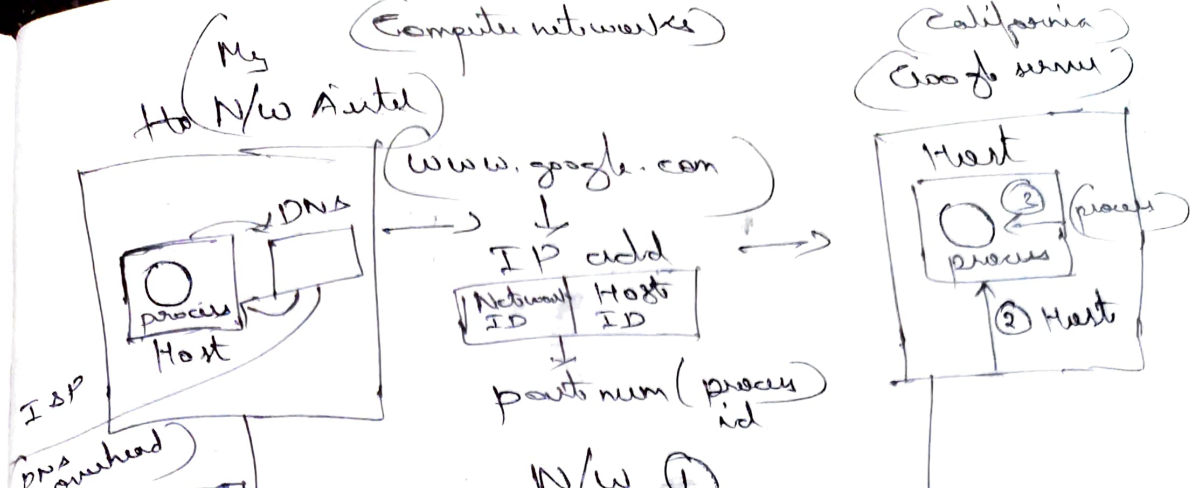
(Computer networks)



Http - 80
ftp - 21
SSHp - 22

(some default port/process id)

$2^1 - 2$

$2^2 - 4$

⋮

$2^{10} - 1024$ K

$m - 2^{20}$

$G - 2^{30}$

$T - 2^{40}$

(Number system)

u - 0, 1 - 0
2 - 00
3 - 000

B - 0, 1

D - 0, 1, 2, 3, 4, ... 9

Octa - 0, 1, 2, 3 .. 7

Hexa - 0, 1, 2, 3 ... 9, A, B, C, D, E, F

1 bit	2 bit	3 bit
0	00	000
1	01	001
	10	010
	11	011
		100
		101
		110
		111

(bits)
first $1 - 2$
second $2 - 2^2 - 4$
third $3 - 2^3 - 8$

K	n-K
network id	Host Id

(IP address)

(If n bits means 2^n - bits)

(K is the bit means 2^K parts - 2^n nums)

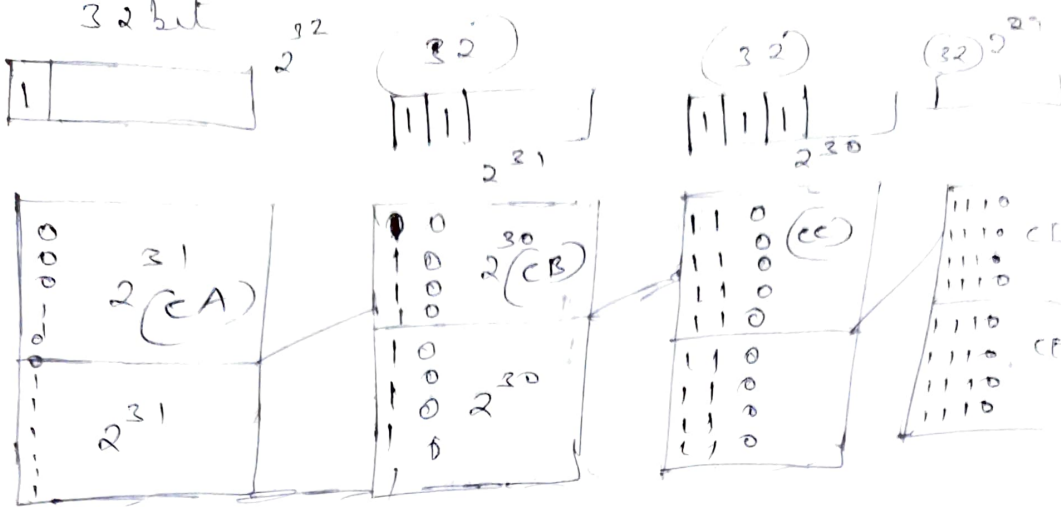
1 part $= \frac{2^n}{2^K} = (2^{n-K}$

(In CN) - IP address takes 32 bits

32 bits → 2^{32}

(This is not scalable 1980 where few network this is okay so we opt for class based ip address)

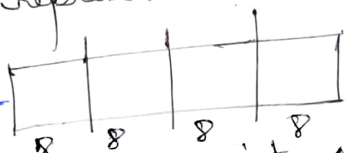
8	24
2^8	HID
NID	$2^{24} = 16$ M
$2^8 = 256$	



(class A) :-

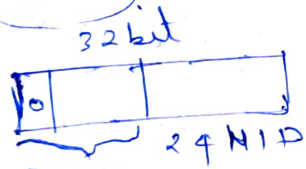
IP Address — represent 32 bit 0 & 1
— represent 32 bit into decimal convert

(popular method) :-



(divide 32 into 4 8 bits
convert 8 bits into decimal
& puts dot in middle called
dotted decimal number)

(CA)



1 bit occupied

$2^7 = 128$ Network of CA

$2^{24} = 16$ Million IP / NW

0 - - - - - - - - 0 X
0 0 0 0 0 0 0 1
0 0 0 0 0 0 1 1
⋮
1 1 1 1 1 1 1 - 127 X

(1 - 126) (Here 0 & 127 will not be choosen)

so class A network range from (1-126)

(NASA)

(CB)

(Range) 64

(128 - 191)

2 bit occupied

$2^6 = 64$

128.0	129.0
⋮	⋮
128.255	129.255

191.0
191.255 } 16K N/W

10 - - - - - - - - 128
0 0 0 0 0 0 0 1 - 129
0 0 0 0 0 0 1
⋮
1 1 1 1 1 1 1 - 191 } 64

(16 NID)

first 8 octal 64×2^8

$(2^6 \times 2^8 = 2^{14})$

$2^{14} \simeq 16K$

16K networks of CB

2^{16} = IP / network (64K)

(SBI
IRCTC)

(class - C) 32 bit (192 - 223)

1	1	0	

(2^{29}) 24 bits NID (8 HID)

(3 bit occupied)

2^{21} - 2m n/w of CC

2^{8} - IP/N/W (256)

110 _ _ _ _ _

0 0 0 0 0 - 192

⋮

1 1 1 1 1 - 223

$32 \times 2^{8} \times 2^{8}$

(32×2^{16})

192.0.0
192.0.1
⋮
192.0.255
192.1.0
192.1.1
⋮
192.1.255
⋮

class D (224 - 239

1110 _ _ _ _

0 0 0 0 - 224

⋮

1 1 1 1 - 239

(class E) (Range) (Military Application)

(240 - 255)

(Not used for any practical purposes)

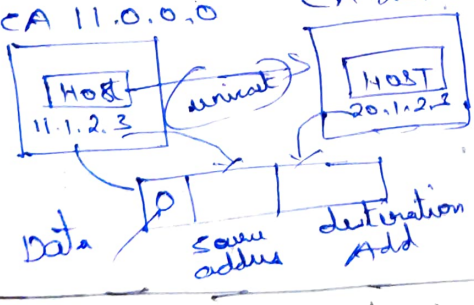
(2^{28} = 256m - ip address reserved for multicasting)

Send packet from one host to other host

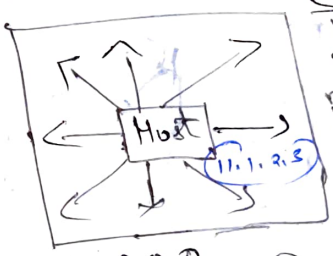
* Unicast - one to one
* broadcast - one to many

limited directed

CA 11.0.0.0 CA 20.0.0.0



Limited broadcasting (255.255.255.255)



Sending data to multiply data ip inside same network

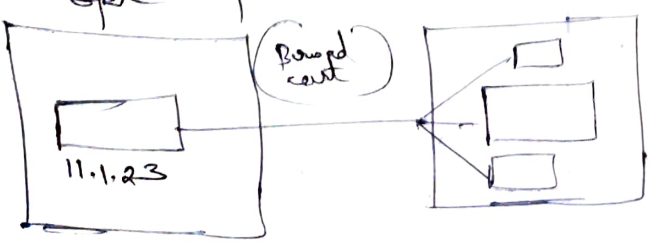
11.0.0.0

(8 bit) (24 bit)

data	SA	DA
m	11.1.2.3	255.255.255.255

(Directed Broadcast)

(one packet to all host in other network)



	SA	DA
m	11.1.2.3	20.255.255.255

$2^{n} - 2$ = Host

11.0.0.0 - (NID)

11.255.255.255 - (Broadcast)

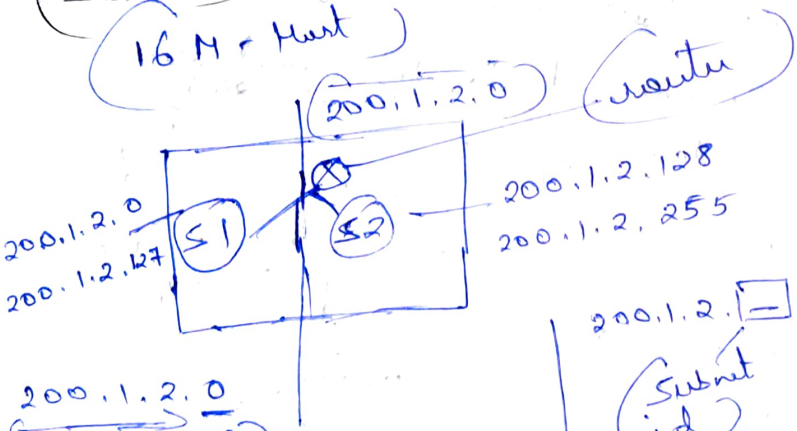
(CA - 1-126, CB - 128-191, CC - 192-223)

8 bit, 16 bit, 24 bit

IP	NID	DBA	LBA
(CA)			
1.2.3.4	1.0.0.0	1.255.255.255	255.255.255.255
10.15.20.60	10.0.0.0	10.255.255.255	"
(CB)			
130.1.2.3	130.1.0.0	130.1.255.255	"
150.0.150.150	150.0.0.0	150.0.255.255	
CC			
200.1.1~~50~~.100	200.1.10.0	200.1.10.255	"
220.15.1.10	220.15.1.0	200.15.1.255	
250.0.1.2	×	×	×
300.1.2.3			

(subnet)

(16 M - Host)



200.1.2.0

2^{24}, 2^8

$\boxed{0}$ 0 0 0 0 0 0 0

(during 2^8 Ip into 2)

0 - - - - - - -

0 0 0 0 0 0 0 1 - 1

0 0 0 0 0 1 0 - 2

⋮

0 1 1 1 1 1 1 1 - 127

(S1 range) 200.1.2.0
200.1.2.127

200.1.2.$\boxed{-}$ - - - - - - -

(subnet id)

1 0 0 0 0 0 0 0

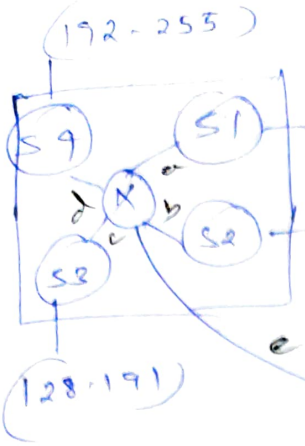
1 0 0 0 0 0 0 1 - 128

1 0 0 0 0 0 1 0 - 129

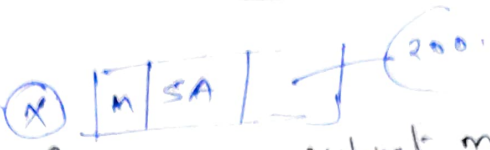
⋮

1 1 1 1 1 1 1 1 - 255

(S2 range)
200.1.2.128
200.1.2.255



If a outside wants to get into subnet, they go to router & router divide them to correct network (Here 20)-20<1)



(subnet masking) (router uses subnet masking to redirect correctly)

1's - NID & SID (200.1.2.130)

0's - HID

11111111	11111111	11111111	11000000
255	255	255	192

11 001000 . 00000001 . 00000010 . 10000010

And (Subnet mask & Ip 200.1.2.130) (And operation)
(Ip in binary format)

11001000 . 00000001 . 00000010 . 10000000

200 . 1 . 2 . 128

(Given (IP & SM) we can find the network the IP address belongs to)

(Routing table)

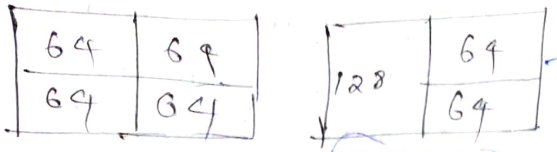
NID	SM	Interface
200.1.2.0	255.255.255.192	a
200.1.2.64	"	b
200.1.2.128	"	c
200.1.2.192	"	d
0.0.0.0	0.0.0.0	e

If incoming Ip (200.1.2.10) means this will do And operation with SM & sent it to the interface that matches the NID

(default entry) (send back)

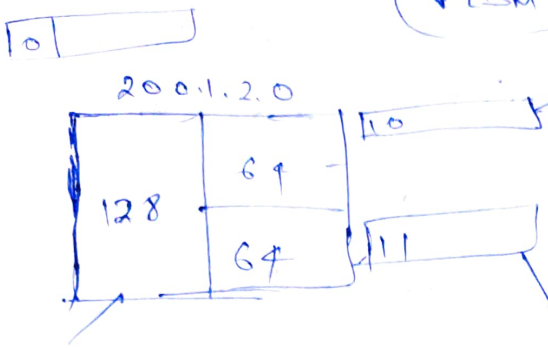
If 2 subnet match send that has many 1's in SM, interface

* (Variable length subnetting)



(different requirement)
(different sizes)

(VLSM)



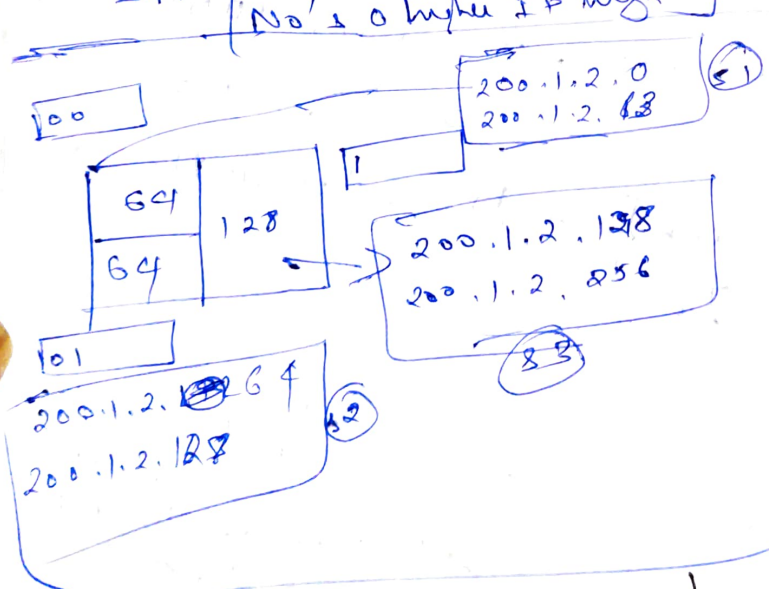
200.1.2.10 - - - - - - -
128 - 0 0 0 0 0 0
191 - 1 1 1 1 1 1
200.1.2.128
200.1.2.191
SM - 255.255.255.192
1 . 1 . 1 . 11 000000

200.1.2.0 - - - - - - - -
00000000 - 128
⋮
01000000 - 128

200.1.2.11 - - - - - -
200.1.2.192
200.1.2.255
SM - 255.255.255.192

200.1.2.0 - (NW id)
200.1.2.128 - (direct broadcast)
subnet masking - (255.255.255.128)

SM { No's 1 higher IP low
No's 0 higher IP high }

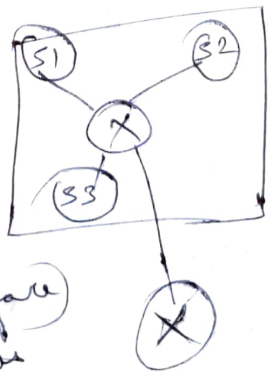


(Another way all are correct)

(2 matches same send to the interface with longest 1's SM)

NID	SM	interface
200.1.2.0	255.255.255.128	a
200.1.2.128	255.255.255.192	b
200.1.2.192	255.255.255.192	c
0.0.0.0	0.0.0.0	d

(SM & dest IP) → interface matches (NID)



255.255.255.192

11111111.11111111.11111111.11000000

8 8 8 2

26 1's

(NID + SID)

6 0's

($2^6 = 64$ IP/LN)

CA : 8 + SID = 26

SID = 18

(SN = 2^{18})

CB : 16 + SID = 26

SID = 10

(SN = 2^{10})

CC : 24 + SID = 26

SID = 2

(SN $2^2 = 4$)

(255.255.255.15 in CC)

(How many SN possible)

NID + SID = 28

24 + SID = 28

SID = 4

$= 2^4$ (16 SN possible)

Subnet mask	No of Hosts	(min 255.0.0.0) subnets in class A	(min 255.255.0.0) subnets in class B	(min 255.255.255.0) subnets in class C
255.0.0.0	$2^{24} - 2$	1	X	X
255.128.0.0 (8 1's (only 1's))	$2^{23} - 2$	2^1	X	X
255.192.0.0	$2^{22} - 2$	2^2	X	X
255.240.0.0	$2^{20} - 2$	2^4	X	X
255.255.0.0	$2^{16} - 2$	2^8	1	X
255.255.254.0	$2^9 - 2$	2^{15}	2^7	X
255.255.255.0	$2^8 - 2$	2^{16}	2^8	1
255.255.255.224	$2^5 - 2$	2^{19}	2^{11}	2^3
255.255.255.240	$2^4 - 2$	2^{20}	2^{12}	2^4

Classless Inter Domain Routing (CIDR)

NID | HID
BID

a.b.c.d / n — (no of bits in NID) (Block CIDR)

(CIDR)

20.10.50.100	20

NID : 20

HID : 32 − 20 = 12

IP = 2^{12}

Rules for CIDR formation

1. All IP address be contiguous
2. 2^n — Block should be
   n — binary bit

(n − K) K / 2^K

1011 /	2^1	1	5
	2^2	3	2
1011	2^3	3	1
1011	2^4	11	0
1011			

(LSB) Remainder
MSB — Quotient

3. first IP address in the block should be evenly divisible by the size of the block

(Block id)
100.1.2.32
100.1.2.33
⋮
100.1.2.47

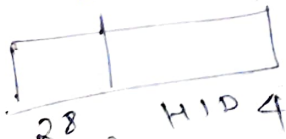
(1) contiguous
(2) $2^n = 16$
2^4
(3) 100.1.2.00100000 (2^4)

All rules satisfy

2^4 (DBA)

$2^4 - 2 = (16 - 2 = 14)$
out of 16 only 14 can be used

2^4



28 HID 4
(32 − 4)

2^4 : IP
HID : 4
NID / BID } = 28

Means
28 bits in block id part
4 bits in Host id part

(100.1.2.32/28)
(100.1.2.35/28) — find range
100.1.2.00100011
8 8 8 4
(28) (BID)

HID
0000 − 32
1111 − 47

Range b/w
32 to 47

(VLSM in CDIR)

20.30.40.10/25 - (Isp)

(ISP) 2^7 (25) - 25 - BID

7 - HID

2^6 2^6

company (A) company B (Keep itself)

20.30.40.0

weight (64) 0 0 0 0 0 0 0 - 0 } (range)

1 1 1 1 1 1 1 - 127

1 0

64 32

20.30.40.64/27

20.30.40.0/26 - 2^6 CA

2^5 CB

2^5 (ISP)

20.30.40.96/27

00

20.30.40.0/27 2^5 (CB)

20.30.40.64/26 2^6 CA

01

20.30.40.32/27

(LSP)

(splits into 2 parts)

(first part split into 2)

2^{7-1}

64 32 (weightage)

2) 40.30.10.10/20

20 - BID

12 - 2^{12} - HID

2^4 (CA)

2^{11}

2^{10} (CB)

2^{10} (itself)

40.30.0000 1010.00001010

BID 2^{32} 00.00000000

2^2 11111111 11111111

40.30.0.0 } (range)

40.30.15.255

$2^3 2^2$

0

10 ($2^3 2^2$)

40.30.0.0/21

40.30.7.255/21

40.30.8.0/22

40.30.11.255/22

11 ($2^3 2^2$)

40.30.12.0/22

40.30.15.255/22

Subnet mask & Network mask same

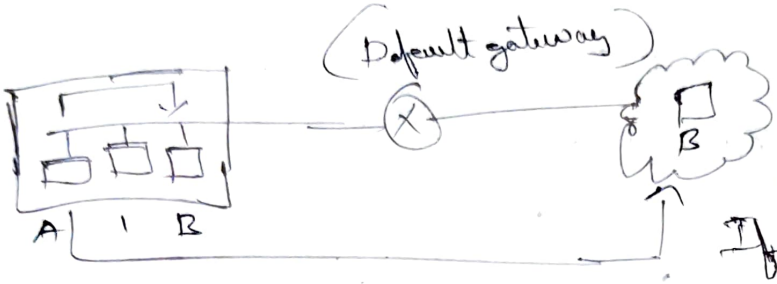
(ISP provide)

IPV4 - (IP address)

Default gateway - default router connected to network

SM - subnet mask

DNS - (Domain to IP address)



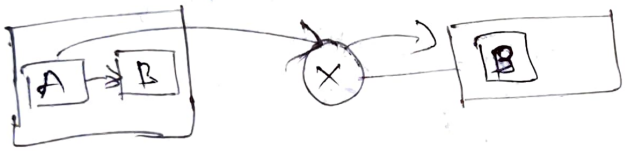
A : I_A B : I_B

S_A S_B

NID_{AA} NID_{BA}

If equals same network

If not equals (router)



A : I_A : 200.1.2.10

S_A : 255.255.255.128

(B : I_B : 200.12.130)

I_A : 1100 1000 . 00000001 . 00000010 . 00001010

S_A : 1111 1111 . 11111111 . 11111111 . 1000000

NID_{AA} : 11001000 . 00000001 . 00000010 . 00000000

(200 . 1 . 2 . 0)

I_B : 11001000 . 00000001 . 00000010 . 10000010

S_A : 11111111 . 11111111 . 11111111 . 10000000

NID_{BA} : 11001000 . 00000001 . 00000010 . 10000000

200 . 1 . 2 . 128

I_A : 200.1.2.10
S_A : 255.255.255.128

I_B : 200.1.2.69
S_B : 255.255.255.192

I_A
S_A
NID_{AA} : 201.1.2.0

I_B
S_A
NID_{BA} : 200.1.2.0

I_B
S_B
NID_{BB} : 200.1.2.64

$NID_{AA} == NID_{BA}$
(A according A == B according A)
(B according B == A according B)

I_A
S_B
NID_{AB} : 200.1.2.0

0-127	128-255

This is how A thinks B (SM thinks)

0-63	64-127
128-191	192-255

This is how B thinks of A (SM thinks)

A → B

(Access control risk)

Sm : 255.255.255.255
I_A : (And)
(I_A = NID)

SM
I_B
I_B

(router A int)

(By this mechanism I_A & SM gives your IP to your network, so this let you confirm no one can be in your

(Supernetting & Aggregation)

1) contiguous network
2) size same (2^n)
3) HNID (2^n)

200.1.0.0/24 * contiguous
200.1.1.1/24 * (2^8) same size
200.1.2./24 * $4 \times 2^8 = 2^{10}$
200.1.3/24

(200.1.0.0/2^{10})

network mask (same)
Subnet mask — 255.255.255.0
supernet mask 255.255.252.0

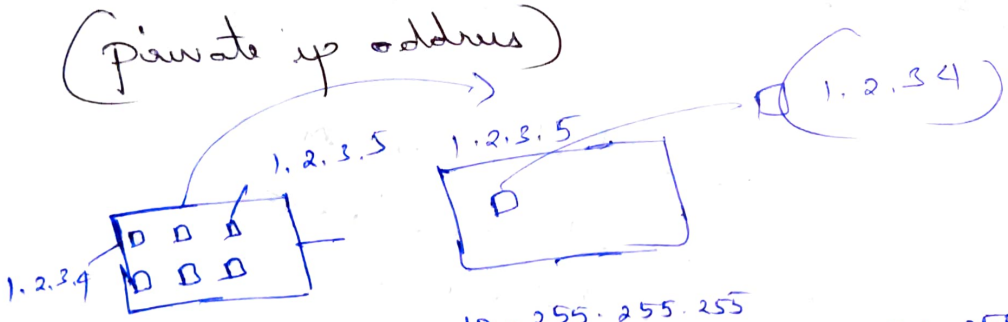
IP & NM → NID

200.1.00000000.00000000
200.1.00000001.00000000
200.1.00000010.00000000
200.1.00000011.00000000

255.255.252.0

Supernet : 2^8 [HID - 12
NID - 20]

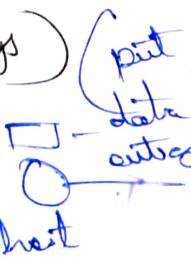
(Private ip address)



class A 10.0.0.0 – 10.255.255.255
class (B) – 16 address (172.16.0.0 – 172.31.255.255)
class (C) – 192.168.0.0 – 192.168.255.255

(Delays)

* Transmission delay



Bandwidth = 1bps
pkt = 10 bits
(10 secs)

L bits (data) B bps (tranmo)

$T_t = (L/B)$ seconds

Data : $2^n = 1024$
Bandwidth : $K = 1000$, $m = 10^6$ | $G = 10^9$

* (propagation delay)

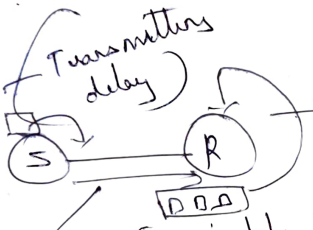
last bit to receiver @ receiver end Tp propagation delay

$T_R = \left\{ \frac{\text{Distance}}{\text{velocity}} \right\}$

$T_p = \frac{d}{v}$

velocity = speed inside optical fibre = $(3 \times 10^8 \times 0.7)$ m/s

speed of light of 70 %



Transmitting delay

processing delay

To take last date

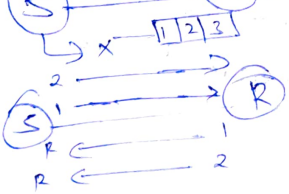
queuing delay — All data will be queued & then processed

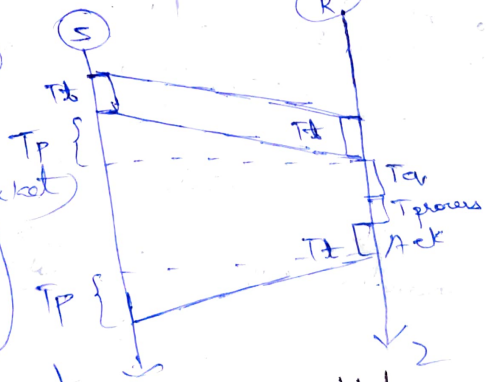
propagation delay

$T_t + T_p + T_q + T_{pro}$ — depends on processor @ (R)

Flow control mechanism

Here sender sends data @ rate that receiver cannot process





Total time = (one packet)

$[T_t \text{ data} + T_p + T_q + T_{process} + T_{ack} + T_p]$

Throughput = (send) No of bits / second } effective Bandwidth / Bandwidth utilisation

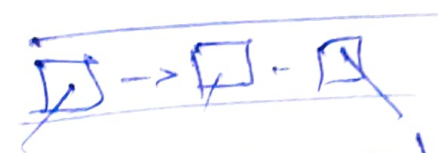
Capacity of channel / wire / link

Half duplex - ═══ - one way data / bits sending

full duplex - ═══ - two way data / bits sending

Half - BW * Tp

full - 2(BW * Tp)



1 Mbps 1 Mbps - 1 Mbps

(10 sec - Tp)

(1 Mbps)

10 sec for one Mps to reach receiver @ same time another 1 Mbps starts sending

3
3

(Computer Networks)

protocols - sets of rules set by Internet society

TCP, UDP, Http, Packets, DNS

IP - (X.X.X.X) NAT (modem)

(0 - 255)

Internet → Google.com

ISP

modem / Router

global IP address

Network Address translation ensures to route the response from where request was placed

IP1 D1 — google.com

IP2 D2

IP3 D3

DHCP → Assigns IP addr

(ports)

(16 bits nos)

[IP/port] C1 ⇄ [IP/port] C2

$2^{16} \approx 65000$ ports

$(2^{16}-1)$ 65535

Reserved ports

(0 - 1023)

(1024 - 49152)

(Application ports)

Http - 80

Https - 443

SSH - 22

ftp - 21

sftp - 22

MongDB - 27017

Sql - 3306

(Speed)

1MBPS - 1 Mega bits/second = 1000000 bits/sec

2 ways of connection
- guided (wired)
- unguided (wifi, bluetooth)

physical :- optical fibre cable, coaxial cable

wireless :- Bluetooth, wifi, (3G, 4G, 5G LTE) (long range)

(wired faster than satellite)

LAN :- small house / office / ethernet / (wifi) — local area network

NAN :- Across city — metropolitan area network

WAN :- Across country (fibre optical cables) — wide area network

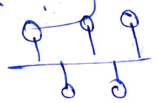
- SONET (synchronous optical network)
- Frame relay (connects LAN to WAN)

Modem :- (digital ⇄ Analogue) (Both side)

Router :- Routes the packets based on IP address

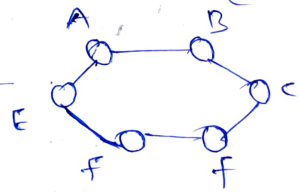
(Topologies :-)

① Bus



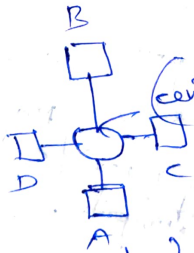
{ one (sender) @ a time }

② Ring



{ A to F goes through B & C
limitation
* cable break
* unnecessary calls to B & C }

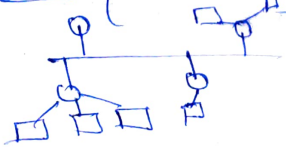
③ Star



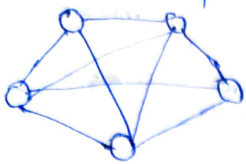
(centralized device)

{ If CD fails, }

④ Tree { Bus + star }



5) Mesh (Every single computer connected to every single computer)



* Expensive
* scalable issues (each new should connect to all)

Structure of Networks

OSI (open system inter connection model)

Application	software (send msg to presentation layer) HTTP
presentation	Extra msg from Application layer & converts data to machine code / encoding / encryption (SSL)
Session	managing / termination of connection (Authorize Authori)
Transports	Segment, flowctrl, error ctrl (tcp udp)
Network	routing (communicates to other network) ip packets { sender receiver logical addr
Data link layer	physical addres (mac address) frame (sender & address receiver)
physical layer	Hardware (wires cables) (bits transfer)

TCP / IP Model

Internet protocol suite (Another model) (practically used)

Application

Transport

Network

DLL

physical layer

① (Application layer)

* user interaction
* whats App, browser

whats App
- process : send msg
- thread : sets up page
- record video (open camera)

(protocols)

most protocols

* TCP/IP
  1. HTTP
  2. DHCP (Dynamic host ctrl) (Allocate IP addr to devices)
  3. FTP
  4. SMTP (simple mail transfer) - (send email)
     (POP3 & IMAC - receive email)
  5. SSH
  6. VNC
* Telnet - (port - 23)

* UDP
  stateless
* socket - (Interface b/w process & internet)

(ports) :
- Ephemeral ports { Assign random ports & once process is completed the ports will be freed (like multiple tabs on chrome each tab assign random ports }

(Http) : - (App layer)

* uses TCP - (Transport layer)
* stateless

(cookies) : (unique string) stores in browser

(Third party cookies)

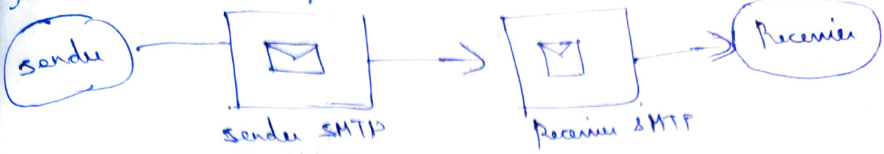
Status code

1XX → Information
2XX - success
3XX - Redirection
4XX - client error
5XX - (server error)

Send - SMTP

receive - POP 3

(establish connection)



(DNS) - (ICANN org for storing DNS)

maps.google.com

(subdomain) (secondary domain) (top level)

top

.com .net .org

(google.com)

1. local cache
2. local DNS server (ISP)
3. Root server for .com
4. top level domain

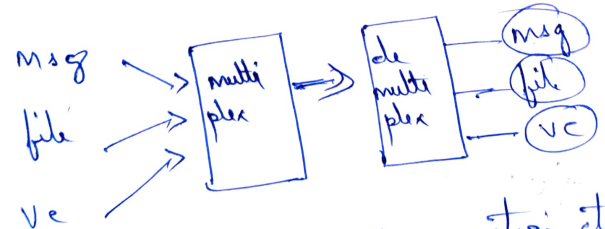
(Transport layer)

Network - sends data from one network to another

TL - gets data from network & sends to app layer

TL attaches port nos

data travel in packets



* TL takes care of congestion ctrl
* congestion ctrl Algo in TCP

(check sum)

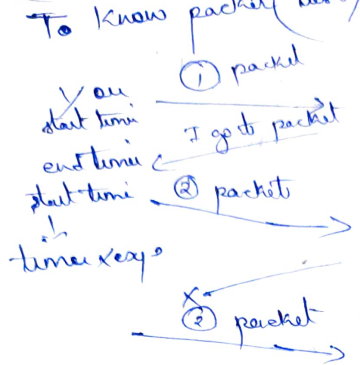
data + checksum (Both equal) (no corrup)

(destination) ← source

data + checksum

(Timers)

To know packet (lost/not) (timer is used)



(To solve this sequence nos)

(UDP) - (user datagram protocol)

*) data may not be delivered
*) data may change
*) not in order
*) connection less
*) uses checksums

Source port No	length of data gram
Destination port No	check sum
Data	

Total size = 2^{16}

Header = 8 bytes

($2^{16} - 8 = 65536$ bytes)

*) fast
*) video conference app
*) gaming

(TCP)

*) Transport layer
*) App layer sends raw data TCP segments
this data → divide chunks & add headers
*) congestion ctrl
*) Takes care
- when data not arrived
- maintain order of data (seq nos)

features :-

* connection oriented
* Error ctrl
* congestion ctrl
* full duplex (bi directional)

(3 way handshake) :-

client —syn - (flag)→ server
seq No - 32

ACK
seq no - 32 - (with on 32) (new no) (eg : 56)

ACK
seq no : 33

(Network layer) — (Routers) Hop by Hop (forwading / Routing table)

Transport - segments
Network - packets
DLL - frames

A — R1 — R2 — R3 — B

control Plane :-

* Builds the Routing table

Router - Nodes
links - Edges

① static Routing
Manually add routes

② Dynamic Routing
Evolves
(path finding algo
DJIKTRA)

(protocols :-)
IP (Internet protocol)

IPV4 - 32 bit, 4 words
IPV6 - 128 bits

192.168.2.30
network addr (subnet id) → device address (host id)

class of IP address

A - 0.0.0.0 - 127.255.255.255
B - 128.0.0.0 - 191.255.255.255
C - 192.0.0.0 - 223.255.255.255
D - 224.0.0.0 - ~~255~~ 239.255.255.255
E - 240.0.0.0 - 255.255.255.255