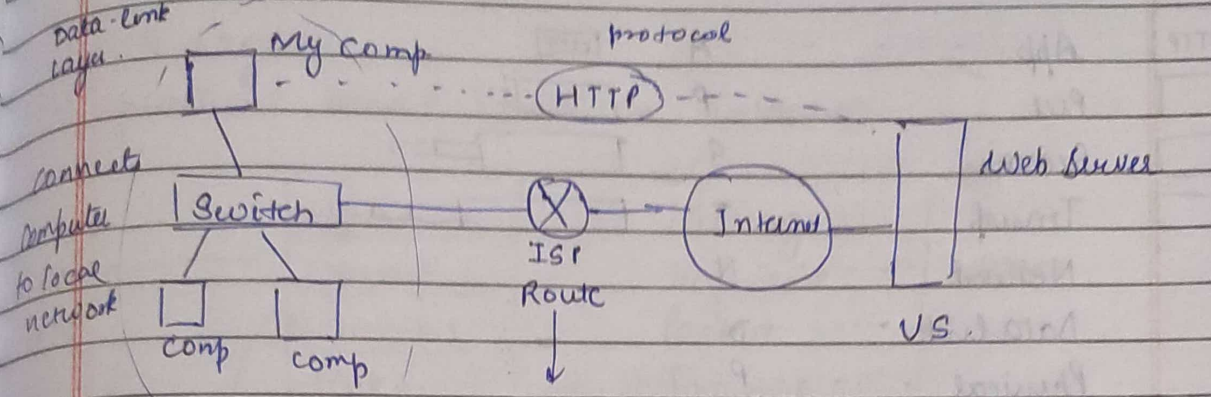
Implemented LRU Cache



Protocol → set of rules for communication
(decided by headers)
↓
may be destination, source etc.

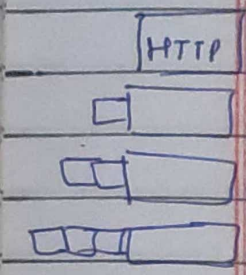
Q Why do we need Layers?

⇒ To organise diff. protocols.

Q Imp. Models are → OSI & TCP/IP

Sender → attaches header
Reciever → processes the header & dattach it.

① OSI



App. - - - - A

Pres. P

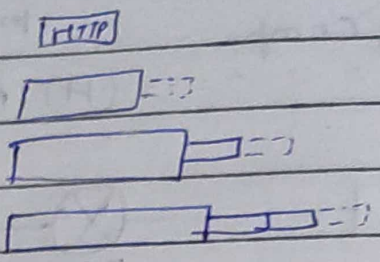
Session S

Transp T

Network - - - - N

Data link - - - - D

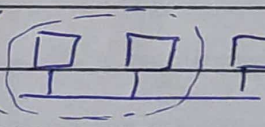
Physical - - - - - P



App. (R) talks to App. (Sender) only.
No Need to worry about Lower Layers.

(Encapsulation)

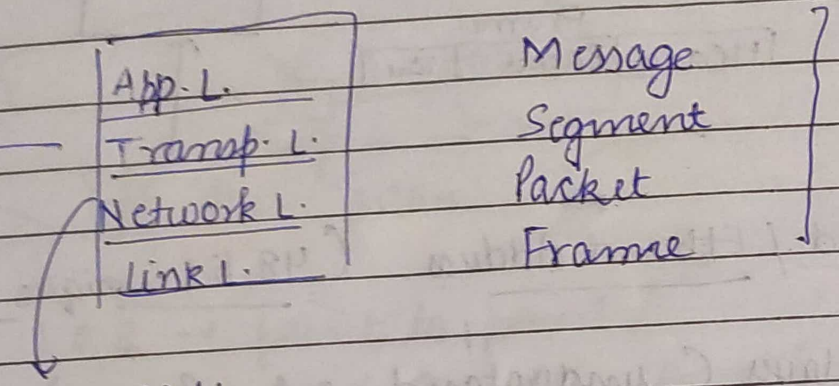
→ It is theoritical Model.

→ DL Layer ( , wifi)

Date: __/__/____ Page no: ____

⇒ TCP/IP Model

- 4 Layers
- practical Model used in Internet

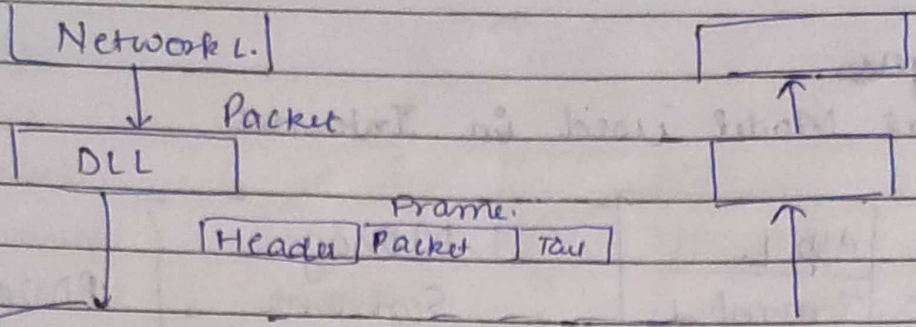


HOL ← Transp. L. ie TCP

Network L. → unreliable connectionless IP

Date: / / Page no:

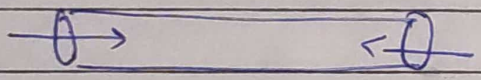
* Data link layer.



Mac Add / Ethernet Addess. (48 bit unique add.)

Physical layer (understand only Binary)

DLL Eg. (Police Man)

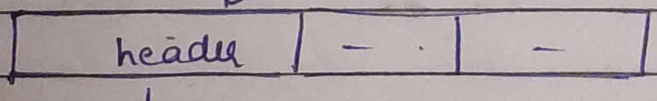


- people should reach from 1 end to another without any issue.
- Source / Dest. does not matter.

* DLL access in ~~fixed~~ Network Intrface Card

↓

It contains a fixed 48 bit Mac add (uniq. worldwide)

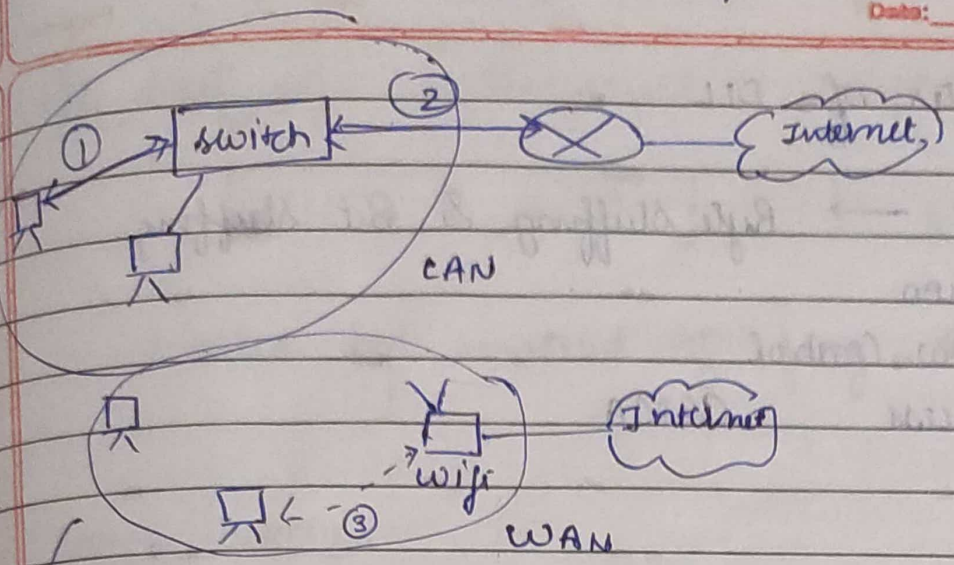


↓

Source / dist mac address.

Router → 2 or more interfaces
→ create multiple Sub nets
(allow multiple Networks to connect)

Date: / / Page no:



1, 2, 3 → point to point /
end to end connection is done by
Data link layer

Switch →

① DLL device
② multiple ports to connect diff. devices
③ 1 port connect to router which further connects Internet
④ switch allows i) devices to connect with each other (DLL) without collision.
ii) connect device with router (Internet)

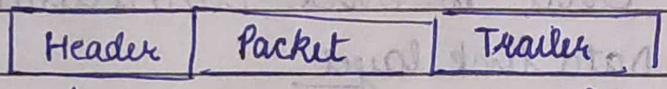
2 Collision is possible.
(CSMA/CA protocol handles it).

Date: / / Page no:

* fundionalites of DLL

① Framing → Byte Stuffing & Bit Stuffing.
② Error Detection
③ Error & Flow Control
④ Multiple Access CSMA

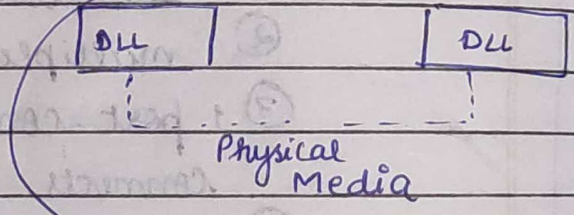
→ DLL receives a packets & add header & trailer (frame)



Frame

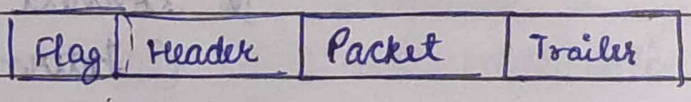
contains source mac address
dest mac address
(use ARP protocol to get this)

contains checksum
(To detect error while sending frame via phy media)



→ 2 Methods for framing.

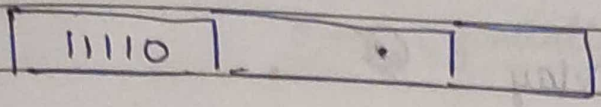
① Byte Stuffing



- When reciever see flag it knows new frame is present
- It can also be present in Packet as
  i) ESC Flag
  ii) ESC ESC Flag (if ESC already exists)

Date: / / Page no:

② Bit Stuffing: (difficult to implement)



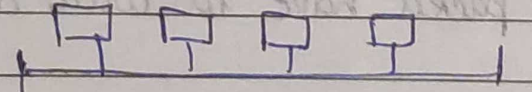
Similar but instead of flag Bits are used

* Error Detection
① Parity
② Checksum
③ CRC

* Error & Flow Control
Retransmission → Re
Alg are
① Stop & wait
② Go Back
③ Selective Repeat
} ARQ (automatic Repeat Request)
(Sliding window)

* Multiple Access
Wifi → CSMA/CA
Collision sense multiple Access / Collision Avoidance

multiple device sharing same medium



classical Ethernet

(In Present we use switches)

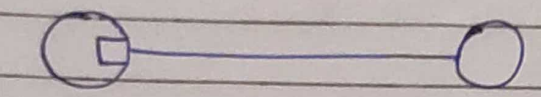
eff. alg.
① ALOHA
② CSMA/CD

Delay

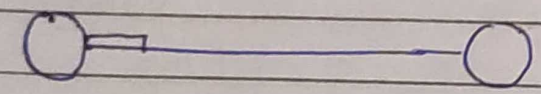
① Transmission Delay

↓

time taken by sender to put data on the channel.

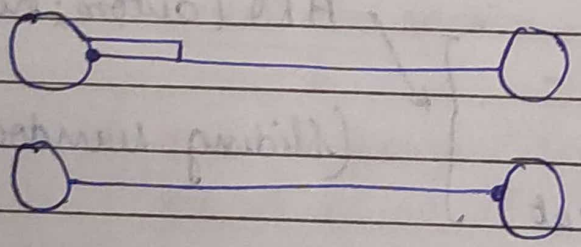


Depends on
① line Bandwidth (B)
② Packet length (L)



$$\text{delay} = \frac{L}{B}$$

② Propogation Delay



Depends on
① Distance
② velocity

$$\text{delay} = \frac{d}{v}$$

time taken by last bit of packet to reach receiver.

③ Queing Delay

(May be packet have to wait at router)

④ Processing Delay.

(router process the packet

↓

① encryption
② decryp.
③ processing headers

Date: __/__/____ Page no: ____

Delay = $T_t + T_p + T_q + T_f$

$T_q + T_f$ → 0 + 0 (generally)

* ~~Stop & Wait~~ Error & Flow Control

It occurs in 2 layers:

① Transport

② DLL

Transport layer works on IP & are unreliable - connection less

∴ It is imp. to check for Errors

DLL works on phy. layer & due to signal issues ~~bits~~ bits may change specially while using wifi.

∴ It is imp.

Transmission Time = $\frac{\text{Frame Size}}{\text{Channel capacity / Bandwidth}}$

Date: / / Page no:

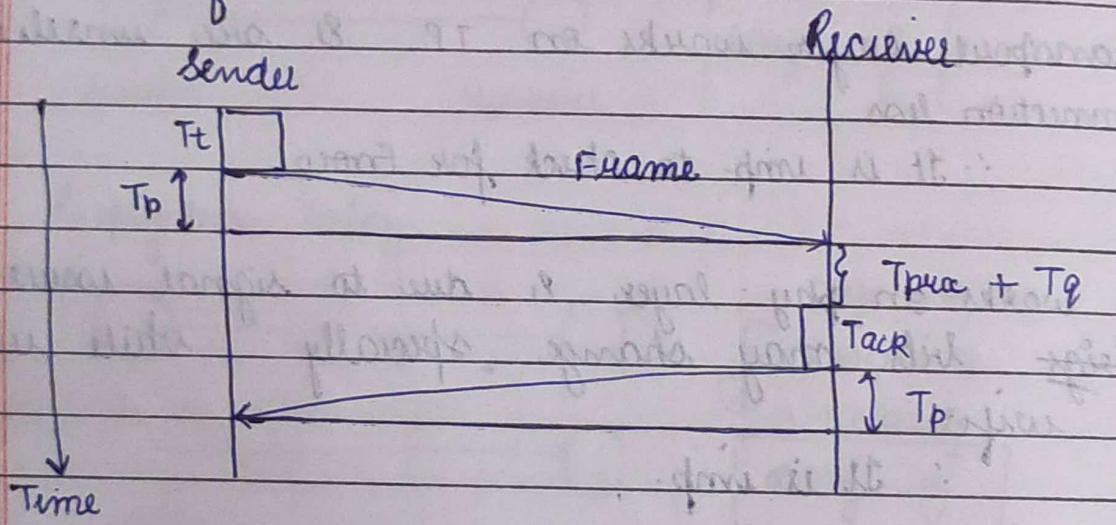
Stop & Wait

Sender ⟶ Reciever

(It do error detections)

→ Sender sends a ~~is~~ frame & waits for acknowl.

→ Once ack. is recieved then only sends the next frame.



$$T_t + T_p + T_{proc} + T_q + T_{ack} + T_p$$

X X X

ignore.

$\therefore$ $2T_p + T_q$ ← Total time

Useful Time ⇒ T_t

⇒ $\text{Eff.} = \dfrac{T_t}{2T_p + T_t}$

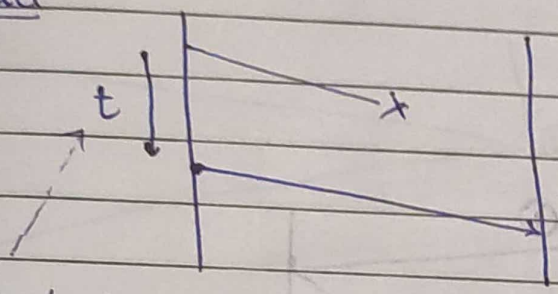
⇒ $\text{throughput} = \dfrac{L}{T_t + 2T_p}$

Date: / / Page no:

It must handle.

1. Flow Control (1 frame at a time (wait till ack.))
2. Lost Data
3. Lost Ack
4. Delayed Data
5. Delayed Ack.

Lost Data

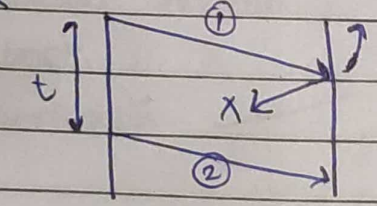


Timeout :

Sender waits for fixed amt. of time for ack., if not received then again send the frame.

7:30

Lost ACK



Before getting ② rec: already send signal to above layer
∴ Duplication may occur.

∴ Seq. No. are used.

if same frame (of same sq: No.) is received. No signal is sent to above layer.

Because it is already done.

Seq. No must be small.

Stop & wait → (0, 1) Only.

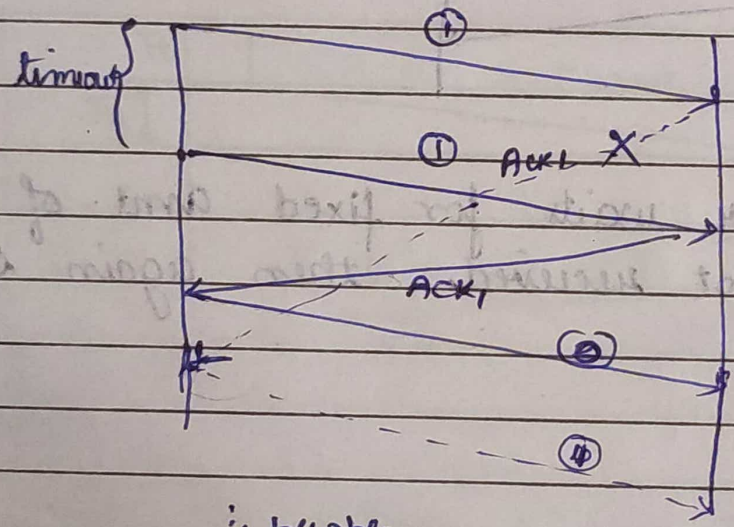
∴ ↓

also k/a Alternative Bit Protocol.

Delayed Data

automatic

Delayed Ack



∴ problem.

Soln:

Seq. No. to Ack:

Frame Seq. No / ACK Seq. No.

No. of frames expected by the recievers.

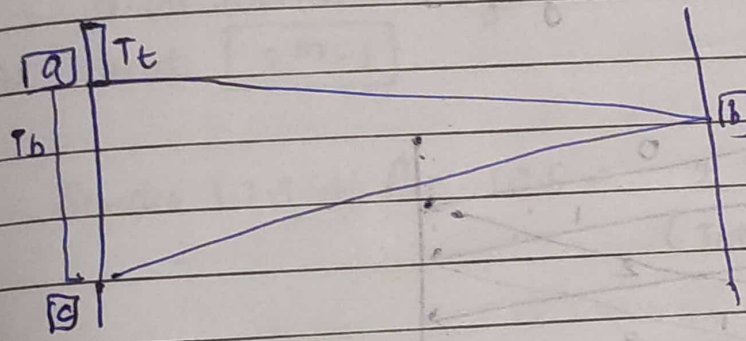
Date: __/__/____ Page no: ____

Sliding Window Protocol

To utilize communication channel more efficiently.

Prob. with Stop & wait (not work good. with high Bandwidth-Delay Product)

→ Not utilizing com. channel effectively



$Eff = \frac{1}{1+2a}$ $\quad a = \frac{T_b}{T_t}$

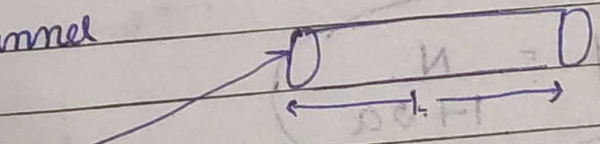
Bandwidth Delay product

RTT = time (a→b) + time (b→c)

⇒ Bandwidth * Round Trip Time

= $B * 2T_b$

Indicates channel
① length
② Bandwidth



~~If 2 Ways~~ → $B * 2$

Date: / / Page no:

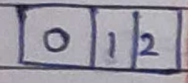
* window is a buffer used to keep copy of frames to handle error & do flow control.

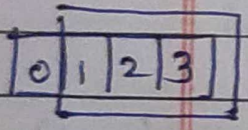
if something is lost we can check in window.

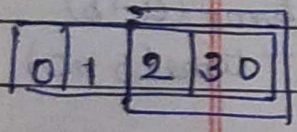
Go Back N

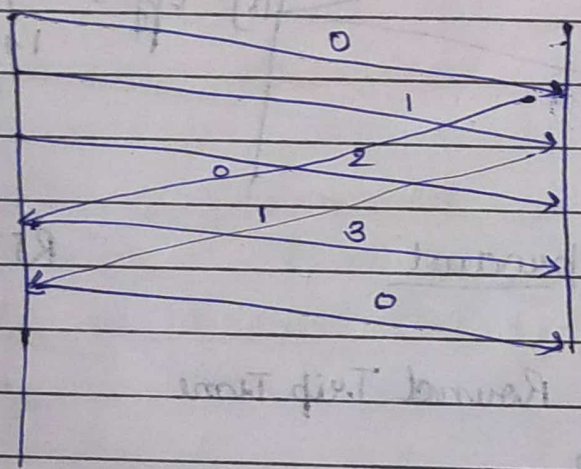
Sender window Size → N
Receiver Window Size → 1

S-window





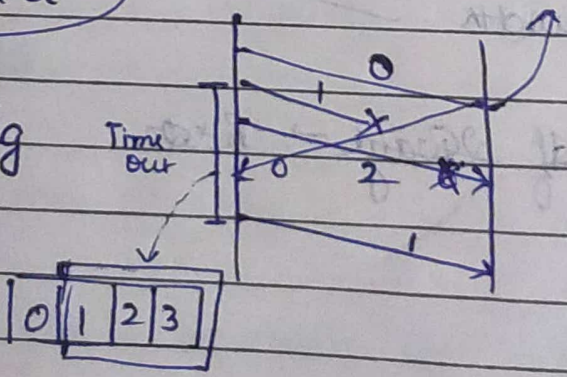




0 1 2 3 0 1 2 3 0 1 2 3 ---

$$Eff = \frac{N}{1+2a}$$

* Error Handling



If 1 is discarded then receiver also discards 2 (Not in sequence)

after time out 1 is send again.

* Send 2 is also Need to be send.

Date: / / Page no:

② Ack Lost → (?)

If Seq. No is of m bits
then window size is
at most $\boxed{2^m - 1}$.

* Sender W.S + Rec. W.S = 2^m
(Total no of seq. No)

Go Back N ⇒ $n+1 = 2^m$
$n = 2^m - 1$

Stop & wait = $2 = 2^m$
$\therefore m = 1$

Selective Rep = $2n = 2^m$
$n = 2^{m-1}$

* Problem (Go Back N)

If frame is lost then after time out, that frames as well as frames sended after it is required to send again.

Max. window size with n bit frame sequence No. is

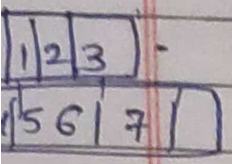
$2^{(n-1)}$

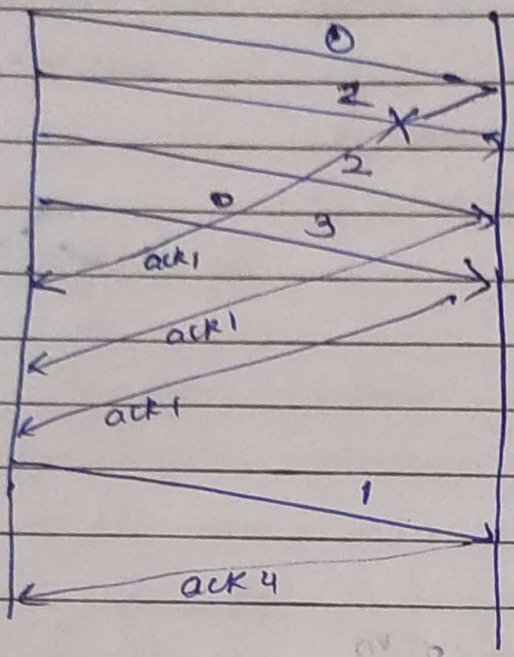
4+4 = 8

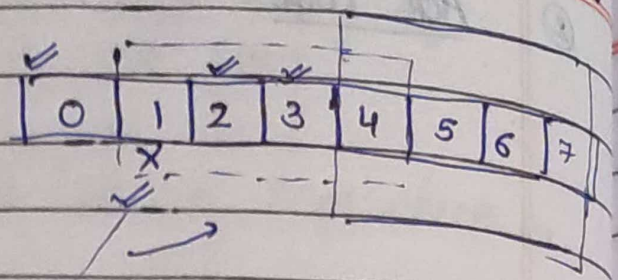
8-1 ⇒ 7

∴ 7 is min. seq. No. bits

Selective Repeat







No need to send 2, 3 again if 1 is lost

$W + R = 2^n$

$W + W = 2^n$

$2W = 2^n$

$W = 2^{n-1}$

→ Used in noisy channels

→ causes more work to receiver

→ Used with go back in Transport layer.

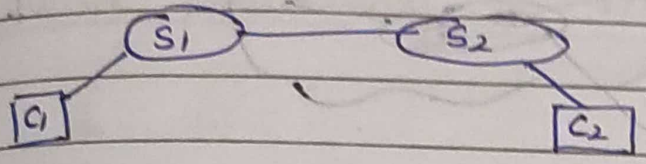
*

Eff.	$\frac{1}{1+2a}$	$\frac{N}{1+2a}$	$\frac{N}{2a}$
Window Sz	1+1	N+1	N+N
Seq NO	2	N+1	2N
Retransm.	1	N	N
Implement	Easy	Moderate	Complex

Date: / / Page no:

Network layer. (IP Layer - TCP/IP)

1 Protocol only → IP Protocol.



Switch (DLL device)

① If C1 wants to send something to C2.
Then comm. b/w switch is req.
which uses min. spanning Tree algo & ∴
Not Bandwidth efficient
↓
most imp. in networking

∴ In Network layer, there are Routing Alg.
to efficiently use Bandwidth

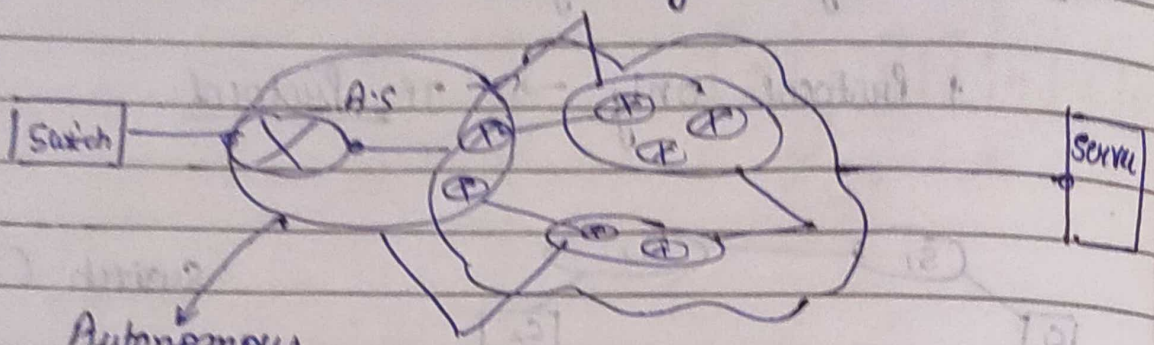
② Broadcast Domain in switches is only 1 Domain.
Eg. Devices connected to switches if any one broadcast then it is to the whole world.

Switches forms a Single Network!

If you want Multiple Networks you Need Router
(Net. Layer Device)

* Routers → DO internetworking.
→ Multiple interfaces connected to diff. Networks.
→ provided by ISP.
→ carry inf. over Internet

Internet is a hierarchy.



⇒ It contains multiple Routers
⇒ Internet consists of multiple A.S.
⇒ all AS are interconnected.
⇒ ISPs or large org.

* Services of Network layer.

① Internetworking (802.11, 3G, Ethernet) ← diff. networks.
② Addressing.
③ Routing & Forwarding
④ Scalability (using Hierarchy in Networks)
⑤ Bandwidth Control
⑥ Fragmentation & Re-assembly

* Routing :

Routers have to decide where to send packet & they maintain Routing Table.
(Look at IP add. of packet & then decide)
↳ Building Routing Table
↳ Decide where to send } ∴ is K/a Routing

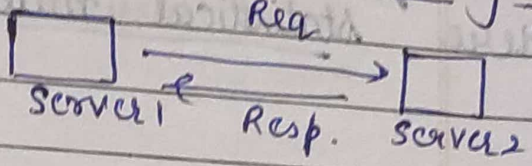
It is global decision (depends on congestion in other routers)

Date: __/__/____ Page no: ____

* Forwarding
→ It is a local decision

⇒ Circuit Switching V/S Packet Switching

Not used in CN
→ used in Telephone or Cellular Network.

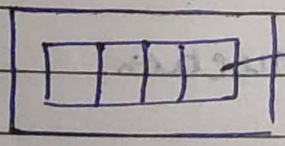


⇒ This is divided into packets which are routed to network
(Req - Resp)

Basically
① Establish Connection (allocate Bandw.)
② use it (use some B.) while talking
③ close it. (close it)

2 Strategis of Routing
① Every Packet have a distination address.
② Ind. Packet indep. of other packet (choose diff. paths)

Reason for diff. path.

⇒ 

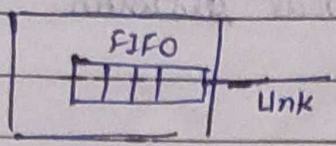
Router

If it is full (due to conjestion) Router starts dropping Packets & so those Packets choose diff. ~~path~~ paths (Routers)

Date / /

*

Store & Forward Packet Switching



It provide better utilization of link

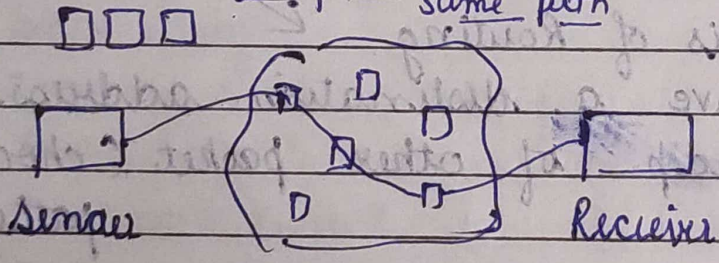
Here Statistical multiplexing

(Telephone
↳ fixed multiplex)
① Time
② freq

⭐ 2 ways of Packet Switching
① Virtual Circuit
② Datagram (used in Internet)

⇒ Virtual Circuit (get idea from Circuit Switching)

1) all follow same path



2) Connection setup needed

3) Routing Happen per Connection

4) Packets have circuit identification

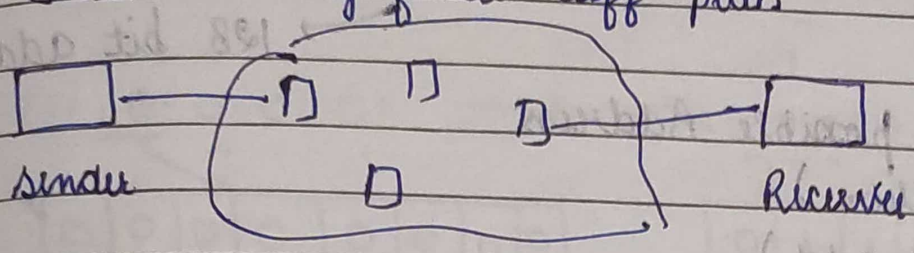
5) Failures are diff. to handle.

6) Quality of service is easy to add.

Date: __/__/____ Page no: ____

Datagrams.

1) individual packets may follow diff path



2) No Connection Req.

3) ——— per packet

4) Packets have destination add.

5) Failures easy to handle.

6) Quality of Service is diff. to guarantee

Date: _/_/_ Page no:

IP Addressing / Classful Addressing

IPV4 → IPV6 (new) → 128 bit address.

↓

2^{32} possible Address

(4.3 billion)

(32 bit add)

(Half of all IP addresses)

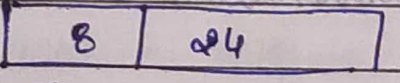
⇒ Class A → Begin with 0 (Half of 2^{32})

B →

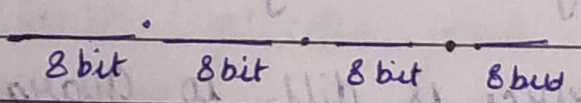
C →

D

E



Normal IP address



⇒ 2^{24} Host IP address in every Network.

⇒ 2 ~~Network~~ Host add. are Not used.

i) Network ID (all host bit 0)

ii) used for (all host bits 1)
Broadcasting

∴ $2^{24} - 2$ IP addresses can be assigned to host.

(60 million in class A)

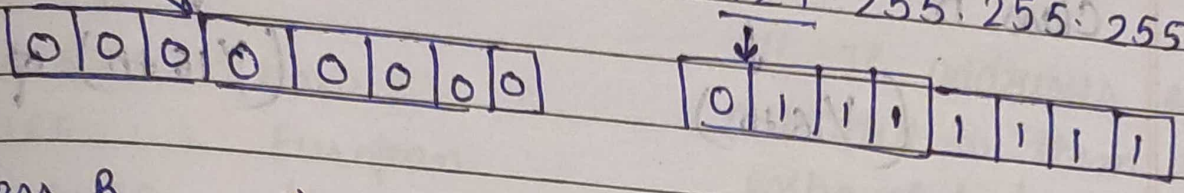
1 bit is fixed → ~~two~~ bit 1st bit as 0 → Class A in class A.

else → other class

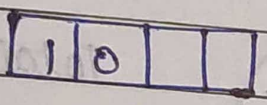
∴ Total Networks in A = 2^7

$(2^{24} - 2)$

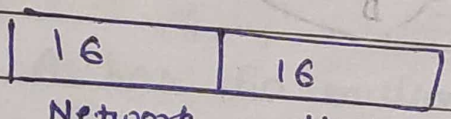
0.0.0.0 → 127.255.255.255



⇒ Class B ⇒ ~~Half Network~~ & ~~Half Host~~



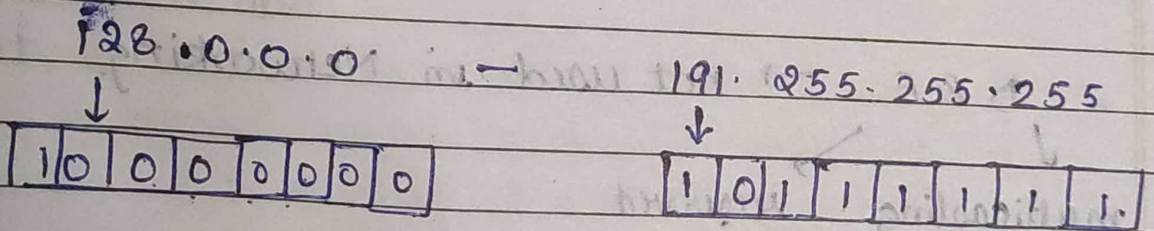
→ Begins with 10 (1/4 of all IP addresses)



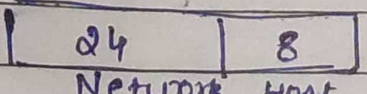
* $2^{16} - 2$ host add. can be assigned to host

* Total Networks = 2^{14} (2 bits are fixed)
16-2

128.0.0.0 → 191.255.255.255



⇒ Class C Begin with 110 (1/8th total IP add)



$2^{(24-3)}$ Network ID

192.0.0.0 → ~~193~~ 223.255.255.255

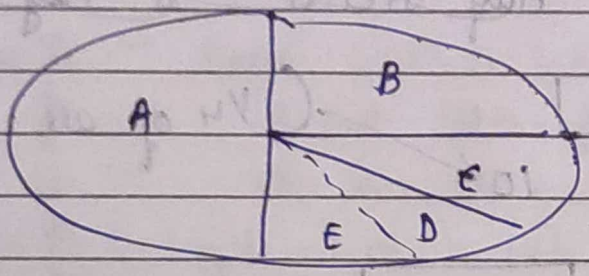
Date: / / Page no:

class C → smaller org.
A → larger org.

Begin with.

Class D → 1110 (1/16th)

class E → 1111 (1/16th) of IP



IPV4

Total No. of IP addresses.

(D) 224.0.0.0 to 239.255.255.255

(E) 240.0.0.0 to 255.255.255.255

D & E → not used in Networking.

↓ multicasting → required for future

No host bit & Net bit.

Date: __/__/____ Page no: ____

IANA (Int. Assigned Number Authority)

Assign No. & IP address to org / company.

ARIN > American
LACNIC
APNIC Asia
RIPE European
AFRINIC African

All IP addresses allocated to them are exhausted !

* Problem with Classful

Class A has 60 million host while class B have less. if org. want something in b/w they have to choose A & will end up wasting IPs.

Solution
① Classless IP add.
② NAT.

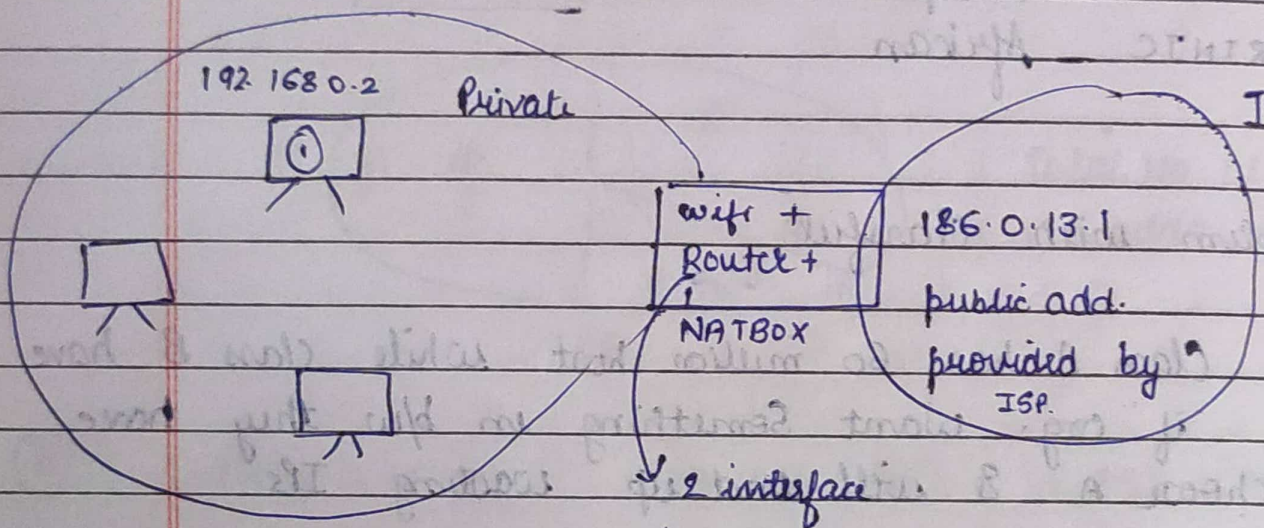
Date: / / Page no.

* NAT (Network Address Translation)

Based on private IP addresses.

⇓

private
- class A 10.0.0.0 to 10.255.255.255
- B 172.16.0.0 to 172.31.255.255
- C 192.168.0.0 to 192.168.255.255



if ① ~~wants~~ sends req. to facebook (on Internet) it goes through router where private IP is translated to public IP & facebook get request from 186.0.13.1

again for response public IP is converted to privated with help of mapping done in NATBOX (TCP/UDP port numbers).

used for

private IP → local network ipconfig
public IP → for Internet: (google whats my public IP address)

NAT Box converts private IP to public IP add

NAT

allow diff devices within a network to connect to single public IP.

11:30

* Subnetting –

dividing networks into smaller sub-networks. is subnetting

Eg Dividing IPs for diff. dep. in college.

Subnet Mask.

→ Seq. of 1s followed by 0s
→ 32 bit
→ 255.0.0.0 or 255.255.0.0 or – or –

1s are network address
∴ 8 bit Network add.
0s are host ID.
255.255.0.0 16 bit Network add.
16 bit host ID bits

supernetting → borrow from Network ID.
applicable for 2 or more Network.
improve flexibility of IP add. allotment
Security ✗

Date: / / Page no:

IP 192 . 168 . 13 . 3
Subnet Mask 255 . 255 . 255 . 0

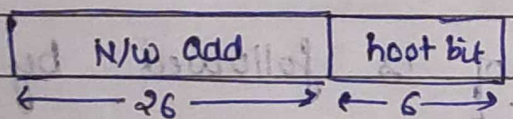
✗ Network 192 . 168 . 13 . 0
add of IP
(bitwise and)

Classful addressing is also subneting.

Subnetting (Single Network, imp. Security, borrow host ID bits.

IPs 193 . 8 . 10 . 0 to 193 . 8 . 10 . 255
No. of subnets = 4. = 2^2

Class C ∴ host bits = 8
∴ 2 bits shifted

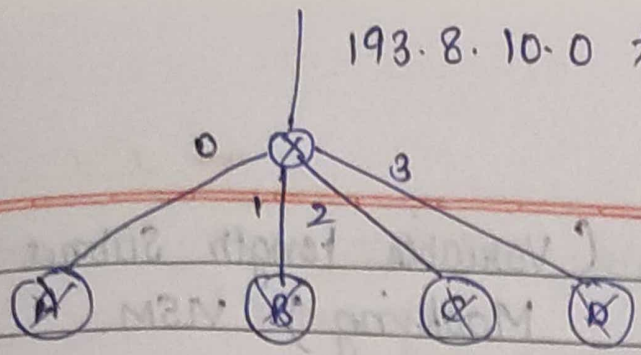


subnet 1 = 193 . 8 . 10 . 0 | 00 | 111111
to 193 . 8 . 10 . 63
(Total 62 addresses)

subnet 2 = 193 . 8 . 10 . 64 | 01 | 111111
to 193 . 8 . 10 . 127

3 = . 128 to . 191 | 10 | 111111

4 = . 192 to . 255 | 11 | 111111



Subnet mask of all = 255.255.255.192

11 | 000000

Problems

Initially 256 - 2 IP address
But now 2 address are lost in every Subnet
256 - 2 (No of Sub-net)
256 - 8 = 248

Routing Table → Subnet Mask, Interface (0, 1, 2, 3)

Eg Packet is sent to 193.8.10.130 (add)
∴ Routing Table do Mapping.

∞ It finds Network add by bitwise and

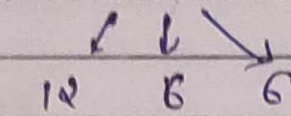
193.8.10.130
& 255.255.255.192
193.8.10.191 → (C) or (2)

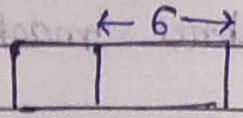
Bi send it to correct Network add.

Date: / / Page no:

unequal Subnets (Variable length Subnet Masking) VLSM

divide into 3





0 (A) + (B)
10 (C)
11 (D)

* CIDR (classless Inter Domain Routing)

Improvement of classful addressing with help of ~~subnet~~ subnetting.

Problems with classful (Boundaries in bytes)

Boundaries for customers who fall in b/w.
- A 2^{24} I/P's
- B 2^{16}
- C 2^{8}

CIDR (Boundaries in bits)

~~if~~ any random No. in Network Bound

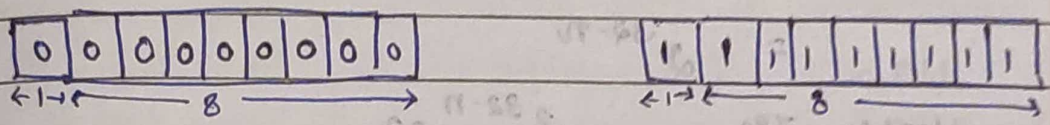
Eg. 23 ~~bits~~ bits → N/w → 512 IP add.
9 bits → Host →
↓
Anything b/w 256 & 512 then 9 bits.

Date: __/__/____ Page no: ____

1000 → 1024 IP add allocated
(req) (Power of 2)

* start (all host bit should be 0)
end (all host bit must be 1)

Eg. 193.10.10.0 to 193.10.11.255



193.10.11.0
X invalid

Subnet Mask

⇒ 255.255.254.0
(9 bits 0)

Alternative Way (CIDR Notation)

IP Subnet Mask

193.10.10.1/23 → No of Set values

any 1 b/w the range

IP / Subnet Mask

Date: / / Page No:

✗ a.b.c.d /n

⇒ Subnet mask $\underbrace{11\ldots1}_{n}\underbrace{00\ldots0}_{32-n}$

⇒ ~~host~~ Total avail. IPs in this network with a subnet mask of n bits

$= 2^{32-n}$

⇒ Usable IPs $2^{32-n} - 2$

⇒ All IP addresses in the Network will have first n bits same.

⇒ 1st IP add is last (32-n) bit must be 0

⇒ last 1

More on Subnetting.

① gateway → It is a device that connected to internet & makes your device connected to internet. (IP add. of g. router)

② DNS Server → It gives IP address from URL name.

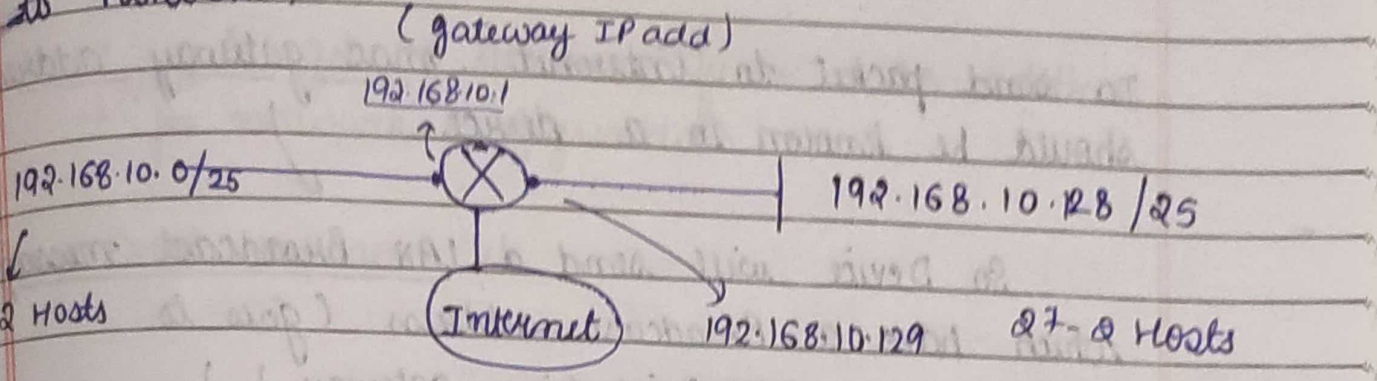
③ IP add.

④ Subnet mask

(Nowdays DHCP are used)

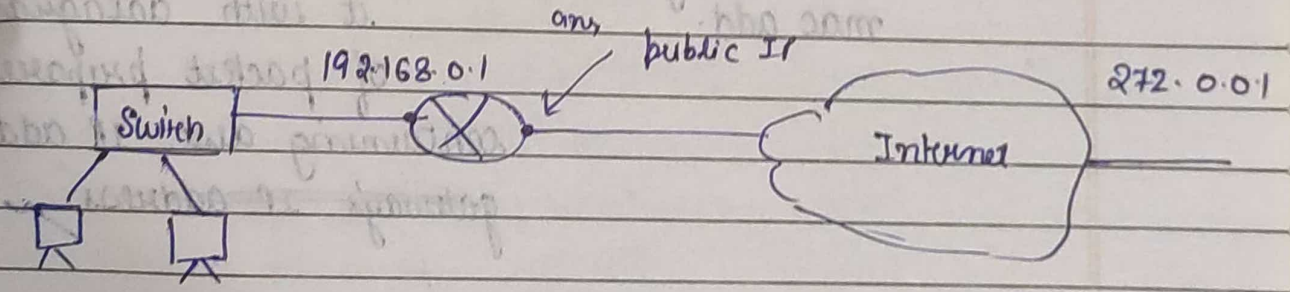
Date: / / Page no:

All 4 things are required to be set to connect to network.



* ARP (address Resolution protocol) ← DLL protocol

Used to get MAC address of a device in LAN with given IP address.



DHCP does all configuration. All are default!

* To Send packet to Internet Device should have a LAN connection with gateway (Router).

Typically all devices have ARP cache where IP add to Mac add. mapping is stored.

Eg: ARP command

(caches)

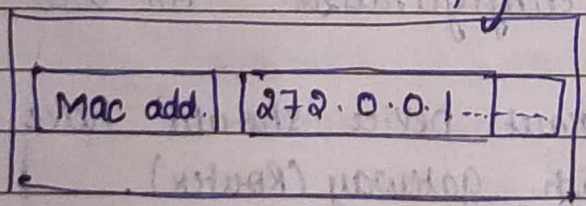
To send packet to internet, mac gateway addrss should be known to a device.

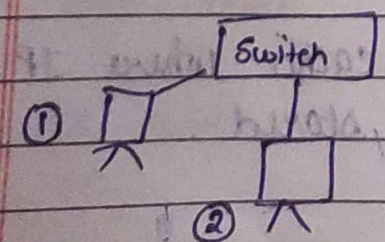
So Device will send a LAN Broadcast message which have a Broadcast address (goes to everybody of LAN (including gateway))

Now Router responds to ARP req with its ARP response containing ~~its IP address~~ mapping of its IP address & its mac address

Now Device prepares a D.L.L. frame with destination, as gateways mac. add & attach it with datagram of packet prepared containing dest. IP add as ~~gateways IP address~~.

(refer diag) prev. page





⇒ If ① wants to connect to ② No need of route

① can broadcast its IP add of ② & ② rnd in resp. to ARP req. sends mapping of its mac & IP add to ①. Now ① stores it in cache for future use. Both can now send packets.

Date: __/__/____ Page no: ____

RARP (Reverse ARP)

→ used to get IP address from mac address.
→ for network configuration.

DHCP replaced it.

Date: / / Page no

Transport layer

(Most Complex Layer)

Implemented in OS

Services

1. End to End Commun / Process to Process Comm.
2. Connection Oriented Communication. (TCP only)
3. Reliability (only TCP)
4. Flow Control (TCP only) — (Sliding window protocol)
5. Congestion Avoidance & control (TCP only)
6. Error Handling (TCP, UDP → only through checksum) (Basic Error Handling)

* It uses a concept of Port No. (16 bit unsigned int)

80	HTTP
443	HTTPS
53	DNS
22	SSH
110	POP3
25	SMTP

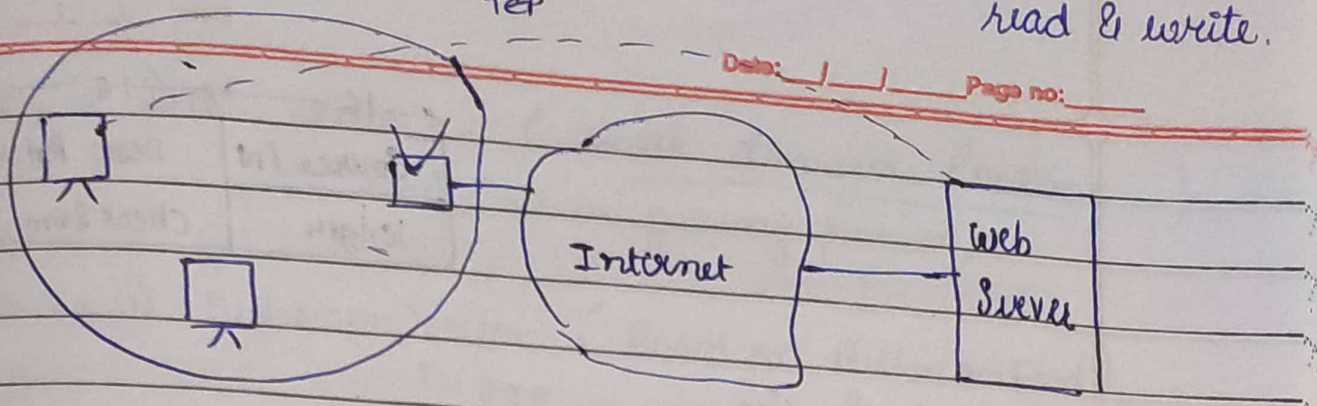
OS is req. for Port NO & for sockets

It is combination of ~~Port NO~~

1. Port NO
2. IPs
3. TCP/UDP (Transport layer Protocol)

Socket APIs

VDP only read & write.



socket()
connect()
write()
read()
close()

So

TCP Connection establishing.

socket()
bind()
listen()
accept()
read()
write()
close()

* HTTP works on TCP (Web Browser & Servers)

* DNS (Video Streaming) works on UDP

⇒	TCP	UDP
①	Heavy weight (20 Byte Header)	Light weight (8 Byte H.)
②	Connection-Oriented	Connectionless
③	Reliable	Delivery is not guaranteed
④	Extensive Error Handling	Basic E.H.
⑤	Order of packet maintained	Not Maintained
⑥	HTTP, HTTPS, SMTP, FTP, & SSH	DNS, DHCP, VoIP, Online games, Network Time Protocol
		Multicasting / Broadcasting

Suitable for

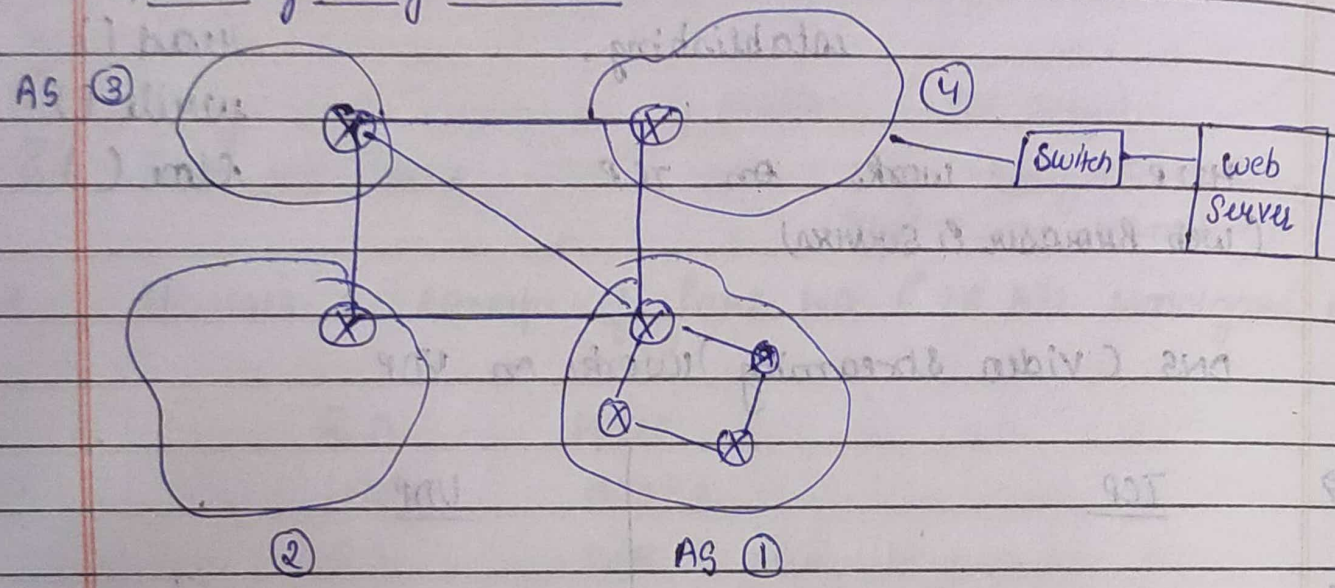
VoIP → Call / talk on phone

Date: / / Page no:

← 16 →	← 16 →
Source PN	Dest Port No
length	Check Sum

Quality Service

* Routing Algorithms



Travelling happens to reach packet from source to destination.

∴ Routing Protocols are imp.

2 Types

① Exterior gateway Protocol : Used to route across Autonomous system.

② Interior gateway Protocol : Used within an A.S.

Date: __/__/____ Page no: ____

Ext : Path Vector (inter-domain Routing)
↳ BGP (border gateway protocol)

Int : ① Distance Vector (Based on BellmanFord)
↳ RIP
↳ EIGRP (By CISCO)
(Not used much
② Link State (Dijkstra)
↳ IS-IS
↳ OSPF.
faster but every router needs all inf.

(Intra Domain Routing)

unicast + dynamic

① Static Routing :
Manually enter next hop.
Eg: small organization.

② Dynamic Routing.

* UniCast Only → 1 source - 1 distination

Broadcast → send to everybody.
Traffic → MultiCast → Subset of broadcast
Transfer Routing alg. send to multiple distination
AnyCast → speciae
→ content delivery
Traffic → any server which use the req URL
Routing alg. is fine.

Date: / / Page no:

All algo. are based on Cost.

* How do we decide cost of link ?

① Bandwidth (If using B.W. is most imp. for Rout. alg.)
② Network Delay
③ Hop Count
④ Path Cost
⑤ load
⑥ Max. Transmission Unit.

RIP → uses Hop Count

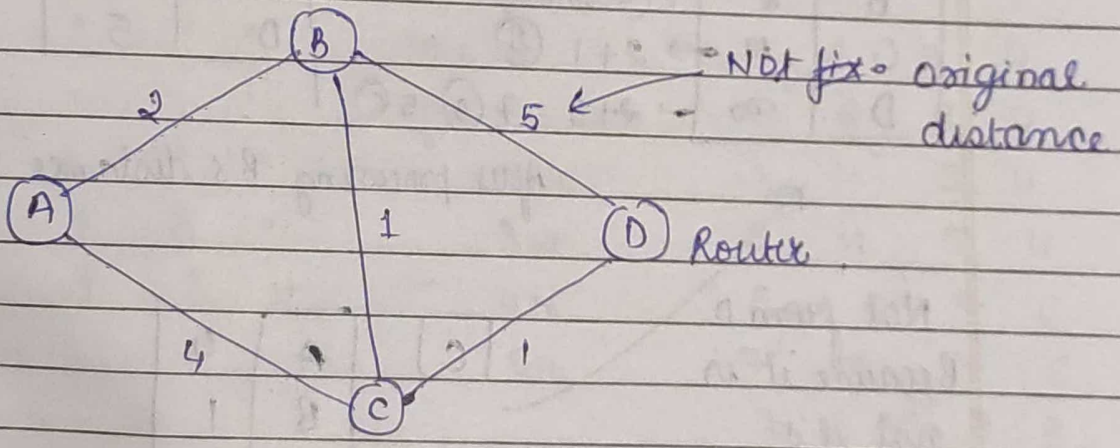
Goals

① should correctly deliver.
② Efficient utilization of bandwidth.
③ should not starve a Node.
④ should handle changes well.
(~~slow~~ fast Convergence)
↳ Router goes down
↳ link
↳ Cost changes
↳ New Router added.

Date: / / Page no:

Distance Vector Routing (DVR)

1. Oldest (used in ARPANET)
2. Bellman-Ford shortest path Algo.
3. Count to inf. prob. (Very slow convergence in case of failure)
4. RIP - used original DVR
   but EIGRP uses its modified version.
5. Less computations & message overhead compared to link state.
   (But Now Routers have good computation power & so Link state is much better)



Not ~~fix~~ original distance

- Each Node maintains Vector of distances & next hop for all distinations.
- Nodes periodically send vector to Neighbors.
- After receiving distance vector, Node simply updates shortest distance (if needed).

Date: / / Page no.

A

Dest	Dist	Next Hop
A	0	A
B	2	B
C	3	B
D	4	B

(Shortest)

required distant vectors

This column is sent

D		
	A	∞
	B	5
	C	1
	D	0

Distance Vectors

A		
	A	0
	B	2
	C	4
	D	∞

(A→B)+(B→C)
→ 2+1 (B)
→ 2+5 = 7 (B) 5 (C)

B		
	A	2
	B	0
	C	1
	D	5

After processing B's distance vector

Not from D Because it is not it's neighbour.

C		
	A	4
	B	1
	C	0
	D	1

* After every router has exchanged 1st (initial) distance vectors & have updated distances, every router knows shortest distance with almost one intermediate hop.

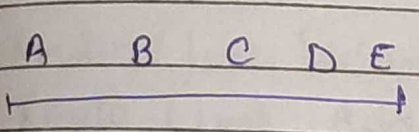
Date: / / Page no:

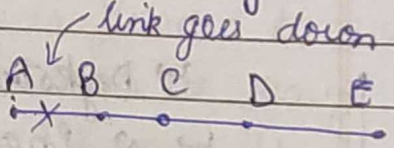
$$d(A,z) = \min\left(d(A,z),\ d(A,B) + d(B,z) \right)$$

generally after (V-1) iterations we get min distance but practically it is an iterative process. (due to router can go down)

Count to ∞ Problem in D.V. Routing

good news travel fast & bad news slowly





•	•	•	•	Initial
1	•	•	•	after 1st Exch
1	2	•	•	2nd
1	2	3	•	3rd
1	3	3	4	4th

1	2	3	4
3	2	3	4
3	4	3	4
5	4	5	4
5	6	5	6
7	6	7	6
7	8	7	8

:

Solution

① ~~Splitwise~~ } Not Efficient
② Poison Reverse

gfg course → CNT → Theory → Routing Protocol (DVR & LSR)

Date: / / Page no:

RIP (1988)

→ Used original D.V.
→ Hop count was used as distance.
→ Max. 15 distance. 16 was considered as ∞.
→ Use Poison Reverse & Split Horizon for Count to ∞ problem.
→ Router sends distance vector after 30 sec.

* Link State Routing

↳ Dijkstra's
↳ 2 Phases
① Flood inf. of Neighbours
② Every Node run Dij. after recieving Neighbor inf. of every other Node.
③ Widely used in Practice (IS-IS & OSPF)

It uses

→ Flooding Protocol :

If ① wants to send some packet then it (broadcast to) everybody) that is flooding.

Router detect duplicate & store 1 for eff. They themself do flooding.

Disadv. → Wastage of Bandwidth.

Date: __/__/____ Page no: ____

→ Every Node have inf. of whole graph.

OSPF (open shortest path first)

→ Every Nodes maintains source of packet & Seq. No. of packet to avoid duplication while flooding. * Process Neigh. inf. only when (Seq No from Same Source is Higher)

⇒ Every Node contains inf. of Neighbours which they flood

- Router / link Failure
- Router / link Add.
- Link cost chang

↙ this inf. is flooded to every ↙ Node as soon as any ↙ neighbour knows it.

↓

∴ It is must fasta & better

Date: / / Page no.

* Application Layer.

which uses diff. services

applications & built on this layer or app.

both are considered A.L.

as part of

Need:

Webbrowser + Youtube

There are diff. browsers & diff. servers.

∴ It is imp to maintain standard protocol.

So HTTP protocol or APP layer protocols are req.

App → App. Prs. Session.

~~Email App~~

Compute working

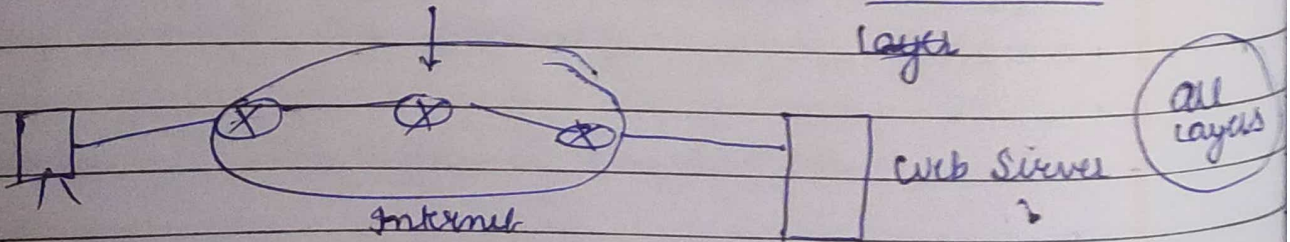
Applic. data with APP Header

Mac Header	N/W Router	TCP Header	App Data (1 … n)

↓ ARP

↓ to router

Intermediate Routers looks Mac H & N/W layer



PC — Web Server

Implement all layers

Date: / / Page no:

Int. routers implement 3 layers
1. N/w
2. DLL
3. Phy. L.L.

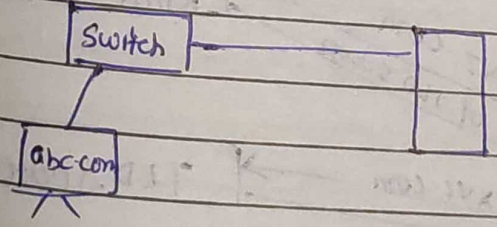
* Domain Name System.

Eg. Phone directory
↓
we know names but to comm. No. is req. So it maintains record of its mapping.

↓
converts URL to IP
↓
which user use to identify

↘ which comp. uses

Resolver (May be part of Router)
→ Provided by ISP
→ converts URL to IP



It have to do a lot of work if (URL - IP) is not present in Cache.

To communicate we need
1. Port NO. (HTTP - 80)
2. IP add.

(if Not in Cache)

Date: / / Page no:

Complex

13 Root Servers are present.

① Resolver goes to Root Server to ask IP for abc.com.

② It gives IP add of Top level, because it does not contain it. (Domain Server)

TLD Server → .com, .org, .in etc.

③ Resolver Now goes to TLD.

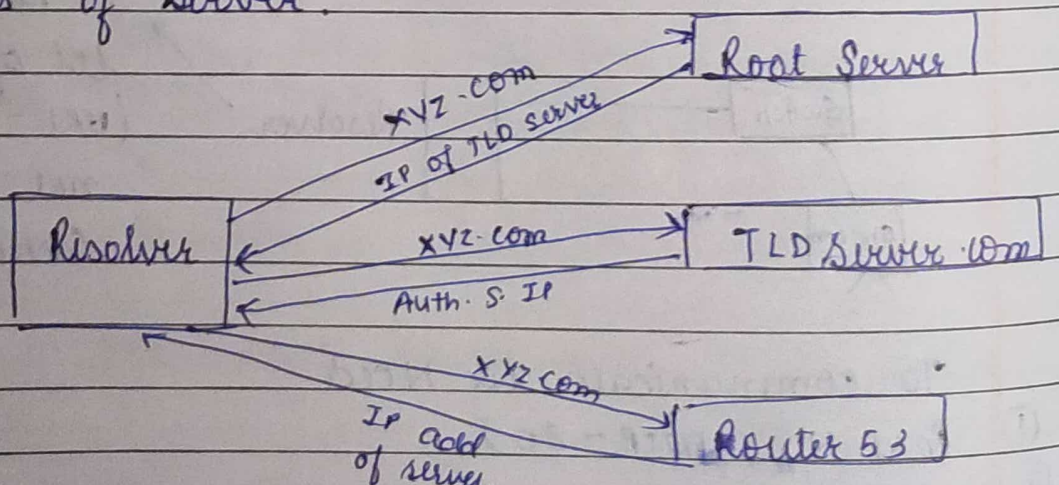
④ TLD gives IP add of Authoritative server.

(that have authority over Domain Name)

Predefined! & they Know everything about Server

Eg. AWS Route 53 ↓ (Port No.)

⑤ Resolver goes to Authority Server & it gives IP address of server.



Date: / / Page no:

TTL provide by Domain Owner.

↓

(Nowdays this are smaller)

(Time to maintain caches) :

URL to IP mapping :

* DNS Resolution
① Recursive Lookup (Resolver) ← whole resp. to find IP is given
② Iterative Lookup. (gives direction only NO Responsibility)

↓

TLD, Route53, Root server.

① Anycasting used by Root Server.
② DNS works on UDP

Pg No:

67

client 68 server.

(Application Layer Protocol) Date: / / Page no:

* DHCP (Dynamic Host Configuration Protocol)
(It is client-server model & based on discovery, offer, req, Ack)

⇒ (It do basic conf. automatically) DORA (4 Message)

Eg. If at office, just on your wifi & you are connected
Same ~~wifi~~ in Home

This is done using DHCP.

⇒ To connect to internet basic conf must be done.

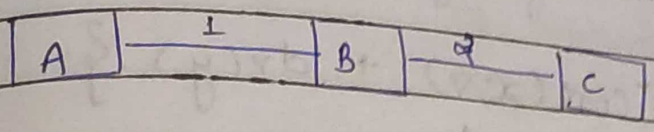
all 4 conf can be done
① Manually
② Automatically (DHCP)

Admistrator just sits the DHCP conf.
Now whenever New computer comes to Network it broadcast for DHCP ~~conf~~. In respond DHCP sends a unicast (only to that pc) msg. ~~to~~ & then do configuration.

They has release time after which IP assigned to PC ~~is~~ is removed & new IP is given.

→ ① Subnet Mask
② IP address
③ DNS Address
④ Vendor class Identifier

Count to ∞



B	1	A
C	2	B

A	1	B
C	2	B

A	3	B
B	2	C

Now if B — C is disconnected

B & C will update (neighbour)

B	1	A
C	3	B

A	1	B
C	-	-

A	3	B
B	-	-

Now B will ~~the~~ consider that there is path to C via A.

∴

B	1	A
C	3	B

A	1	B
C	4	A

(A responds to change in its neighbour B)

Now A will consider that there is path to ~~A~~ C via B.

B	1	A
C	5	B

A	1	B
C	4	A

Date: / / Page no:

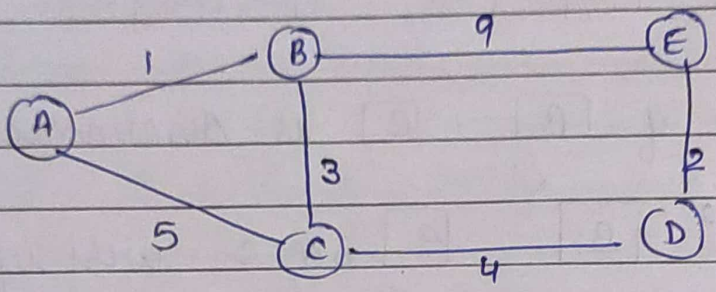
Distance Vector Routing Algo. (Bellman Ford)

⇒ $d_x(y) = \min\{ cost(x,v) + d_v(y) \}$

↓ ↘ ↓

Source dest in. intermed.

Eg



B source

A dest.

B Table

A	$\min(1, 8, (9+2+4+5)) = 1$
B	0
C	$(3, 6, 15) = 3$
D	$(10, 7, 11) = 7$
E	(9 ✓ 9)

↓ B-E B-C-D-E

Date: / / Page no:

* Bridge - Routing → Spanning Tree Protocol is used to ~~minimize~~ avoid loops in routing paths.

* Packets of same session may be routed through diff. path in TCP & UDP.

* Transmission time to transfer 1 byte for all Nodes be t

1st packet will take time = packet size * 3*t

other packets = packet size * t
(due to parallelism)

* Collision detection
↓
Min frame size = Round Trip Prop Delay * Transmission speed.

* Min. size of Packet = 2* Prop. Delay * Bandwidth

* PPP → Data link layer Protocol.

Date: __/__/____ Page no: ____

CRC (cyclic Redundancy check).

→ use binary division
→ forms codeword.
→ G(x) generator polynomial.

Remainder = 0 → No error
≠ 0 → error.

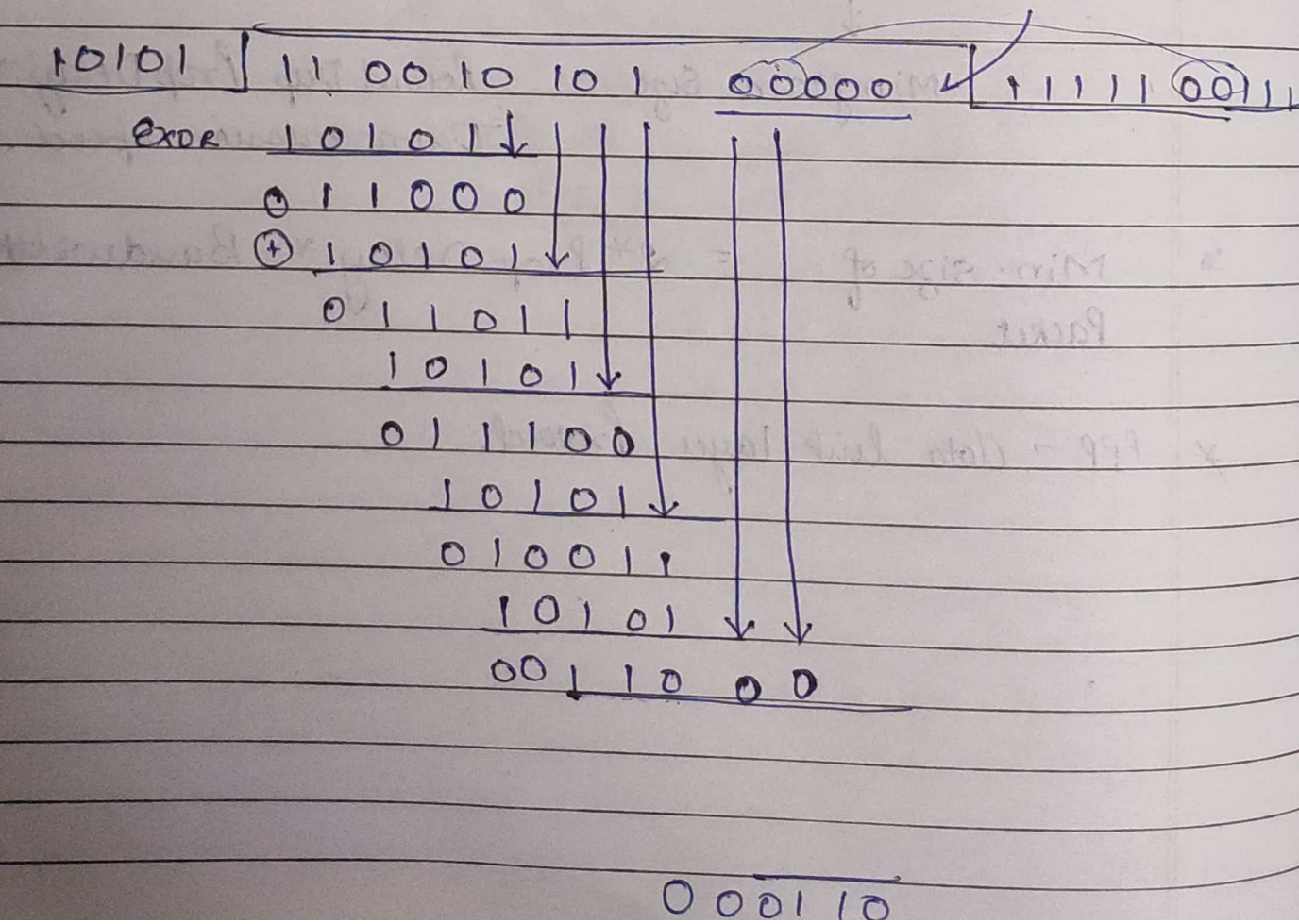
Prob.

① Cannot detect all types of error.
② dependent on divisor

Type 1 (sender)

1.1 00 10 10 1 divisor is 10101 (5)

generate CRC code.



Date: __/__/____ Page no: __

Codeword = Dividend + Remainder

```
  1 1 0 0  1 0 1 0  1 _ 0 0 0 0
                    0 0 0 1 1 0
```

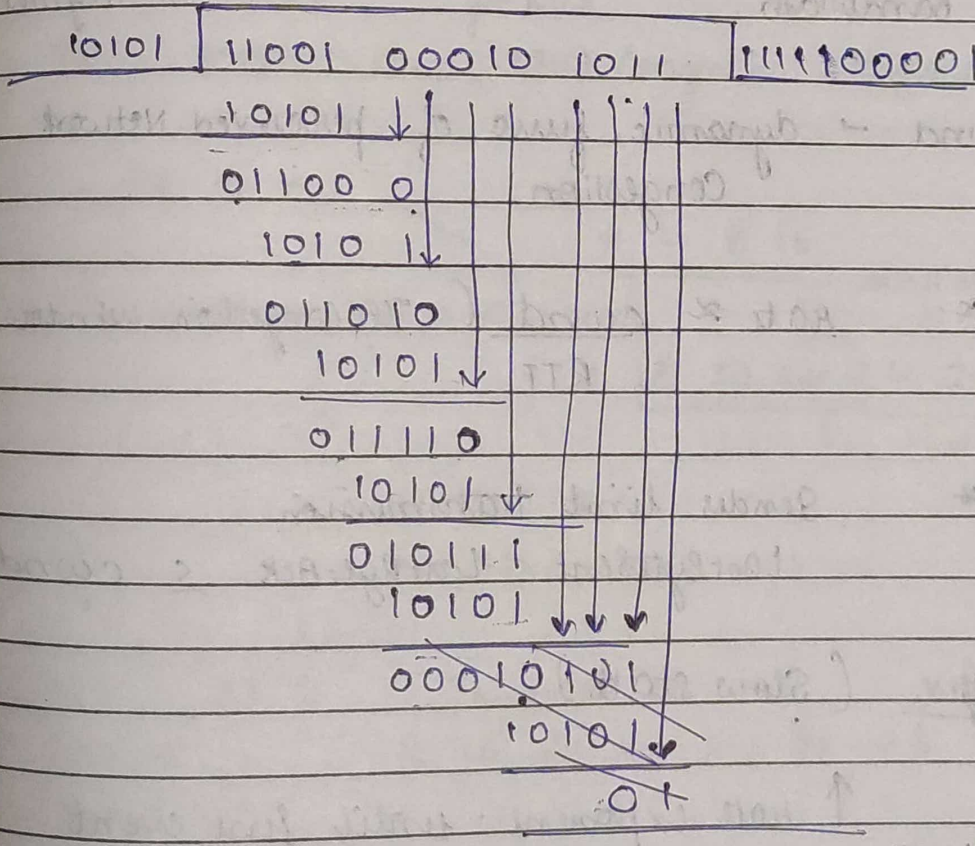
⇒ 1 1 0 0 1 0 1 0 1 0 0 1 1 0 ← Codeword

Types (Reciever's perspective)

We will divide codeword with divisor

Dataword ⇒ 1 1 0 0 1 0 1 0 1 0 1 1

Divisor ⇒ 1 0 1 0 1



Non Zero

∴ Error

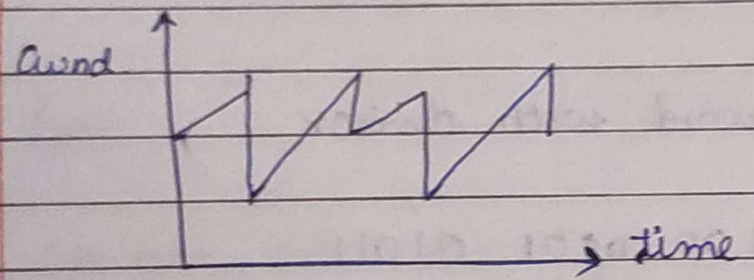
Date: __/__/____ Page no: ____

⇒ TCP Congestion Control

AI — Additive ↑ (↑ cwnd by 1 MSS every RTT until loss
MD — Multip ↓ . (cut cwnd in half after loss)

⇒ AIMD

Sender ↑ transmission rate (window size), probable for usable bandwidth until loss occurs



Congestion Window Size.

tooth Behaviour for Bandwidth.

MSS (Max. Segment Size)

cwnd → dynamic, func of perceived Network Congestion.

* rate ≈ cwnd / RTT (TCP congestion window)

* Sender limit transmission:
LastByteSent - LastByteAck ≤ cwnd

3 Phase

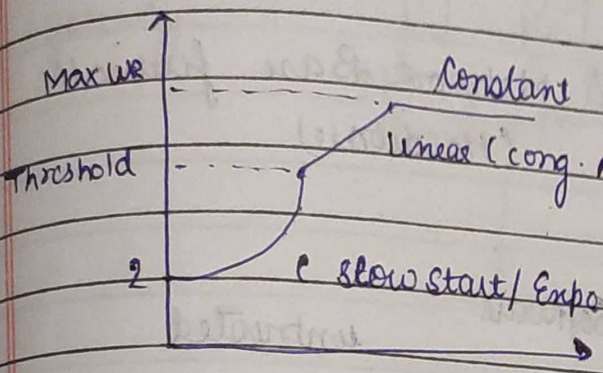
1st Phase (Slow start).

↑ rate exponent until first event
(initial rate is slow but ramp up exp. fast)

Date: / / Page no:

Congestion Detection

① Timeout occurs (Severe) starts from slow start
② 3 Ack Received. (Light) Linear



Q. initial window size = 32 KB
RTT = 100 msec
MSS = 2 KB
Time taken when 32 KB Congestion window get back

Timeout

⇒ ① 32 ↑ ⇒ 2 → 2 4 8 16 slow start

Threshold = 32/2 = 16

18 20 22 24 26 28 30 32
Linear

∴ 12 * 100

⇒ 1200 ms

3 Ack

Case ② 32 ↑ Threshold = 16

16 18 20 22 24 26 28 30 32

9 * 100 = 900 ms

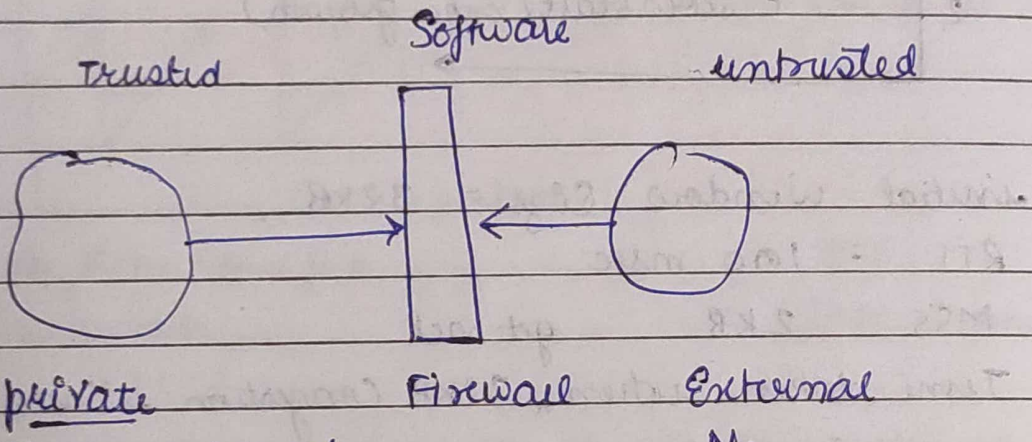
33/2 = |16.5|

= 16

Date: __/__/____ Page no: ____

Firewalls

→ Monitor & Control Incoming & outgoing traffic based on predefined rules.
→ Acts like Barrier.
→ Host Based (within Machine) Software & Network Base firewall (Hardware)



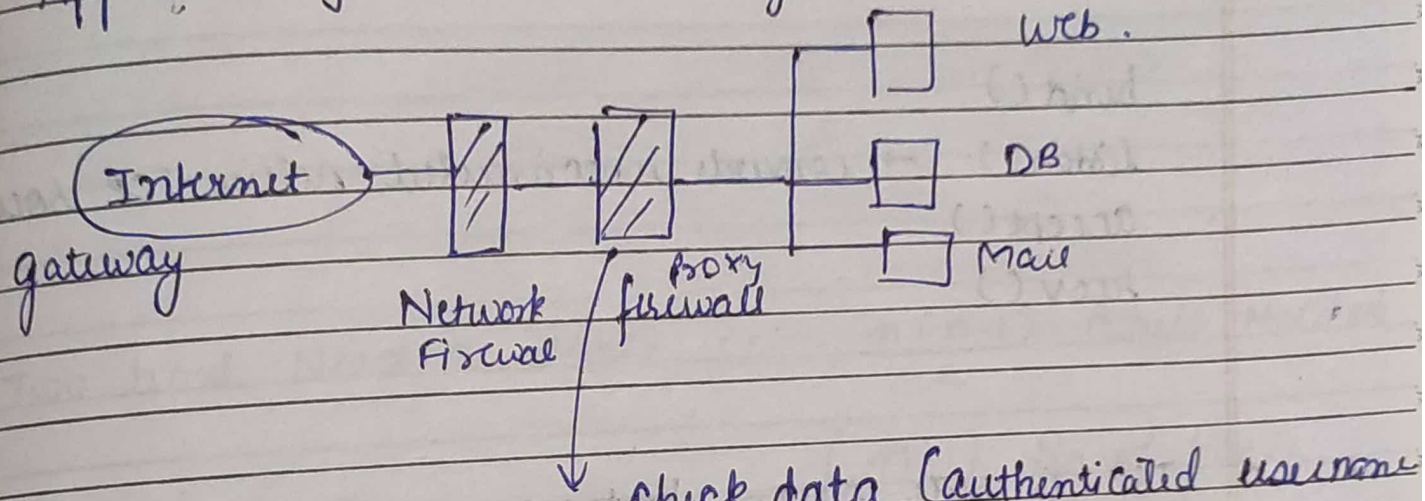
① Packet filtering firewall (layer 4) (layer 3 also)

→ Check IP header, TCP header (Port No)
→ works on Network & Transport layer.
→ Can block IP add, full Network.
→ Can block a service (Http)

	Rules.	Source IP	Source P.	Dest IP	Dest P
Default Allow	1.	179.9.4.80 & 152.32.0.0	A.	A.	A.
Mentioned not allow	2.	Any 152.32.0.0	A.	A.	A.
	3.	Any	A.	172.9.0.3	A.
	4.	Any	80 (Http)	A.	A.
	5.	Any	A.	A.	21 (FTP)

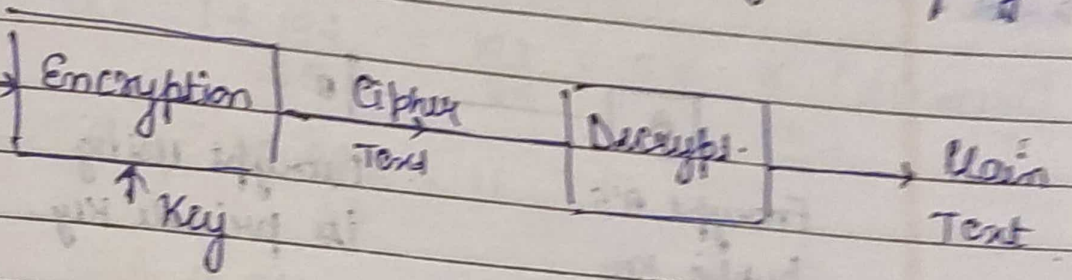
Date: __/__/____ Page no: _____

(2) App. (Proxy Firewall) Layer 5



Crypto - graphy

CIA → Availability
Confidentiality, Integrity



Symmetric Key (1 Key)
Asym. (2 Keys)

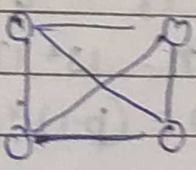
① Sym. Key (Same Key)

DES (Data Encry. standard) → 56 bits
3DES → 192 bits
AES (adv. Enc. Std) → 128, 192, 256 bits

$C = [K_1[m]]$ $M = [K_1[C]]$

Channel challenge
(Key exchange) : $^nC_2$:

n = No. of Keys



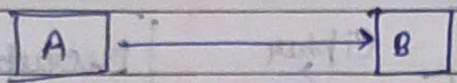
n = No. of Vertex.

$^4C_2 \Rightarrow 6$

Date: / / Page no:

Asymmetric (Public & Private Key) RSA

* Encrypt by Public & ∴ Decrypt only by its private & vice-versa.



Encrypt B's public key

Decrypt using its private key.

n devices
∴ No. of keys = 2n.

RSA

Q p = 13 q = 17 public key of A = 35
find private key of A.

⇒
① Chose p & q (prime No)
② $n = p * q$
③ $\Phi(n) = (p-1)(q-1)$ gcd
④ choose e $1 < e < \Phi(n)$, $(e, \Phi(n)) = 1$
⑤ Calculate d such that $de \equiv 1 \bmod \Phi(n)$
⑥ Public Key 'e'
Private Key 'd'.

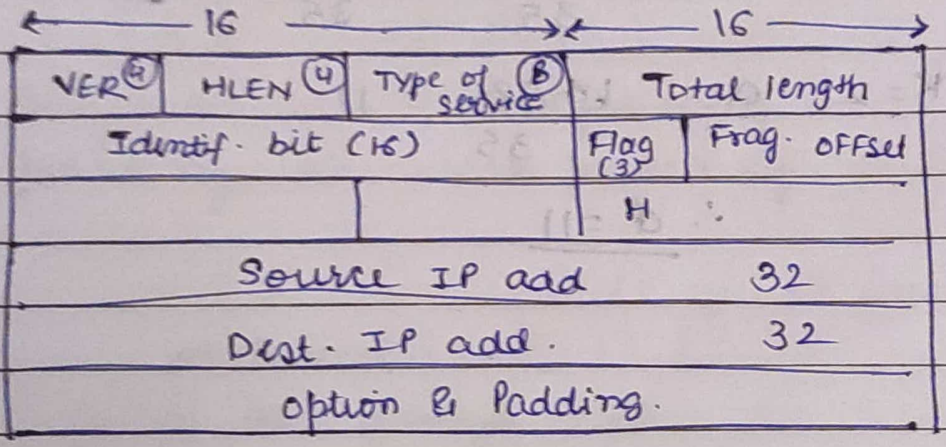
$p = 13 \quad q = 17$
$n = 13 * 17 = 221$
$\Phi(n) = (p-1) * (q-1) = 12 * 16$
$= 192$

$e = 35$ gcd(35, 192) = 1
⇒ $de \bmod \Phi(n) = 1$
$d * 35 \bmod 192 = 1$
∴ d = ...

Date: __/__/____ Page no: ____

Headers

① IPV4 Header (Internetworking Protocol)
- → Connectionless
- → Datagram Service (any Route)



160 bits = 20 byte

* Header size = 20 - 60 Bytes
* Payload = 0 - 65515 Bytes (data coming from app. layer)

① VER (Version) 100 (4)

② Header length (0000 to 1111) Multiply by 4
↓ 15*4 = 60

0, 1, 2, 3, 4 } Can't be used. 5*4 = 20

③ Type of Service (DSCP) Differentiated Service Code Point

Date: / / Page no:

Reliability cost

P	P	P	D	T	R	C	0

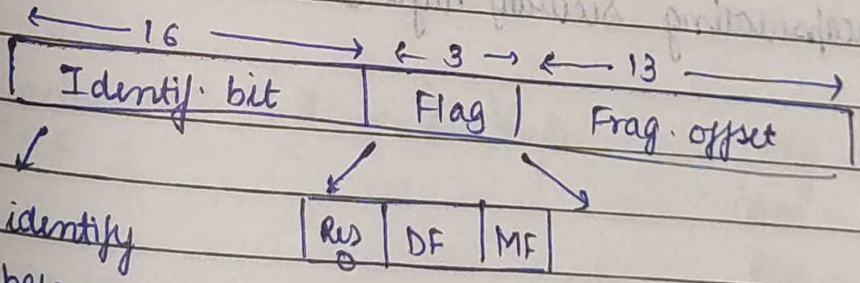
precedence · day throughput ·

④ Total length

⑤ Identification bit
⑥ Flag
⑦ Fragment offset
} unsp (used in fragmentation)

⑧ TTL → to avoid looping & congestion. (0-255)
⑨ Protocol → TCP/UDP (upper layer)
⑩ Checksum → (address detection) to chck integrity.

* Fragmentation

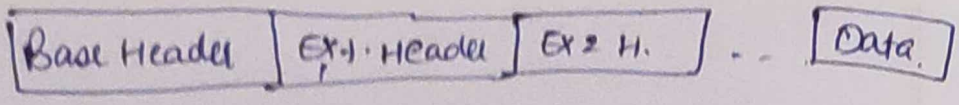


to identify it belongs to which fragment

Frag

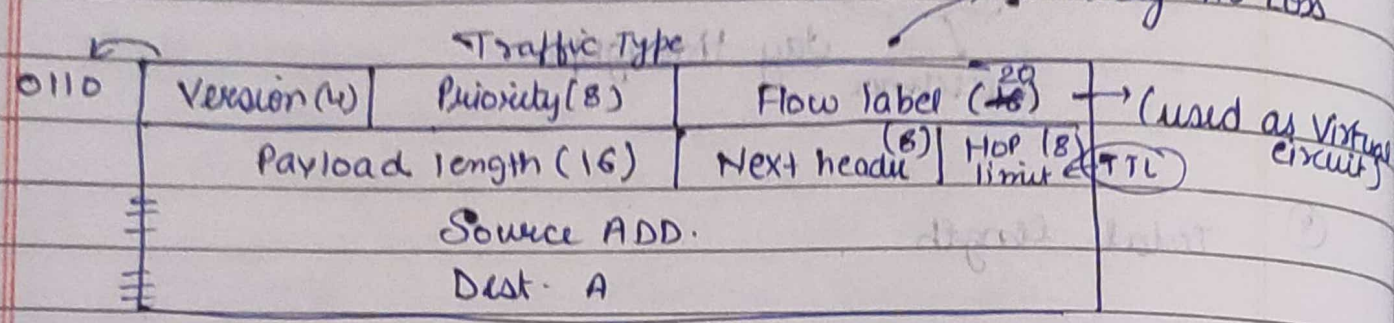
do not Fragment
0 ✓
1 ✗

more Fragment



Date: / / Page no:

IPV6 Header.



Base Header = 40 Bytes (320 bits) Fixed

Next header contains extension header.

1. Routing Header (43)
2. Hop by Hop op (0)
3. Fragment Header (44)
4. Authentication H (51)
5. Destination option (60)
6. Encapsulating Security Payload (50)

Date: / / Page no:

* TCP
  - Byte Streaming
  - Connection Oriented
  - Full Duplex
  - Piggybacking
  - Error Control
  - Flow Control
  - Congestion Control

Source Port (16)								Dest. Port (16)
Seq. No (32)								
Ack No (32)								
HLEN 4bit	// (6)	U R G	A C K	P S H	R S T	S Y N	F I N	Window Size (16 bit)
checksum (16)								URGENT Pointer (16)
option & Padding (40)								

MSS (max. segment Size)

Flag.

Header size → (20 to 60) Bytes

URG - urgent (1/0)
ACK → acknowl.
PSH → push
RST → reset
SYN → Synchronization
FIN → finish

* Port No. 0 - 65535
well K/a Port No 0 - 1023.

Date: __/__/____ Page no: ____

① No. of Subnets = (~~No of bits in difautt Mask~~
↘
(given Mask bits - No. of bits in default Mask)

② Subnet Address → (IP address) & (Subnet Mask)

③ Broadcast Address → put host bits 1 & returning Network bits as in IP address

④ No. of host per subnet = $2^{(32 - \text{given Mask bits})} - 2$

⑤ 1st host ID = Subnet add + 1

⑥ last host ID = Subnet add + No. of Hosts

In Case of Subnetting 4 steps are required for Inter Network Communication.

→ Source Host to Destination Network, Destination Network to proper Subnet, then Subnet to Host & finally Host to Process.

S. Host
↓
D. Network
↓
Subnet
↓
Host → Process

- Cost ↑
- Req. of Bridge, Router ↑
- experiences Network admin.

gfg Course → Theory → DHCP

DNS → Browser Cache
OS Cache
Router Cache
ISP DNS Cache

Page no:

DHCP DORA + adv + disad

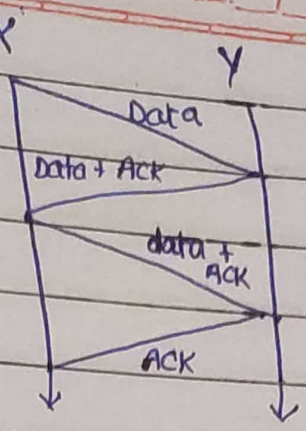
Q what happens when you type in a URL.

1. enter URL into web Browser
2. Browser looks up IP address for domain name via DNS.
3. Browser sends HTTP request to server.
4. Server sends back HTTP response.
5. Browser begins rendering HTML
6. Browser sends request for additional objects embedded in HTML
7. Once Page is loaded, Browser sends further async requests as needed.

→

1. Browser goes to DNS Server & find real address of the server ~~its~~.
2. Browser sends HTTP req message to server, asking it to send copy of website To the client.
   It is sent by client to server using TCP/IP.
3. If server approves client's request, server sends client "200 OK" message.
4. Then starts sending website file to the browser as series of small chunks called packets.
5. Browser assembles small chunks into complete web pages & display it to you.

Piggybacking:



Full duplex comm.
(forwarding channel data & recieving ack & vice-versa)

Traffic load doubles for each unit that is transmitted.

∴ Solution that provides better utilization of bandwidth is piggybacking.

* (sending acknowledgement is delayed until next data frame is available for transmission. Ack is then hooked onto outgoing data frame).