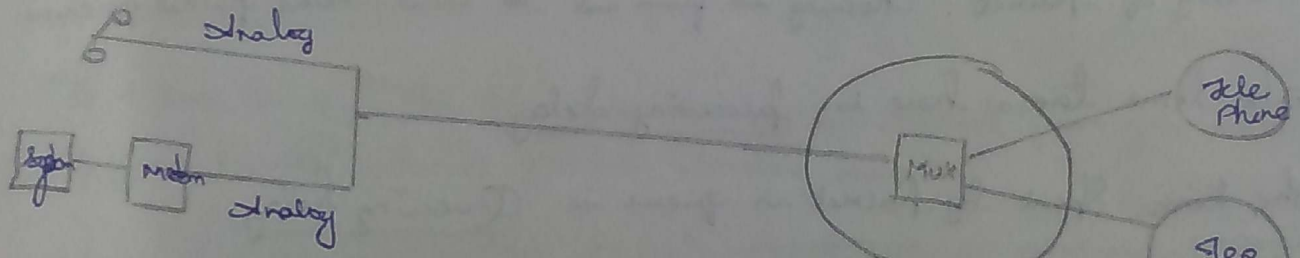
Assignments - 20%.
Mid Term - 20%.
End - 40%.
Project - 20%.

BOOK: Kurose - Ross.



Access Network:

1) Dial-up Connection:



At a time we can't send voice and data, and it is slow.

An improvisation is we use a Splitter near junction of Phone and System.

2) Cable network:

It is very fast but not so secure.

Message Sending:

- The host breaks the message into packets of length L bits.
- It transmits packet into network at rate R.

Packet Transmission Delay = L/R (L bits transfer)

SWITCHING:

1) Packet Switching: F.F.O whichever packet comes first is handled.

It follows a store and forward mechanism.

It completely stores the full packet data and only then it can send data of that packet.

if $\lambda > \nu$ we have queue and packets have to wait

$\lambda \rightarrow$ packet arrival rate

$\nu \rightarrow$ data flow rate

Processing of packet: checking its from and to and checks packet for errors.

Processing of packet: checking its from and to and checks packet for errors.

The time taken here is processing delay. (1)

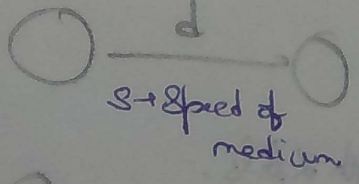
The time spent by packet in queue is Queuing delay. (2)

When packet, reaches front-end of queue, it will be transmitted to medium. The time taken for it is called Transmission Delay. (3)

If packet has α Kb and data rate is β Kbps

$$T.D = \alpha/\beta.$$

Then it should be taken to destination by medium.



Propagation delay (4) $= d/s$.

Propagation Delay: The time spent (in medium) for reaching destination.

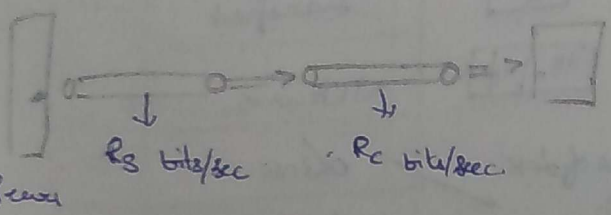
Each packet half-cycle (one way from Sender to Reciever) has

1) Processing Delay 2) Queing Delay 3) Transmission delay 4) Propogation Delay.

HOW PACKET LOSSES OCCUR:

1. The queue at router is full and a packet request comes. It is lost as queue is already full.
2. Each packet has a Time out time / TTL → Time To Live.
   If this time gets out in delay. then packet is Lost.

THROUGHPUT:



The throughput is nothing but min(Rs, Rc).

A chain is as weak as its weakest link.

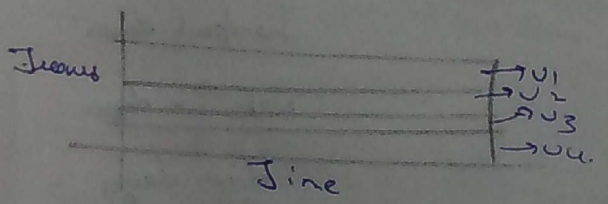
2) Circuit Switching:

If 1 sends to B, the path from 1 to B is reserved from your data-communication. If your packet gets lost you are wasting resources.

And also at every instant packets will not use all reserved resources.

FDM → Frequency division Multi-plexing.

We divide frequencies into slots and allocate to various users.



TDM → It is just the same as Round Robbin.

Most of time packet Switching is used, but only in need high data security circuit switching is used.

Rather than having $N C_2$ connections, we have Global ISP's and every network is connected to some other ISP.

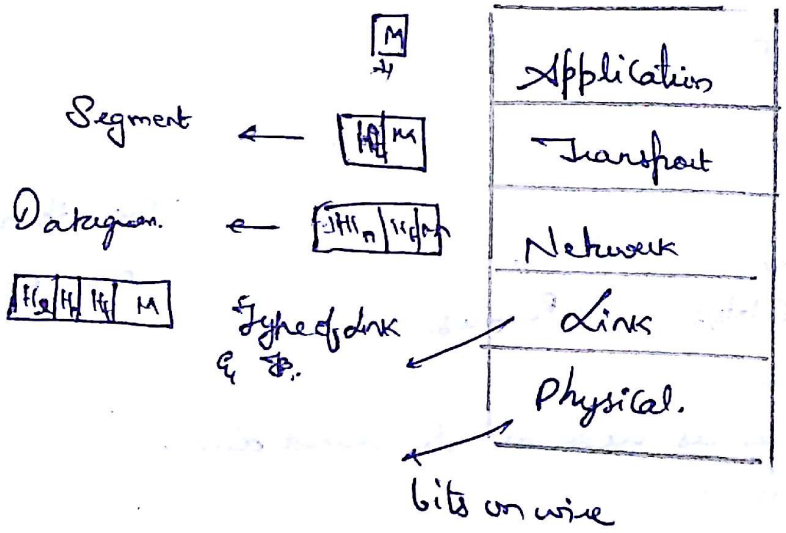
Layering:

The changes done to message for ensuring organisation.

The Internet Protocol Stack:

H_t → Header of Transport

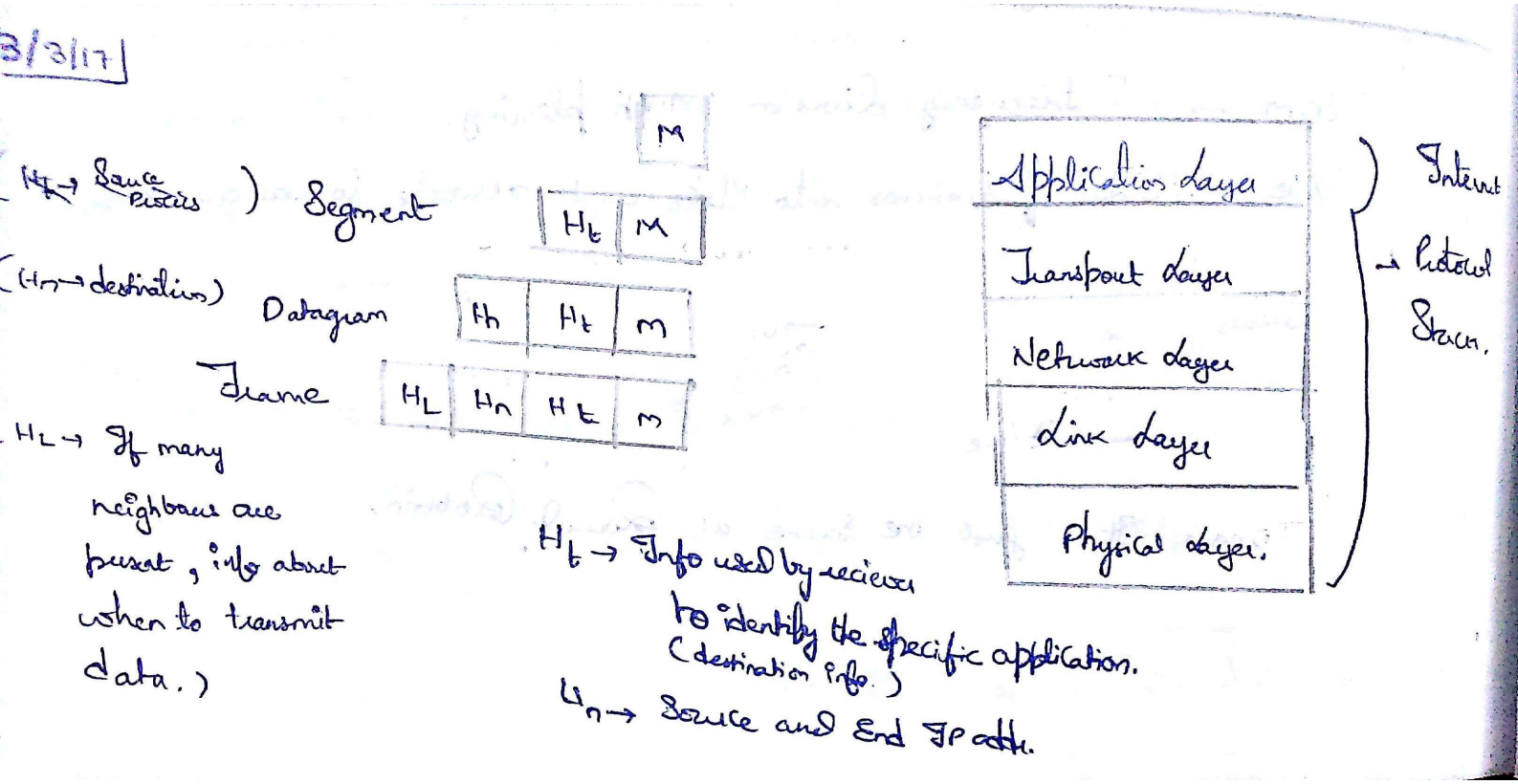
H_n → Header of Network



If M is message,

Doubt: H_e → ?

3/3/17



(Hs → Source Address)

(Hn → destination)

HL → If many neighbours are present, info about when to transmit data.)

Ht → Info used by receiver to identify the specific application. (destination info.)

Hn → Source and End IP addr.

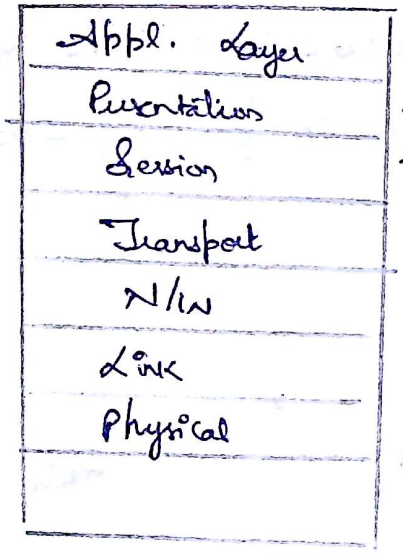
Application Layer → Support n/w application Eg: HTTP, FTP.

Transport Layer → Runs to Process Eg: TCP/, UDP

Network Layer → End to End Eg: IP, routing

Link → Transmission schedule Eg: 802.11

There is another stack, which is OSI → Open Systems InterConnect.

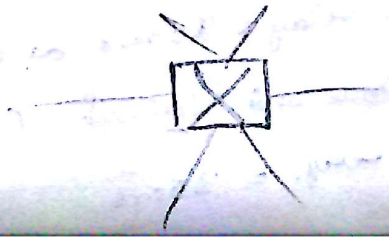


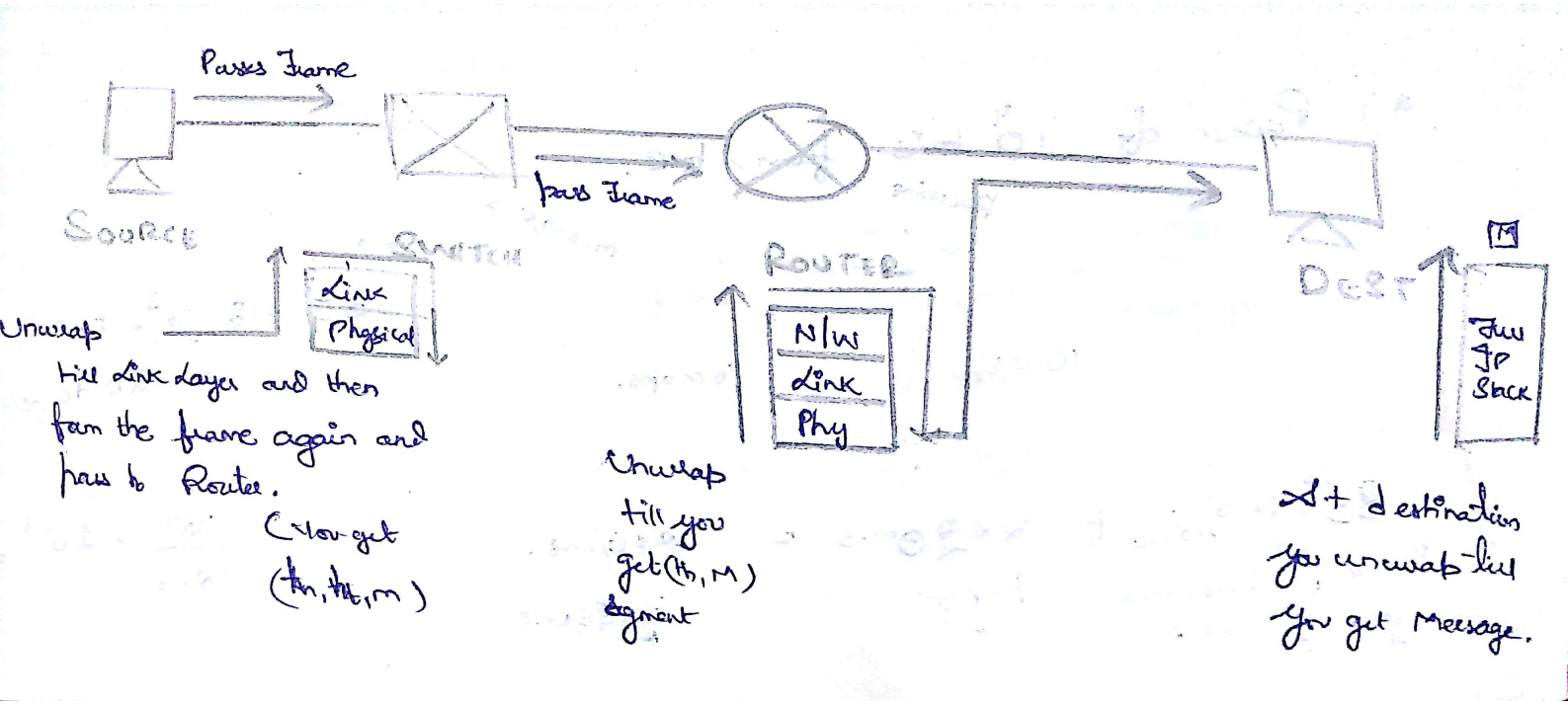
Two optional layers which account for encryption, decryption etc.

OSI

Router

Switch





E.g.: 1.

R_1, R_2 are rates in media and α is packet size. What is transmission Delay?

$$T.D = \alpha/R_1 + \alpha/R_2.$$

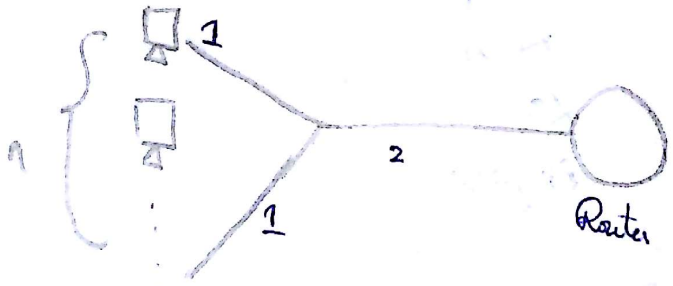
E.g: 2

There is a 2Mbps link shared between users who transmit at 1Mbps. If packet switching is used. Does queing delay occur in following cases.

a) $n = 2$

b) $n > 2$

c) $n < 2$



Ans: There is no queing if $n \leq 2$. If $n > 2$ we have queing.

E.g:3 A and B are linked through a router R via 10 mbps links.

Propagation Delay = 20 m.s at each link

E.g:3 A and B are linked through a router R via 10 mbps links.

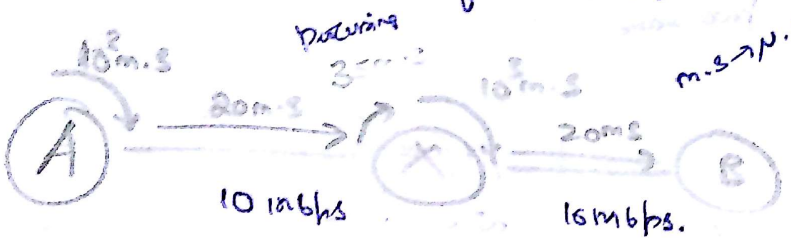
Propagation Delay = 20 m.s at each link.

R is Store and Forward (wait till packet arrive full packet is got).

Processing Delay = 35 m.s

Time total = ?

a) Packet of 10^4 bits from A to B.



→ $35 + 2\times10^{3} + 2\times20$

$= 2075$ μ.s.

$35^{ms} + 2\times10^{4}/10^{6}\times10 + 2\times20$ m.s = 1090 ms.

Processing at A — Transmission at A — propagation — 2075 μ.s

$\frac{10^4}{10^6\times10} = 10^{-3} = 1$

Two packets of size 5000 bits immediately after each other.

10 Mbps → 10^7 Bits/sec.

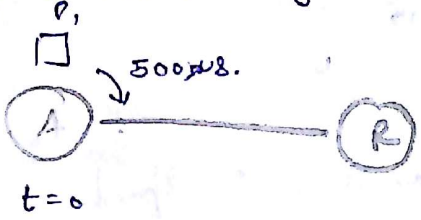
$T.D = \frac{5000}{10 \times 10^6} = 0.5 \times 10^{-3}$

$= 500 \mu s.$

*

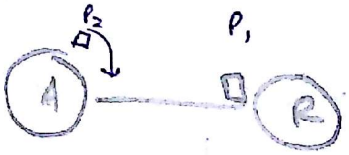
At A No processing delay (Source).

At B, No processing delay (Dest.).

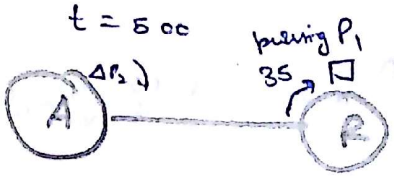


t = 0

t = 500



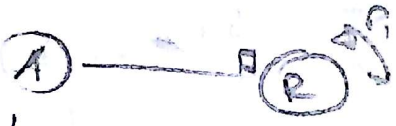
t = 580



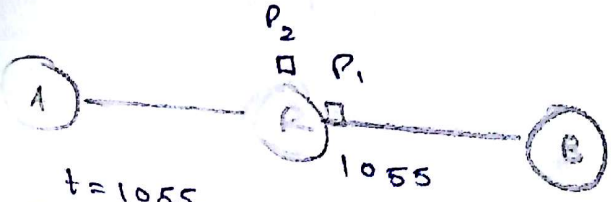
t = 555.

t = 1000

P_2 transmitted P_1 is transmitting.



t = 1020



t = 1055.

P_2 ready to Transmit and P_1 ready to propagate.

t = 1075.

P_1 has reached.

t = 1555.

P_2 is transmitted.

t = 1575 μs

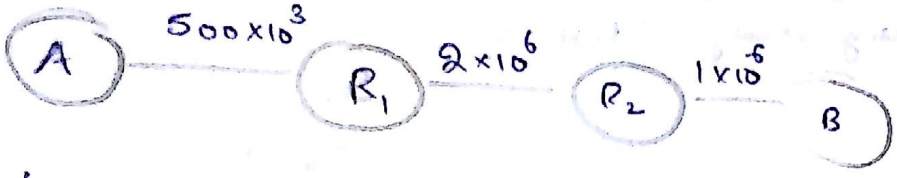
Both packets reach dest.

Q: 4.

A to B has 3 lines of rates $R_1 = 500$ Kbps $R_2 = 2$ Mbps $R_3 = 1$ Mbps

a) What is throughput

b) If file size is 4 million bytes . T.D = ?



a) Throughput is 5×10^5 bits / second as it is min speed.

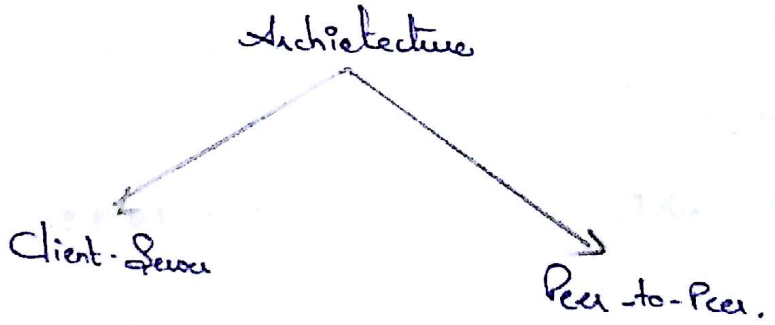
b) T.D = $\min(TD_1, TD_2, TD_3)$

$= \frac{4 \times 10^6 \times 8}{5 \times 10^5}$ (bytes → bits) (? doubt)

$= 64$ sec.

7/3/16

Ch-2: APPLICATION LAYER.



1) Client - Server:

Server

- Server should always be ON.
- Fixed IP address
- Data Center (Storage..)

Client:

- Needn't be always ON.
- Can have a dynamic IP.

2) Peer to Peer:

- No need always Server ON.
- The ends Communicate directly.

IPC in OS happens via, Shared Memory (or) Message Passing (through Kernel.)
Since it is internal to a System.
But now we want process Communication across two Systems.

- So, we have IP for System Identification and Port for process Identification.

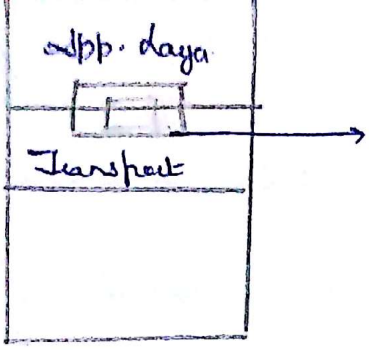
Services required by Certain apps:

1. Data Integrity → 100% reliable data transfer.
2. Timing → Some apps have Strict Low timeout.
3. Throughput (bits/sec) → Some need high throughput (Videos etc.).
4. Security

[See Slide Q-19]. UDP → User Datagram Program Protocol.

TCP doesn't provide timing and throughput guarantee always, but ensures Security
↳ It is a Strict one where it expects for link to establish and as long as transmission occurs, link has to be there.

UDP: It doesn't care about Connection Status, Security etc.
It just sends messages.
It ensures timing, but reliability and Security are at Stake.



Socket: It is a virtual door from App. layer to Transport layer, where protocol is decided.

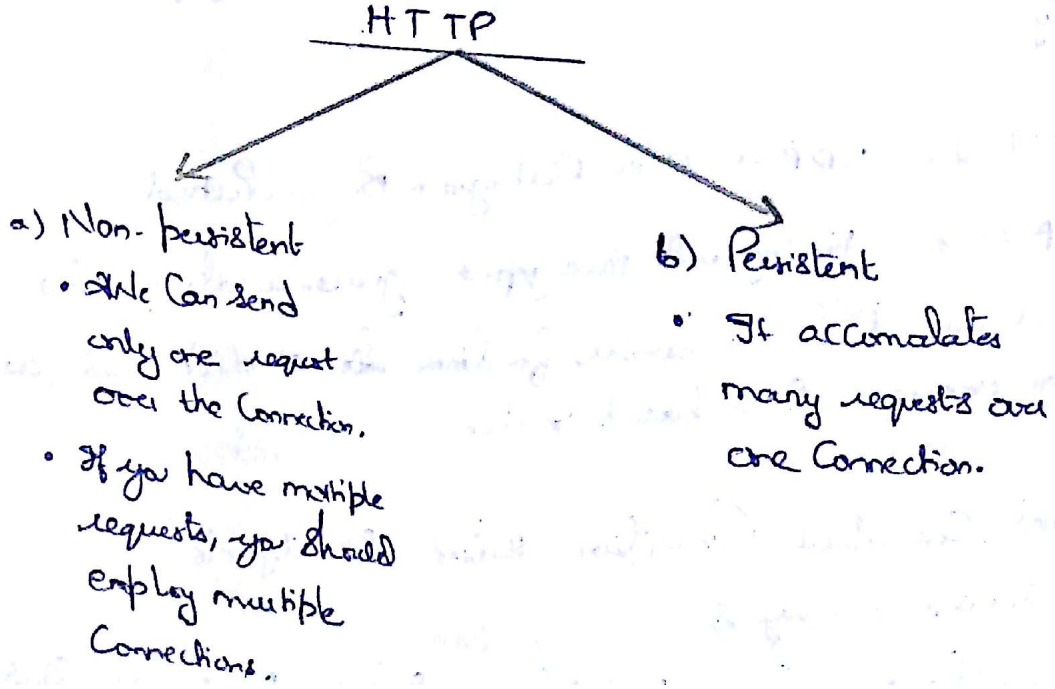
HTTP:

1) Transport layer is TCP
   - First we request Server for a Connection
   - Then we get a response
   - Once Connection is established, client requests for Content transfer.
   - Checks if data is present and Sends data.
   - Immediately after data transfer is done, Server is closed. (client link)

HTTP is called "Stateless" as it does not have State-related info of Client.

Advs: • Simple System design.
• High-performance.

Disadv: • Since no client info; overloading may occur due to multiple requests.



RTT → Round Trip Time. : Request from Client to Server and response from Server to Client. (time)

1RTT + 1RTT + Transmit Time = Total Time (Response Time)

- 1RTT → requesting a link and response from Server
- 1RTT → Your data request and message from Server saying whether it has data or not.
- Transmit Time → Time taken to send the file data.

This Response Time is for Non-persistent.

For Persistent:

Response Time = 1RTT + Transmission Time.

[See slide 2-32]

Methods:

1) GET → Normal request for a Page.
2) POST → Generally used for forms. The input is sent to Server.
3) HEAD → Similar to GET, but used by programmers for debugging.
4) PUT → uploads files to path in URL.
5) DELETE → Can delete data on Server.

} → 3] → only in V 1.1

See Status Codes of Response in 2-36.

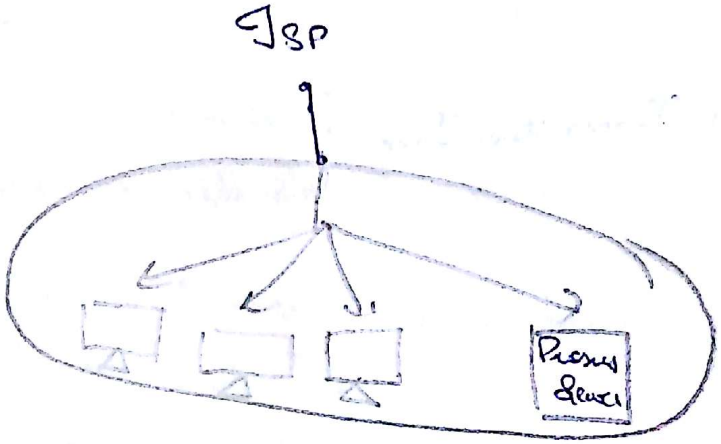
Cookies: Web Server gets info of Client by placing Cookies on host/client.

[Slide 2-38].

Cookie → It is a workaround to overcome statelessness of HTTP.

- When a new request to ~~new~~ site comes, it sends response along with a Cookie id and data, which is stored locally.
- Thenon, if you send request again, Cookie is also sent and they can customise.

Web-Cache : (aka Proxy Server)



Institute Network.

Anyone who requests for data will ask for it to Proxy Server.

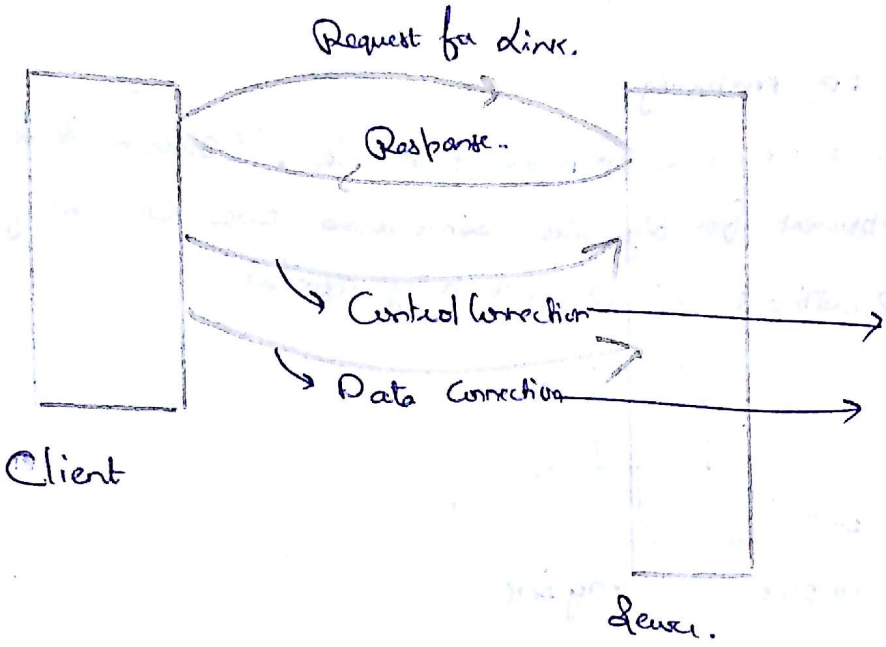
Proxy Server is like Cache (Stores most frequently used and Recently used.)

Cookie Features:

- Starts with response from Server.
- Thenon, each request has a Cookie Id too.
- Locally a CookieFolder is Created.
- Back-End of Server requested has info by Cookie-Id.

Conditional GET : Get only if Contents are modified by Web Cache.
To this, the Server responds by saying No or by sending changes that occur.

FTP: File Transfer Protocol. FTP uses TCP for Data Integrity.



To execute Commands on Server.
To Transfer data.
(Opened when Server recieves File Transfer Command.)
Once a file is transferred data Connection is closed.

FTP: Out of Band Connection as we have more than one Connections from Client to Server, unlike HTTP.

SMTP: Simple Mail Transfer Protocol. Port 25.
It is a persistent Connection → i.e once you make a link, it stays till the data is done.
HTTP → pull based SMTP → push based. (read what is pull-based and push-based.)
It has a few Conventions:
Header should have • To, • From, • Subject of Email.

* Read about POP3 (missed it ; also IMAP.).

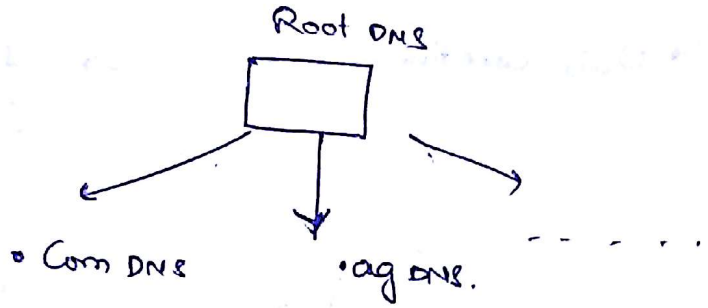
DNS: Domain Naming System.

Basically its a mapping of IP → Name and Name → IP.

It is a server handling high load (and crucial. So, it should map efficiently).

Services of DNS:

- Hostname ↔ IP mapping
- Aliasing (give same name to more than one, (since a server may not be sufficient for big sites. And also we can change internally IP without external not being affected.



So, firstly we check at Local DNS Server, which has a cache of addresses IPs recently used. If present, it will send you.

Else it will forward to Root Server.

Then Root Server will forward accordingly. (to .com / .org DNS ...).

RR is entered to DNS Server in this format. RR → Resource Record.

(name, value, type, ttl).

- value → IP
- type → type of query
- ttl → time to live. (how long you will have this record in Server.)

types = [A NS, CNAME, MX] → Google and learn.

DNS Query and Reply have same format.

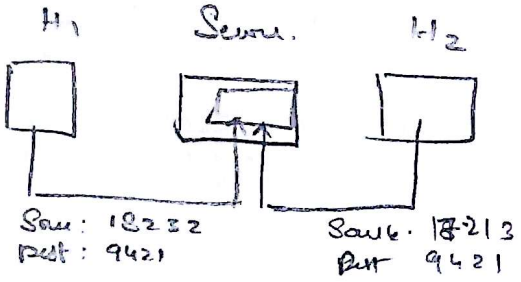
Multi-Plexing : The job of gathering data chunks with header information to Create Segments and passing the Segments to network layer is called Multi-plexing.

De-Multi-plexing : (See def. at Transport layer).

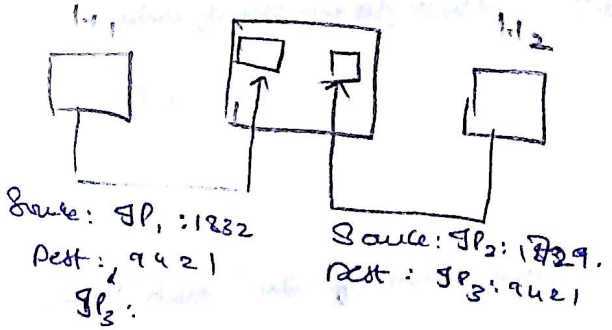
- Each datagram has a Source IP and also a destination IP.
- Each " has one transport layer Segment.

UDP:

1) Create a Socket
2) Port no to be added in Segment
3) Send.



UDP Mode both go to Same Socket.



TCP mode, different Sockets are Created.

TCP:

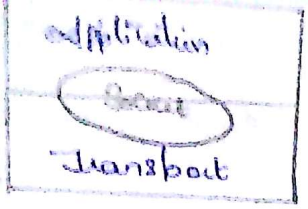
Hand Shaking / Establish Link.

Rest is UDP.

- It has 4 tuple Socket identification.
  - Source IP
  - " Port
  - Dest IP
  - Dest Port.
- It Creates a different Socket for each different Source request even if dest. port is Same which is not the case with UDP.

UDP → User Datagram Protocol.

It Creates the Segment [H|Ht] and does error-checking and Sends it.



M

M | Ht Segment. After this it does an error check.

Features:

1) We have application level control over data.
2) No Connection establishment.
3) No Sequence of State of message is required.
4) Small packet overhead as Header is Small.

Uses:

1) Multi-media
2) DNS
3) SNMP
4) Routing Tables

The UDP header:

Source Port	Dest. Port
Length	Checksum

Length → The length in bits of data.

Checksum:

Sender: Treat all the data as 16-bit integers. Take Sum of all Such 16-bit integers (including headers).

Carry around Carry i.e add Carry to Least Significant one.)

Take two's one's Complement of Sum and Send it as Checksum.

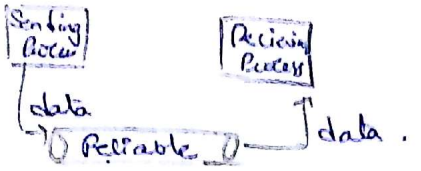
On reciever side you take Checksum and add the addition of 16-bit Chunks. The output should be all 1's, else a problem.

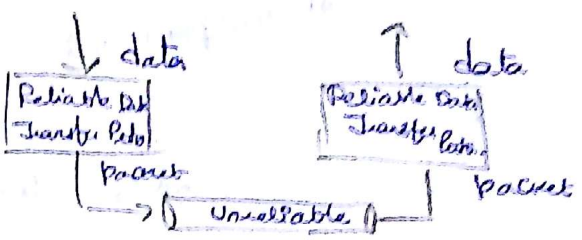
NOTE: It indicates of error but rectification is not present.

Principles of Reliable Data Transfer:

1) Reliable channel: A channel which doesn't induce any data-loss.

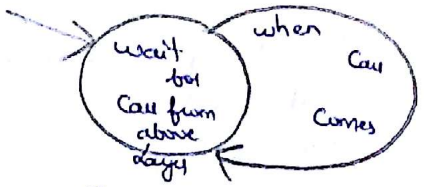
The Complexity of Reliability protocol & unreliability of channel.





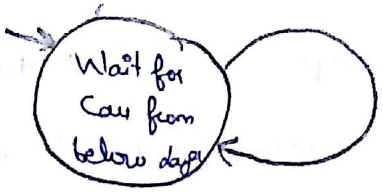
The FSM's of RDT → Reliable Data Transfer.

Case 1: Reliable Channel: RDT 1.0



a) rdt_Send(data)
b) pkt = make_pkt(data)
c) udt_Send(pkt).

Sender's Side.



a) rdt_recieve(pkt)
b) extract_packet(pkt, data)
c) deliver_data(data).

But, this case is hypothetical as Reliable layer doesn't exist in reality.

Case 2: RDT 2.0

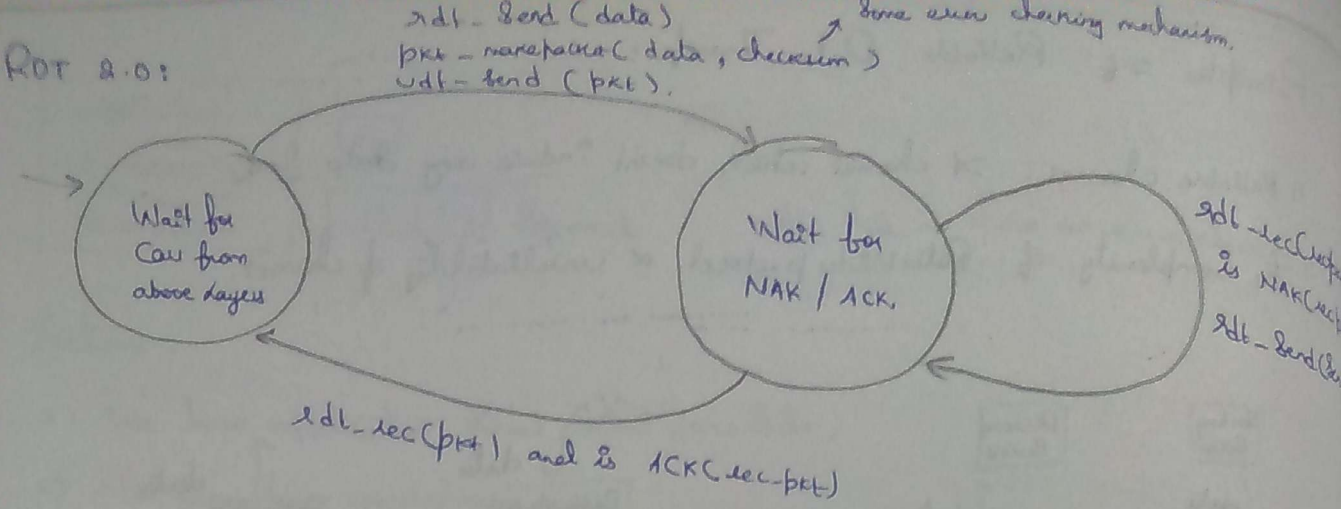
Automatic Repeat Request (ARQ protocols): When NAK is got Sender has to repeat again.

1)
- If you get ACK send new packet
- If you get NAK send the old packet again.

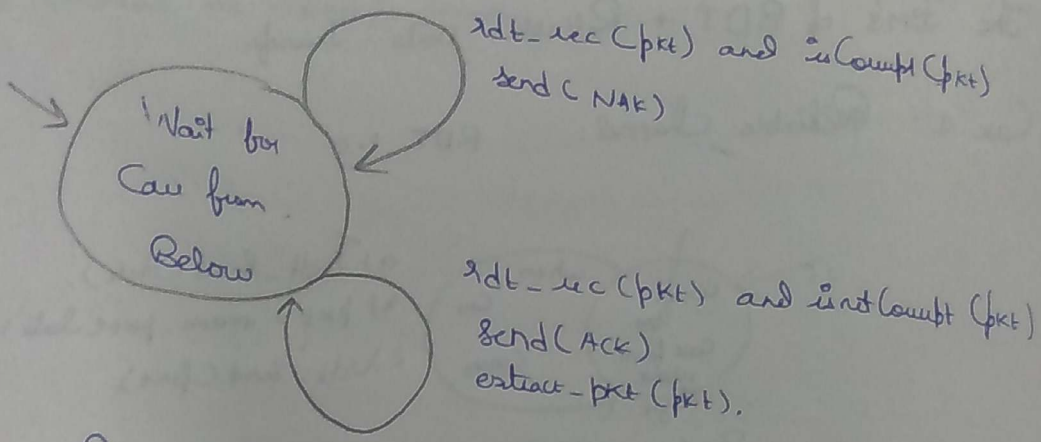
2) Error handling.

some error checking mechanism.

ROT 2.0:



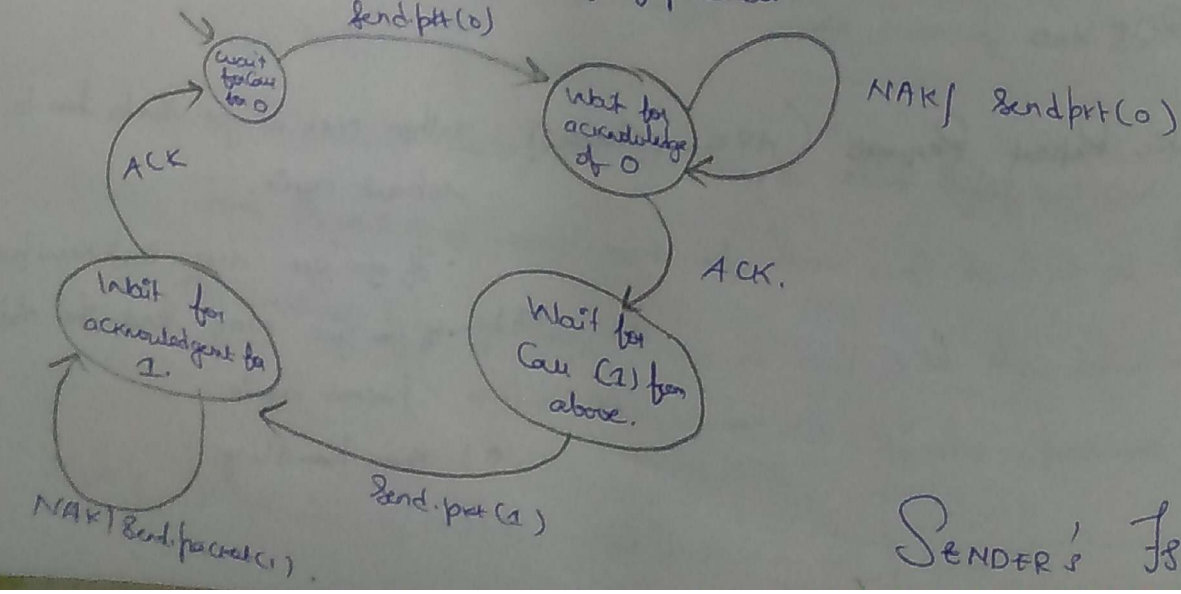
Sender's FSM:



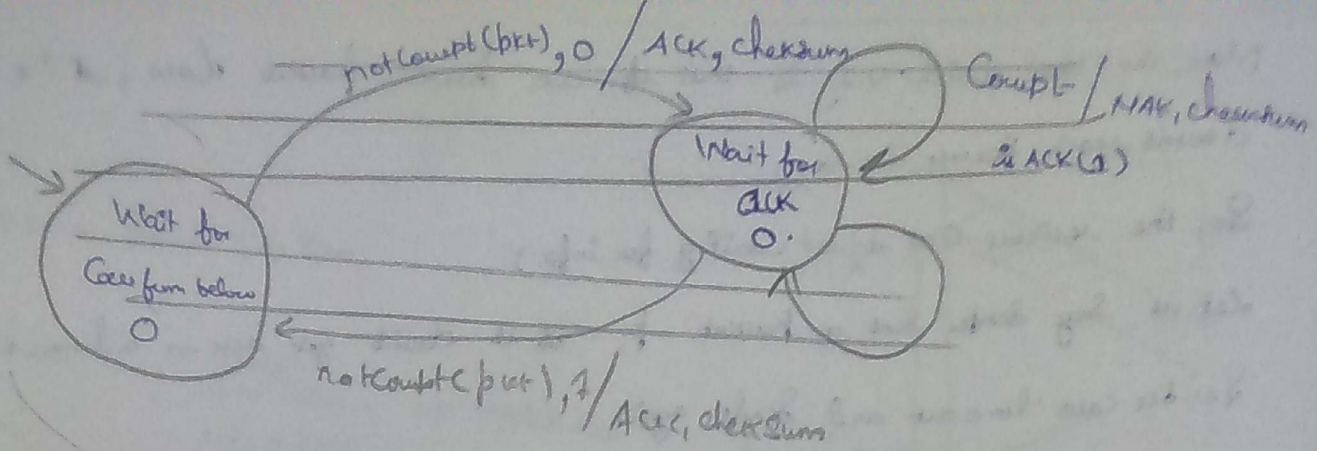
Reciever's FSM:

This type of protocols are known as 'Stop and Wait' Protocols (where the Sender's next Step is dependent on input / acknowledgement from Reciever.).

ROT 2.1: there is some Sequencing of packets.



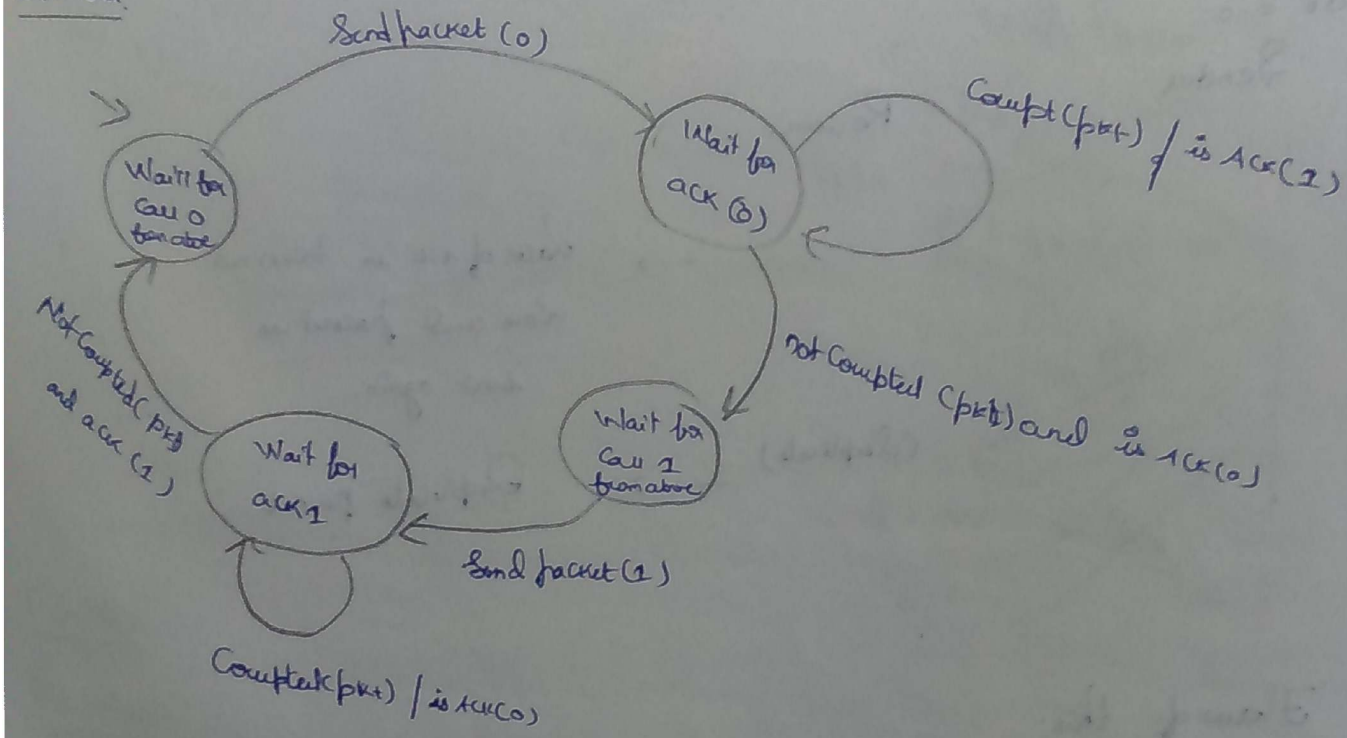
SENDER's FSM



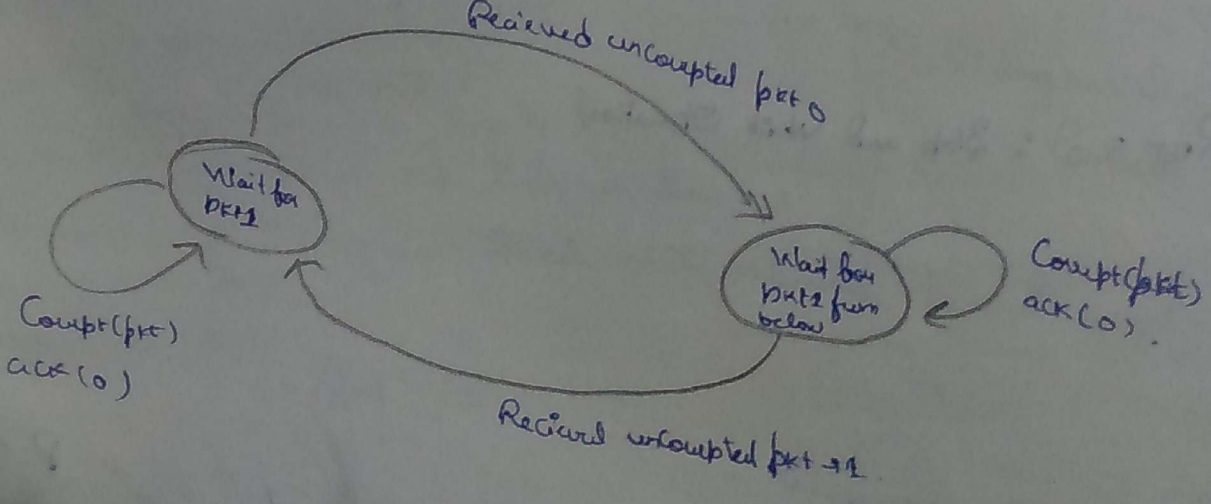
11/3/17

ROT 2.2:

Sender:



Reciever:



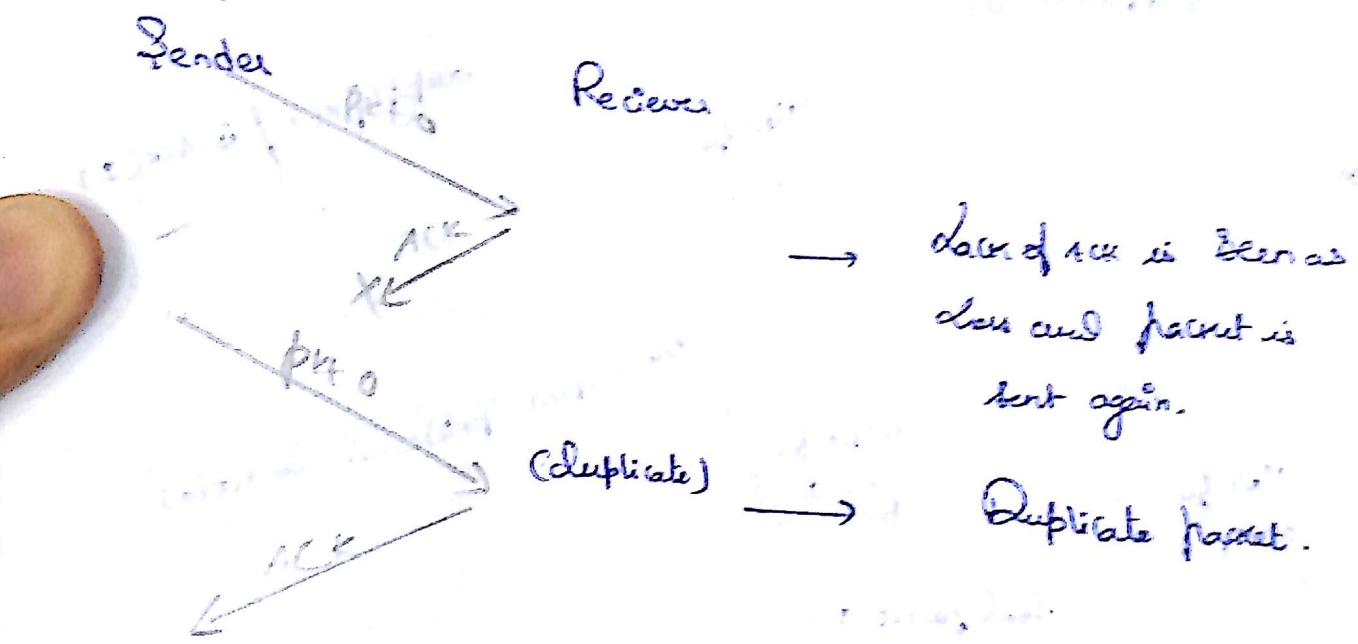
Now, the issue with 2.1 is that if we have packet loss, it doesn't work because at all.

So, the receiver (as it is waiting for info.)

Let us say Sender sent a packet P; If it doesn't get ack in a time, Sender will time out and Send again.

If there is an acknowledgment loss then the Sender sends again as in previous case. So, the Receiver has 2 packets same (Duplicate). It assumes that it was due to ack loss and therefore discards packet and Send Ack.

Rdt 2.0:



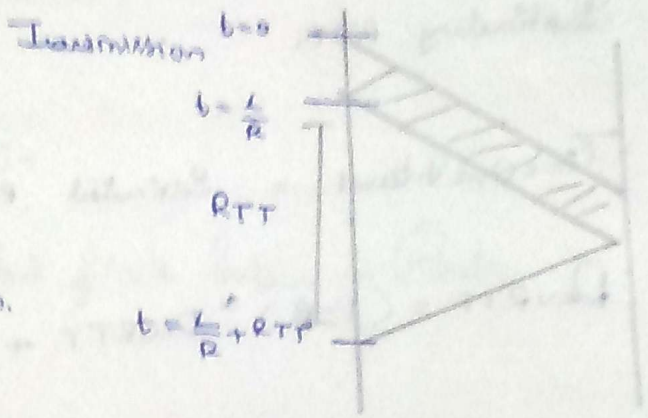
Flaw of this:

If ACK is delivered with a delay, it leads to issues. (i.e after timeout)

ROT 2.0 : Stop and Wait Operation:

RTT → Round Trip Time

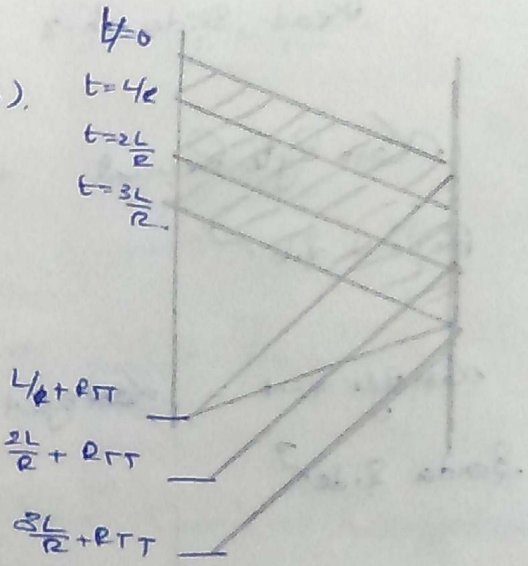
$U_{under} = \frac{L/R}{L/R + RTT}$ → Utilisation.



The flaw here is that until we get ACK, we don't send any packets.
To avoid this we use pipeline.

→ $U_{Under} = \frac{3L/R}{L/R + RTT}$ (we are transmitting 3).

With Pipeline.



Pipelines:

Go Back - N:

- Sliding window protocol.
- Let us say your Pipeline Under = K.
- You maintain a window initially first K.
- Each time you get an ACK slide window by 1.
- Whenever ACK is timedout. Send all the N packets of window again.

Sliding Window Protocol.

Selective ... → Read.

TCP: (Read till RTT)

Estimating RTT.

Variation (S-D)

TimeOutInterval = EstimatedRTT + 4 * DevRTT.

DevRTT = (1-β) * DevRTT + β| .....

TCP flow control:

We have buffers on both sides (Senders and Recievers).

→ Stopped writing in this last from here. Too much Too much.
Read Slides ch-3 from 3-40 → end ←

LastByteRecieved - LastByteRead ≤ RecvBufferSize.

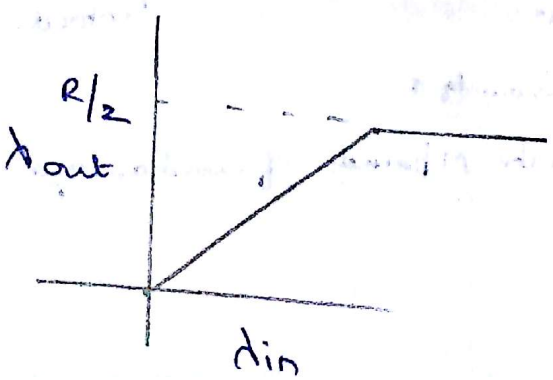
Reciever side ↗

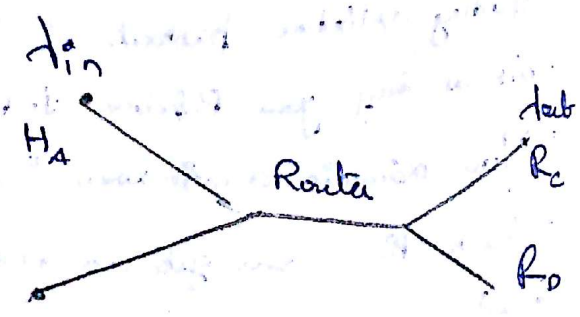
LastByteSent - LastByteAcked ≤ Rwnd.

Sender side ↗ Rwnd → Recieve window.

24/3/17

Congestion: Too many Sources sending too much data, too fast.

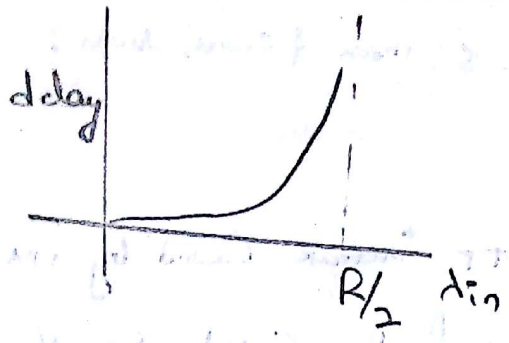




Router has ∞ buffers and output link Rate = R.

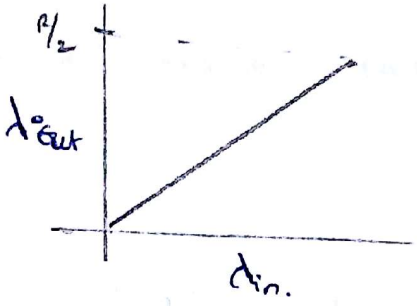
= R/2 for each.

Delay:



Second Situation: Same as first, but finite buffers in Router.

- Sender:
  - Makes a Copy of packet to be Sent
  - Send packet only if free buffer space is present in Router.



This is so as we send only until buffer is free.

Approaches of Congestion Control:

Note: Getting feedback from Router is not so ideal.

End-End Congestion Control:

- Don't get feedback from Router. (we have Cond → Congestion window)
- Infer get from end-System loss, delay. (By means of a parameter in Header.).

Network-assisted:

- Router provides feedback → Single bit for Congestion and rate for Sender to Send at
- We use CHOKE PACKETS.

Tcp Congestion:

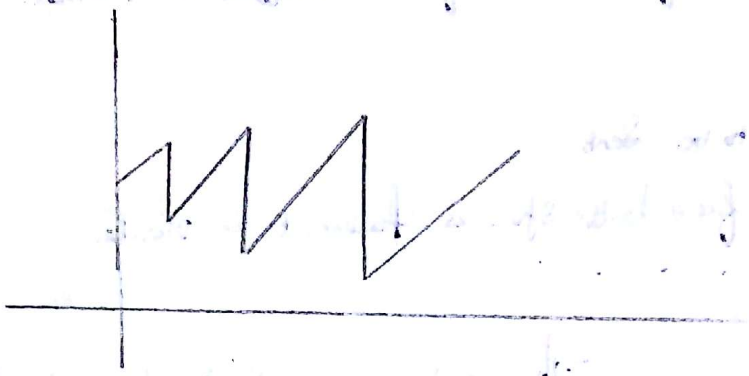
- If you get replicated acknowledgements (means receiver is not getting new packets.)

LastByteSent - LastByteAck → (≤) min { Cwnd, Rwnd }

1. double.

Additive Increase: After each RTT increase Cwnd by 1MSS.

Multiplicative decrease: If loss is found Cwnd = Cwnd/2.



Guiding Principles for TCP:

1) Lost Segment => Congestion => Sender's rate should be dec.

2) ACK <=> Everything is fine.

3) Bandwidth Probing.

TRANSPORT LAYER IS DONE:

CH - NETWORK LAYER

Routing: Determine Route taken by packets from Srce to Dest.

Forwarding: Move packets from Router's input to appropriate output.

Network Layer Service Model:

- Individual guaranteed delivery.
- " " " with bounded delay.
- In-order delivery of flow of datagrams.

Datagram
↓
Connectionless

Virtual - Circuit
↓
has Connection.

VC implementation:

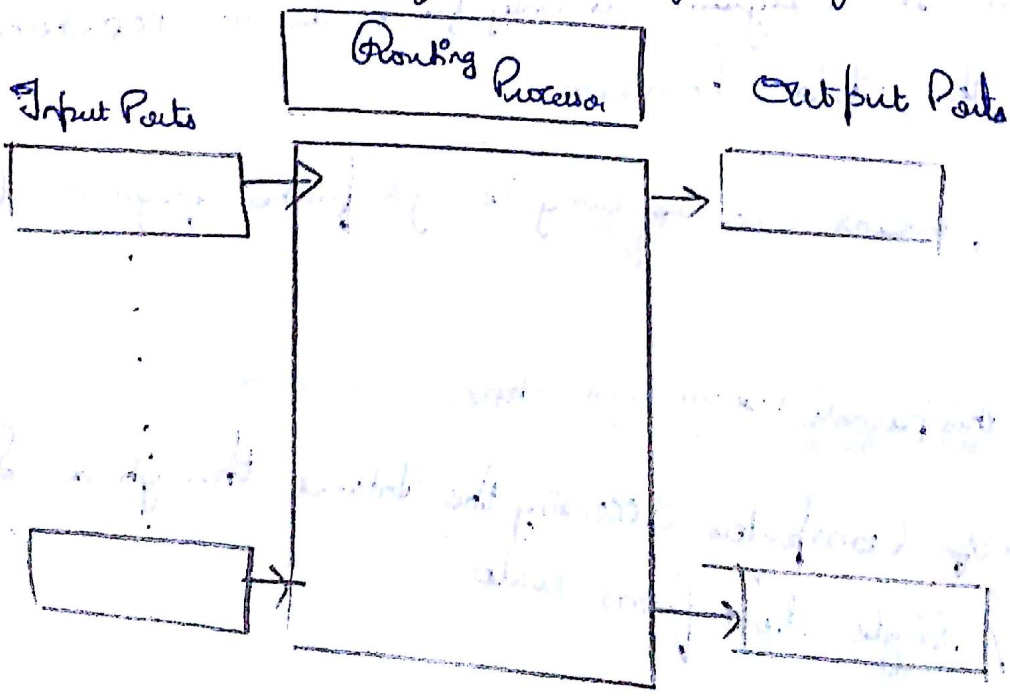
- Path from S → D
- VC numbers → one for each link.
- Entries in Forwarding table.

First establish a path (Initiate Call, Incoming Call, Accept Call, Call Connected, data transfer.).

Datagram Network: Doesn't bother about whether Connection is established. As soon as it gets, it sends.

Routing Architecture Overview:

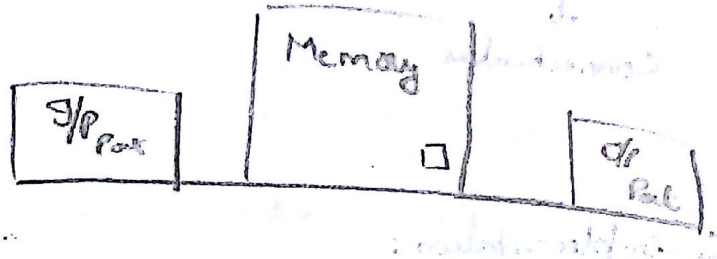
- Run routing algo
- Forwarding datagrams from incoming to outgoing link.



5/2/11 Read what Switching means exactly, and forwarding Table.

Switching via Memory:

The switching is done with help of Memory.



The bandwidth is determined by Memory Speed.

Switching via Bus:

We have a Common bus. It transfers data from input to output.

The speed is increased but at a time one packet/data only.

Cross Bar Switches:

The packet is split into many small packets and is sent over various grid choosing wisely. This is Mostly used.

The three major parts of Networking layer are:

1. IP Protocol 2. ICMP (control) protocol 3. routing algorithm.

The checksum in IP datagram is only for header as TCP checksum takes care of the data Checksum.

fragment flag = 1, means we are going to get further fragments of this packet.

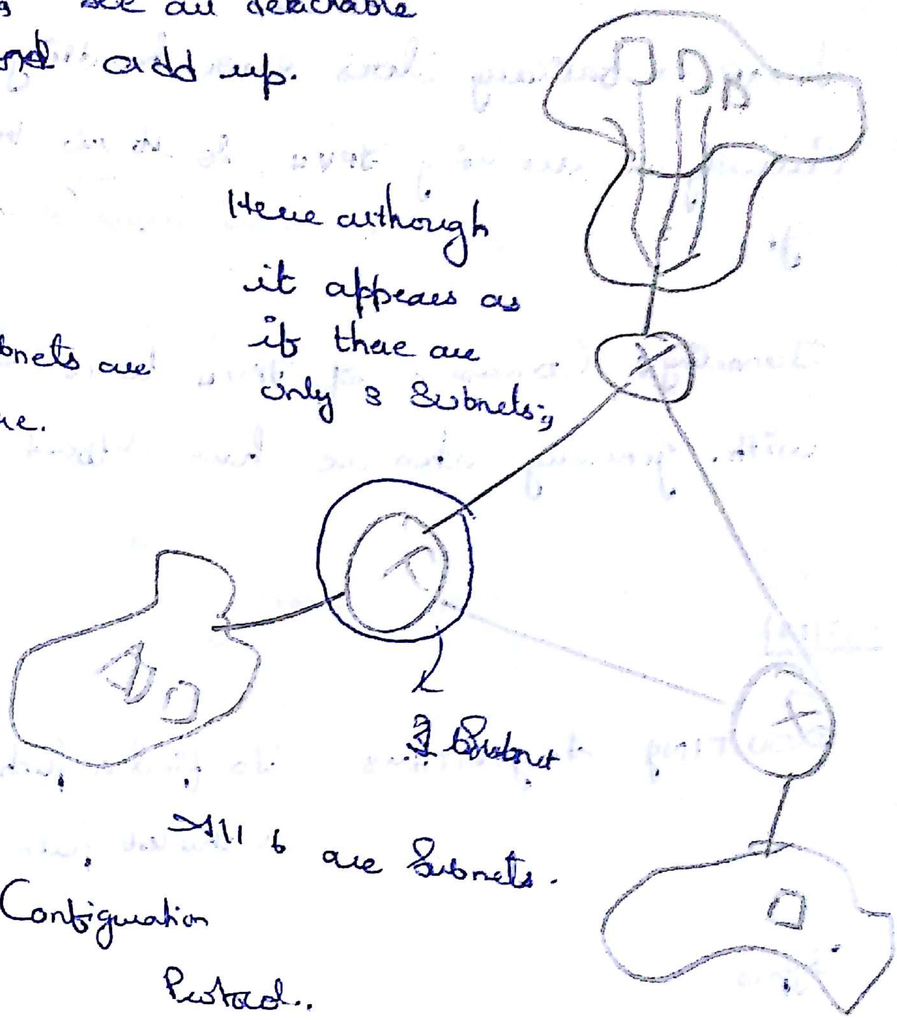
fragment flag = 0, this packet's transmission stops.

Subnet: A set of Computers accessing the Internet through a Single Switch / Single link from router.

finding number of Subnets, see all detachable parts from each router and add up.

Here although it appears as if there are only 3 subnets,

actually 6 subnets are present in here.



All 6 are subnets.

① DHCP → Dynamic Host Configuration Protocol..

నన్నుడు స్వరాలుల మధురిమ ఎప్పుడూ కను రెప్పలా చెప్పిడే 'ఎదలో'
పన్నుK పువ్వులు· కువ-కువ నాజూని నే నువ్వని!

నా చూపు పంపిడితే పన్నుట వే.చూలనే! జొన్న వి త్తులు.

DHCP additional ~~Client~~ uses:

- Give ~~addres~~ address of DNS Server.
- Give first-hop Server for the Client.

200.23.16.0/(23) → Tells that First 23 is Local/Subnet address and rest of them for identifying host-address.

IPV6 doesn't allow fragmentation of messages. Time header.

ICMP → basically does error handling.

Currently we are using IPV4, so it is better to keep IPV4 rather.

Tunneling: Conversion of IPV4 to V6 whichever you are comfortable with. Generally when we have different bodies.

31|2|17

ROUTING ALGORITHMS: To find a path from source to destination (Shortest path.)

Types:

1) Global routing Algo / Link-State Algo
2) Decentralised Routing Algo. / Distance-Vector Algo

Link-State Algo:

- Topology is known.
- Link-State Broadcast
- All nodes have same info with them.

Dijkstra is used for Shortest path.

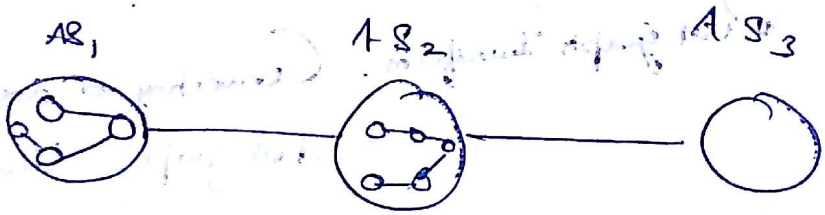
Distance-Vector Algo:

- This uses Bellman Ford Algo.

Count-to-Infinity Problem: It takes large number of iterations.

There are two kinds types of Routing Algos.

- Intra AS-Algo
- Inter AS-Algo.



AS → Autonomous System.

Gateway Router → The router which tells which is best destination in each AS.

Intra AS Routing:

1) RIP – Routing Information Protocol.
   - Distance Vector Algo.
   - The cost is hop. each hop is 1.
   - After sometime (say 30s) the nodes send its value to all nodes (read nicely).

2) OSPF (Open Shortest Path First).
   Link-State Algo.

3) IGRP.

BGP also read.

Longest Notes dec → CSP. dec-2.

Constraint Graph:

Nodes → Edges Variables Edges → Constraints.

Eg:

→ It says that variable x and y have a Constraint.

Dual Graph Transform: Converting a normal Constraint graph to a dual graph. (binary Constraints.)

- Create a variable each for every Constraint in original graph.

Eg:

original.

- $<(x,y,z), C_1>$ C_1 : $x+y=8$
- $<(x,y), C_2>$ C_2 : $x<y$

$D = \{1,2,3\}$

Dual:

$\{C_1, C_2\}$ $C_1 = \{x,y,z\}$ $C_2 = \{x,y\}$

$D_1 = \{<1,2,3>, <1,1,2>, <2,1,3>\}$

$D_2 = \{<1,2>, <2,3>, <1,3>\}$.

Now Define a Relation on C_1, C_2

$C_1(x,y) = C_2(x,y)$.

→ only <1,2,3> and <1,2> Satisfy

Local Consistency:

- If each variable is a node.
- If each binary constraint is an arc.
- Process of enforcing local consistency in each part of graph causes inconsistent values to be eliminated throughout graph.

Node Consistency: A variable is node consistent if all values in variable's domain satisfy its unary constraints.

By running node consistency, you can eliminate

Arc Consistency: A variable is said to be arc consistent if every value in domain satisfies variable's binary constraints.

X_i is arc-consistent wrt to X_j if for every value in domain D_i there is some value in D_j that satisfies the binary constraint on arc (x_i, x_j).

AC-3 (Arc Consistency) Algorithm:

- Start with all arcs in a queue (order is not a matter of concern).
- While queue has more arcs
- Pop an arc (x_i, x_j), and make x_i arc-consistent wrt to x_j.
- If D_i is unchanged go on.
- Else add to queue all (x_k, x_i) arcs neighbours of x_i. (change in domain affects neighbours.).
- If D_i is now, no consistent sol^n.
- Goto while.

Analysis: If CSP has n variables with domain size d and with c binary constraints.

Each arc (x_k, x_i) is atmost introduced d times into queue (as after deleting each time we add.

Checking Consistency for arc → $O(d^2)$ $|D_1| \times |D_2|$

No. of arcs = Constraints = c.

→ Complexity → $O(c \times d \times d^2)$

(c: no. of arc; d^2: max. arc is added; complexity for arc Consistency.)

$= O(cd^3)$

PATH CONSISTENCY:

A two variable set $\{x_i, x_j\}$ is path Consistent wrt to a third variable x_m if,

for any assignment $\{x_i = a \text{ and } x_j = b\}$,

x_i, x_m and x_j, x_m must be arc Consistent.

(just read TB once.)

K-Consistency:

CSP is K-Consistent if,

- For any set of K-1 variables and for any Consistent assignment to theses.

∴ 1 - Consistency → Node

2 - " → Arc

3 - " → Path.

Strongly K Consistent if it is K-Consistent ^ K-1 Consistent ^ ...

For an n-Consistent CSP, we are guaranteed to find a Solution in $O(n^2 d)$. = $O(n^2 \cdot d)$

Boundary for finding Consistent values of other (n-1) values given 1 is

$O(d)$ for each $\rightarrow$ $O(nd)$.

And we have 'n' such variables.

$\rightarrow$ $O(n^2 d)$.

14/11

Border Gateway Protocol.

Inter - AS

TCP Connection exists between various Gateway nodes of different A.S.

Features : (Google)

(i) Preference values

(ii) AS-path

(iii) Next hop.

Network layer ends

Ch - 5 : LINK LAYER PROTOCOL.

Link layer has informations related to when a particular host should be given access to the Network.

Link layer protocol encapsulates incoming Datagram into a frame.

Services:

(i) Framing

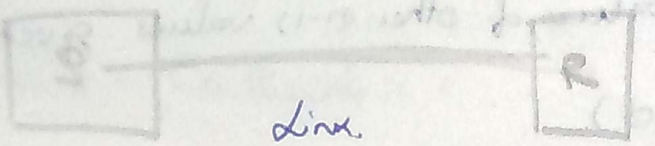
(ii) Link Access.

(iii) Reliable Delivery.

(iv) Flow Control

(v) Error Detection and Correction.

(vi) Half and full Duplex Communication :

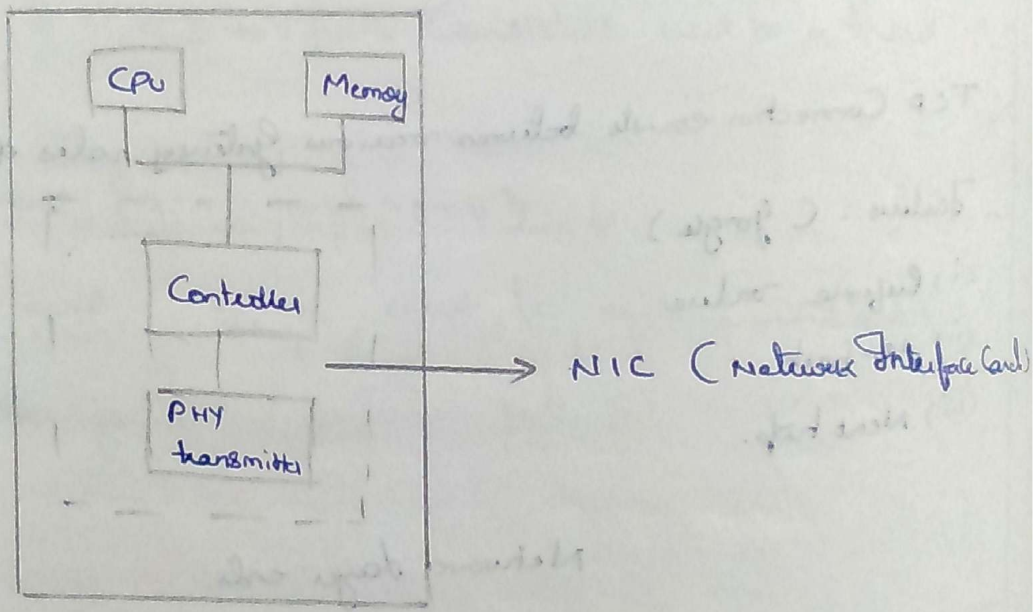


Link.

Half Duplex: We have S-R and R-S possible Communications but not at the Same time.

Full Duplex: We have S-R and R-S and also at Same time we have a possibility of both.

Router.



All Link Layer Operations are done by Controller Chip.

తరాల నా కథ క్షణాలదే కదా! గతించిపోవు గాథ నేననా!
ఆమని పాడవే హాయిగా! మూగవై పోకు ఈ వేళ.

- (వేటూరి సుందరరామమూర్తి)

కలలారని పసిపాప తలవాల్చిన వడిలో, తడినీడలు పడనీకే ఈ దేవత గుడిలో
చిరుచేపల కనుపాపలకు ఇది నా మనవి.

Sender

(i) Encapsulates Datagram to frame.

(ii) Error checking, Flow Control, ROT.

(iii) -

(iv) -

Reciever

(i) Decapsulates the frame.

(ii) Look for error Correction.

(iii) Flow Control, ROT

(iv) Send the Datagram to n/w layer.

NOTE:

Bandwidth: The range of frequencies,

5.9 GHz
→ 10 MHz is B/w.
5.9 GHz

used for data transfer.

Data Rate: The speed of data transfer that is achievable in the link.

Throughput: The effective speed that is got at the Reciever end.

MAC (Medium Access Control) Protocols:

a) Channel Partitioning.

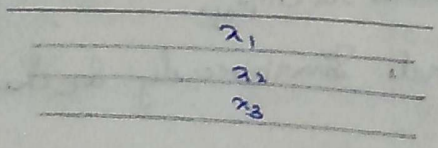
b) Random Access

c) Taking Turns.

a) Channel Partitioning.

In case of clash of two frames with same time preference.

(i) FDMA - Frequency ~~Distribution~~ Division Multiple Access.



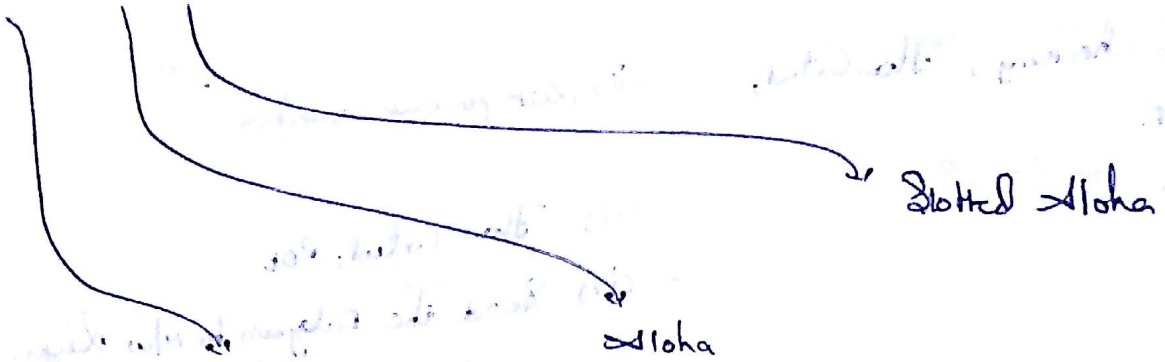
(ii) TDMA - Time Division Multiple Access.

These are different from FDM, TDM.

FDM, TDM is division of full link for the data to be available to all users.

FDMA, TDMA is when many users want to Communicate to a channel.

b) Random Access



Aloha:

Every one transmits at same time and in such a scenario, Collision is indicated to nodes and they again send after sometime with probability p.

Slotted Aloha:

The time is divided into slots of $t_1, t_2, t_3 \ldots$

The time-slot data is known before hand and various packets place their requests.

All the Collisions in a time slot are indicated by end of time slot.

CSMA:

Carrier Sense Multiple Access:

Packet sends its data only if channel is empty.

Collision occurs only if two packets simultaneously see channel as idle and launch themself.

CSMA/CA → Collision Avoidance.

Random exponential Backoff.

$[1 \ldots\ldots CW]$

Whenever we have a Collision we wait for amount of time fixated in window.

After each collision frame moves forward and wait time increases exponentially.

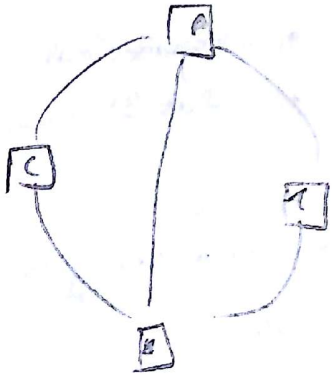
- CSMA/CA → used for wireless / WiFi

CSMA/CD:

- Whenever a collision is detected, Stop transmitting.
- CSMA/CD used for Ethernet.

7/4/17 Read Taking Turns Protocol → CSMA/CD
Token Passing Protocol. → Token Ring / FDDI.

Token Passing:



Pros:

- No Single point of failure / de-centralised.
- No Polling Overhead
- Efficient

Cons:

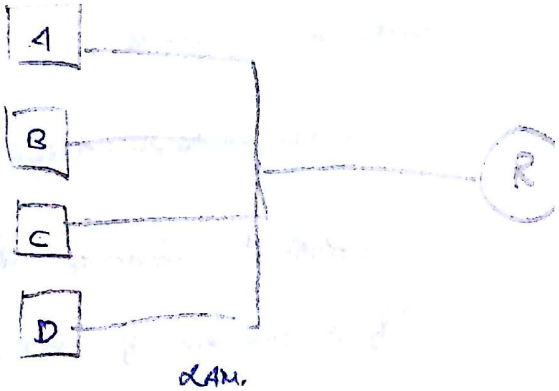
- Failure of link leads to overhead.
- Latency due to Sequence.

MAN - Metropolitan Area Network. WAN - Wide Area Network.

Link layer Address / MAC Address:

If we want to send data within same LAN, the request goes to R and then gets back to within LAN.

So, we want to avoid overlapping therefore we give a MAC-Address.



MAC-Address → Aadhar UID (Same for a given host.)
IP - " → Postal Address (may change over time).

MAC address is present for NIC (Network Interface Card.) → makes the frame to the link layer.

MAC - 6 bytes.

First 3 bytes → The manufacturing Company ID given by IEEE.

Next 3 bytes → The " " decicdes these.

Features of MAC:

- It is fixed.
- FF-AA... → hexadecimal representation.
- Fixed.

MAC	IP
• Fixed	• Dynamic
• Hexadecimal	• Decimal.
• Flat Structure	• Hierarchichal Structure.

Once link layer gets Datagram, it adds H$_E$ which has MAC address of destination.

So, basically we should be able to get MAC from IP address for making link header.

So, Network layer should have an ARP module.

ARP table is maintained for each host, which has IP, MAC and TTL.

If we get an IP in ARP, it is fine and we can get its MAC.

But, if we miss

MISSED ARP-TABLE WHILE MAC-SEARCH:

1) Case-1: Same Subnet:

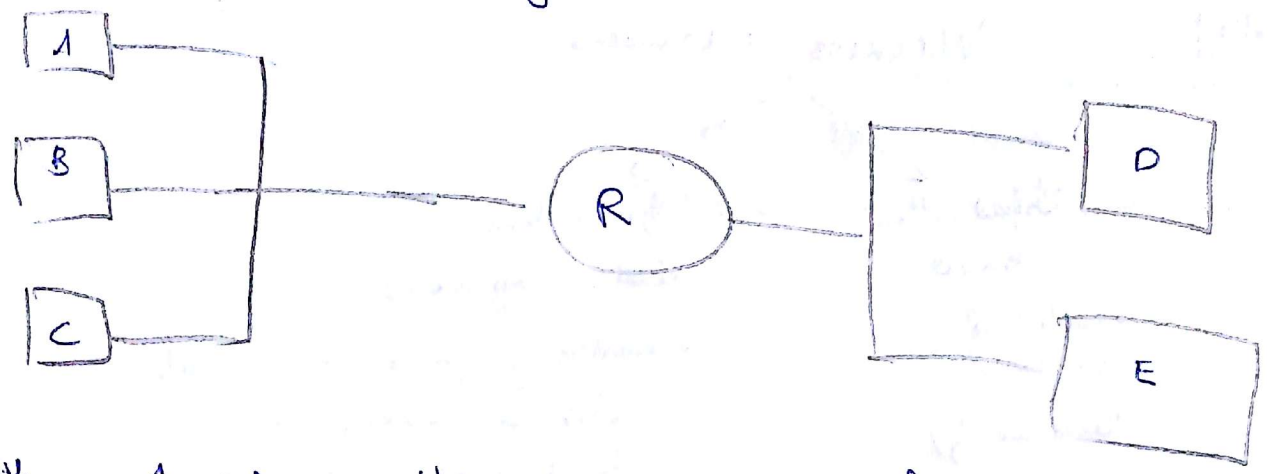
If A and B of same subnet want to communicate. (A → B)

If A's ARP table, we get a miss.

So, it sends a broadcast giving IP of B and MAC of Broadcast. This is received by all and B recognises and sends its MAC to A directly

Case 2 : Not in same Subnet

Let us say A - D



Firstly, A gets MAC address of R and makes it header and sends it. When R gets it, MAC address is same as itself and unpacks. In Datagram A sees that destination is not R.
So, it gets the MAC of the destination IP and passes to A, which puts in header and passes on.

ETHERNET — The standard structure for LAN.

Preamble	Dest. address	Source	Type	Data	CRC
8 bytes	6 bytes	6 bytes	2 bytes	(46-1500) bytes	4 bytes.

FRAME

CSMA/CD :

When A is transmitting a packet, it keeps sensing channel and if it finds signal from anywhere else, it stops itself sends a JAMMING signal and goes to a back-off state. (i.e waits for a random window of time and starts sending again.)

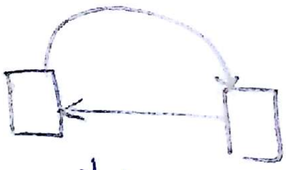
10 - BASE - T

- 10 → 10mbps
- BASE → Baseband Ethernet
- T → Twisted Copper wire.

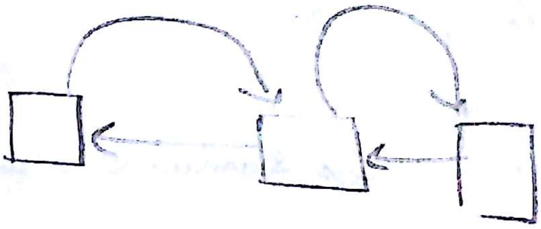
Link Layer Ends.

11/4/17 — WIRELESS NETWORKS

- Infrastructure Based
  - Centralized entity which takes care of / monitors all of traffic flow.
  - Eg: Cellular and Sensor N/w.
- Infrastructure Less. (ad-hoc).
  - Basically peers communicate with each other, directly if it is efficient to do it that way.



1 hop network.



2 - hop Network.

Classification on Hop-Basis:

1) Single-Hop – You have only one device which is intermediate from Source to Destination.

   Eg: Cellular Network.

2) Sinknode; There is a node in the network who is I/f based:

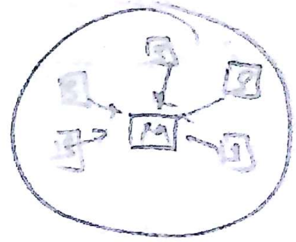
   ~~[illegible]~~

   We have to move more than on hops to reach destination.

3) Single-hop-Infra-less:

We have an adhoc network, where we have master - slave System. → It is single ~~hop~~ hop System as anywhere we get a request, we move to Master.

E.g: Bluetooth.



4) Multi-hop Infra-structure less: It is like an Ambulance appearance on road. The info is passed on peer-to-peer without any Centralised thing.

Wireless Networks properties:

- Less Signal Strength.
- Interference of Signals, Corrupt final audio. But Interference leads to loss in Signal.
- Fading → The Signal Strength Comes down
- For any device, any other Signal than its noise el.
  Why Kaushal-bon Arjun sitham

• Fading → the Signal Strength Comes down

• For any device, any other signal than its is noise.
Why transmission

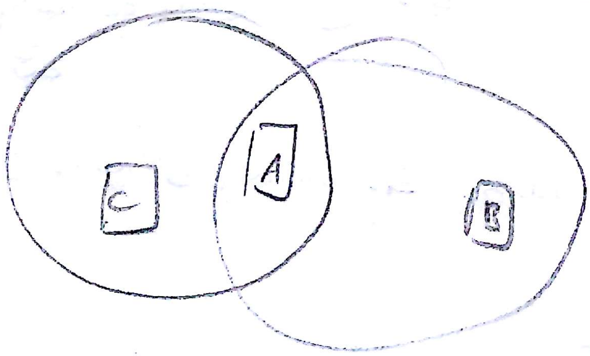
• Due to, many [illegible] refractions and reflection, with

SNR → Signal - Noise Ratio

SNR = dB.

C Signal to Noise Ratio.

the threshold to be selected.



Hidden Terminal.

Let us say the circles are made of interest.

B, C are independent of each one's space.

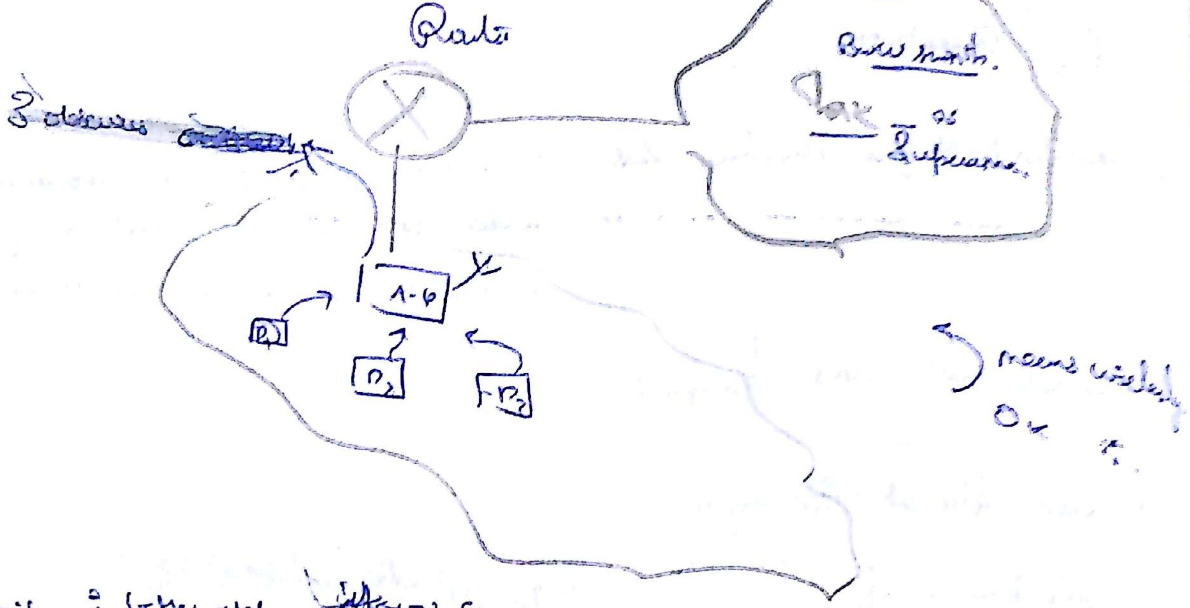
C i.e B, C have no idea of each other.

Scanned by CamScanner

Exposed Terminal Problem → Read it.

Wi-Fi → Wireless Fidelity.

- It uses CDMA/CA
- The simulated outputs a Collision only.



Init is better use [illegible] you work.

DIFS → Distributed Interframe space.

SIFS → Short [illegible] spacing [illegible]

/* [illegible]

When channel is idle, only then x → x-1 transmit [illegible] ?

/* [illegible]

[CTS] - Clear to Send Packet.

RTS → Source ask

RTR → Request Receiver to Send.

Read how CTS, RTS makes 802.11 Collision free wireless Networks.

[4/11] FORMAL METHODS: (NOTES FORGOT):

Grammar:

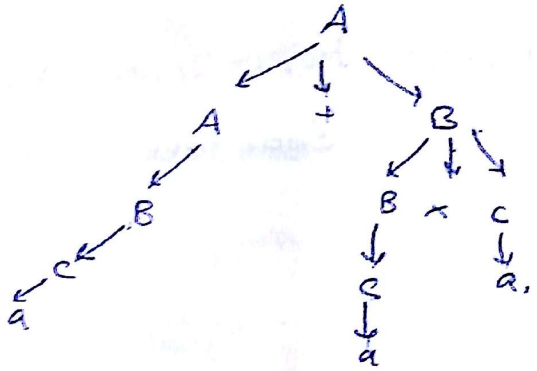
$A \rightarrow A+B \mid B$
$B \rightarrow B \times C \mid C$
$C \rightarrow a.$

This ensures × before +.

Getting i) $a+a\times a$

$A \rightarrow A+B \rightarrow B+B \rightarrow C+B \rightarrow a+B \rightarrow a+B\times C \rightarrow a+C\times C$

$\rightarrow a+a\times C \rightarrow a+a\times a.$ [Left Derivation].



(ii) $a+a+a.$

$A \rightarrow A+B \rightarrow A+B+B \rightarrow a+B B \rightarrow a+CB \rightarrow a+a\times B$

$a+a\times a$ ...

Prove that this grammar is unambiguous. i.e any string has a unique left-Derivation.

~~Let us say you a, b, c ... you~~

Let us say we have α +'s and β x's.

Firstly get all α +'s by A → A+B and again A → A+B.

Finally substitute from left onwards and wherever you need R.. → Derivations.

CHOMSKY NORMAL FORM:

For every CFL L, there is a grammar of following form

A → BC   Such that A, B, C are variables.

A → a   B, C cannot be start variables.

S → ε   S → has to be start variable.

For conversion of Normal Rules to non-ambigous grammar. Do it immediately (Read from TB).

END - SEM SYLLABUS ... Jr somis here

ENDS HERE

Extra Topics:

Compiler → • Syntax checking (one part of it)

→ • [redacted] The intermediate phase using optimising.

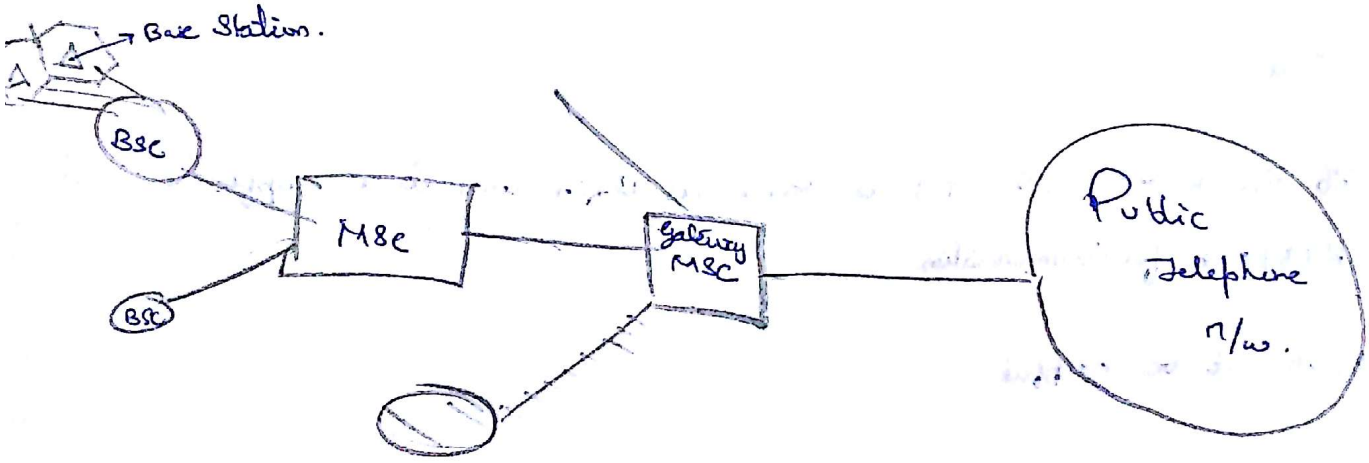
Types of Problems:

- Decision Problems
- Optimisation problems
- Function problems and its [redacted]
- Search Problems.

11/4/17

CN TUTE BY PROF:

GSM (2g):



BSC → Base Station Control.

MSC → Mobile Switching Centre.

The Range of Cellular Tower/Base Station is 10km.

MSC has two databases HLR and VLR

HLR → Home location Register

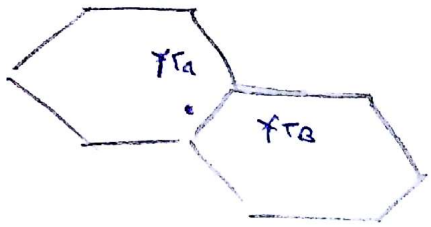
VLR → Visitor location Register.

Flow of a Call:

- You will request your nearest Tower and it puts request to BSC forwarded to MSC.

MSC checks its HLR and if it is [illegible]

If you don't get in HLR you forward to another MSC and check its VLR.



If a terminal moves from one Base Station to another.

In such a scenario, T_A notices that it is losing a node and T_B also gets notice of incoming node.

This is called HANDOFF.

Hard Handoff: T_A loses info about outgoing nodes.

Soft " : Both T_A and T_B maintain data of node.

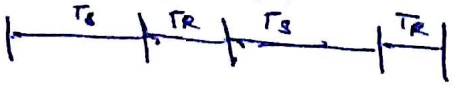
18/4/17

Questions:

1) If Size of RTS and CTS is same as DATA will it be helpful to send RTS/CTS for communication

No, it will be helpful.

2) one way Propagation = P, bandwidth = B bits/sec.



Link → TDM.

- If you are implementing Stop and wait.

a) What is Sender timeout for retransmission.

b) Assuming No packet loss, what is the link efficiency of this system

4) a) The time out time is $T_S + T_R$. The time is divided among both nodes as T_S, T_R, so, within $T_S + T_R$ if you don't get ACK, you send again.

b) The total time for a transmission is $T_S + T_R$.

F → Number of bits sent. B → Transmission Rate.

F/B → Time for transmitting in the line.

$T_S + T_R$ → Total time taken for this.

$$\eta = \frac{F/B}{T_S + T_R}.$$