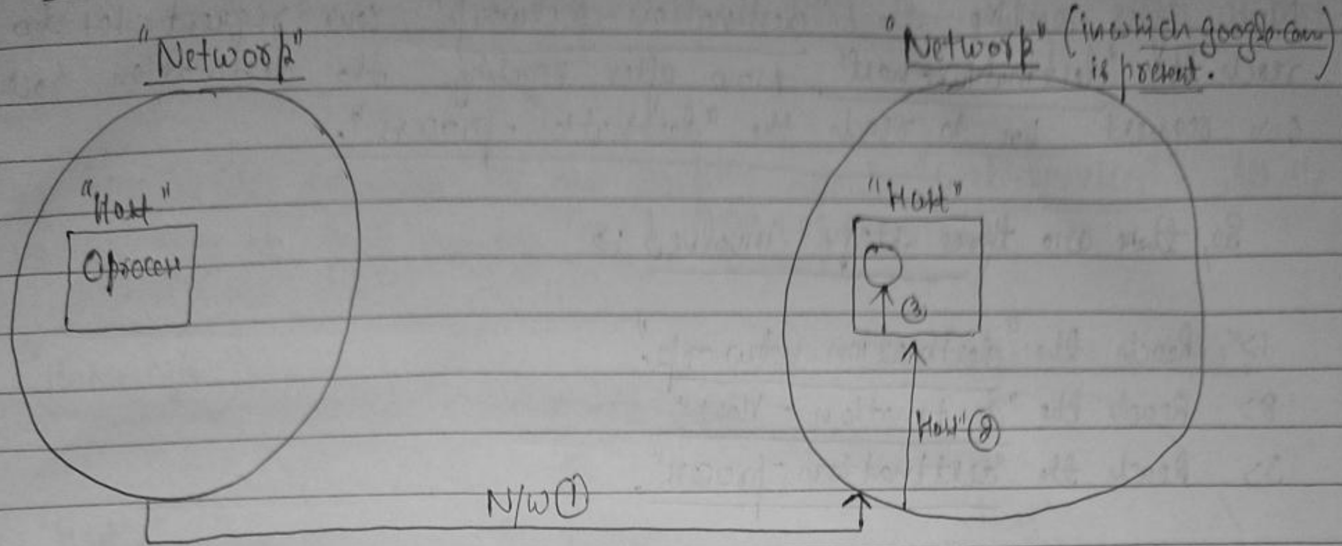
"GATE"

"COMPUTER-NETWORKS"



Given a "Network", there are many hosts present in a "network". (Host means a computer). Now within that "Host" there are basically many processes running.

Let us assume that once process is "we have opened our web-browser & have typed "www.google.com". Now we want to go to google & get a "web-page"

Now assume that "Google's network" is in "California". & Inside that network one of the server is "Google-Server". & one of the process with the Google network's "Host" is the "Server" we want to connect to.

It means from our "process" at "Home" we want to connect to a "process" of "google" in order to get the web-page.

In the browser, we have typed "www.google.com"

↓

"This is known as domain name or URL"

Now, this "domain name" has to be converted in such a way that, we are going to identify the network address.

So, we should first reach the "destination Network".

Now After reaching the "destination-Network", our request has to reach the "destination-host". Now after reaching the destination host, our request has to reach the "destination-process".

So, there are three steps involved :→

1> Reach the "destination network".
2> Reach the "destination - Host"
3> Reach the "destination-process".

But we are given only the domain name "www.google.com". Now using this "domain-name"

→ we have to identify the "destination n/w".
→ then the "destination host".
→ & then the "destination process."

So, somehow we have to convert this domain name ["www.google.com"] into some "numbers". Called "IP-address"

So, there has to be some service, which will convert "Domain-names" into "IP-addresses".

This "IP-address" consists of two parts:-

"IP add"

"N/w Part"	"Host Part"
NW ID	HOST ID

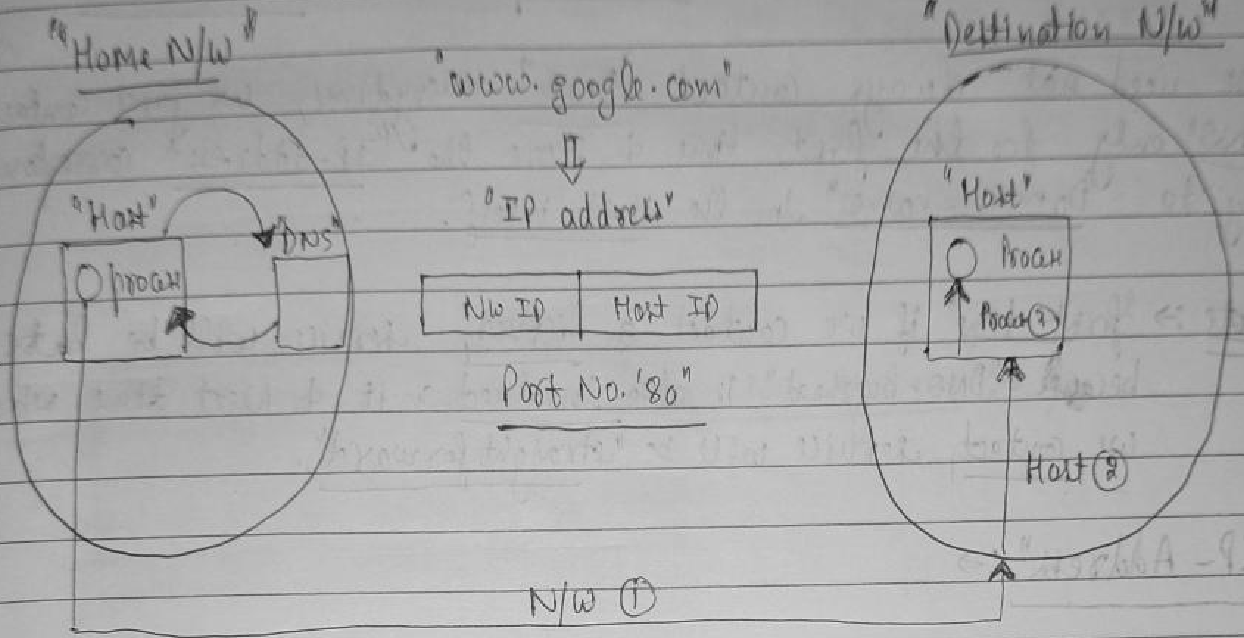
∴ using "Network-ID" we can reach the "target Network"

&

Using "Host-ID" we can reach the "target-Host"

& Now in order to reach the "process", we need something called as "Port Numbers".

Since here in this example we are talking about "Web-services". All the webservices run at port number (80) (http services)



In order to reach the process, we need "process-no." which is actually the port no., & most of the "port nos" are already predefined and they are fixed.

FTP → 21
SMTP → 25
HTTP → 80

Port nos. of these "services/protocols".

If we know the "IP-address" we can directly reach the destination but since these nos are difficult to remember. So, "ISP" (Internet service-Provider) is supposed to provide us with a service known as "Domain-Name-service".

It means inside the "Network" we have configured a server called "Domain Name server". Before contacting the destination N/w directly we need to contact "DNS", to provide "IP address" of ["google.com"]

Now This "entire procedure" of converting the "Domain-Name" into "IP-address" is basically an overhead.

& This entire step is called "DNS-overhead"

We need not always contact "DNS" everytime, we just contact "DNS" only for the first time & store the "IP-address" corresponding to "Domain-name" in the Host "Itself".

Note :⇒ "first time" if we contact a "server", service will be late because "DNS-overhead" is also involved in it. & Next time when we contact, service will be "straight forward".

"IP-Address" :⇒

1> "Unary No. system" :⇒ only one digit "0"

1 ⟶ 0 Base is '1'
2 ⟶ 0 0
3 ⟶ 0 0 0
|
|
so-on

2> "Decimal No systems" :⇒ "10" digits "0 to 9" Base is "10"

3> "Binary No system" :⇒ In this base is "2" & we have only two digits "0" & "1"

Note :⇒ If the base of a no. system is 'n', then it means that

We have $\boxed{n}$ symbols in the "Number - system" & all the symbols will start from
↳ {0, 1, 2, 3, - - - - n-1}

4> "Octal No. system" :-> "Base is 8" 8 digits "0 - 7"

5> "Hexadecimal No. system" :-> "Base is 16" "16 nos are there

"0 to 9" & "A to F"
↓ ↓
"10" "15"

Coming to "Computers", we have "Binary No. system" :->

"Important Nos to Remember" :->

$2^1 \rightarrow 2$
$2^2 \rightarrow 4$
$2^3 \rightarrow 8$
|
|
$2^{10} \rightarrow 1024$ (K)
$2^{20} \rightarrow$ (M)
$2^{30} \rightarrow$ (G)
$2^{40} \rightarrow$ (T)

Let us say we have "One bit" Now with '1' bit how many nos are possible.

~~with "1 bit" we have two nos. "0 & 1"~~.

with "2 bits" we have four nos possible

"1 bit"	"2 bits"	"3-bits"
0 → 0	00 → 0	000 → 0
1 → 1	01 → 1	001 → 1
	10 → 2	010 → 2
	11 → 3	011 → 3
		100 → 4
		101 → 5
		110 → 6
		111 → 7

[Address space or No. space]

Now if we choose '1' bit (one bit). How many parts will I be dividing the entire space into

It turns out that we can divide the "entire space" into "two parts" if we choose "one bit".

One part contains all $\boxed{\text{"zeroes"}}$ in the selected bit & second part contains all $\boxed{\text{"ones"}}$ in the selected bit as shown.

"selected bit" ↓ _

Two address space

0 0	"one part"	all 0's
0 1		
1 0	"second part"	all 1's
1 1		

selected bit

"Three add. space"

0 0 0	All "zeroes" in "first part"
0 0 1	
0 1 0	
0 1 1	
1 0 0	All "ones" in "second part".
1 0 1	
1 1 0	
1 1 1	

Now if we choose "two bits" then we can divide the entire address space into "four parts".

- "One" part will have $\boxed{00}$ in selected bits.
- "2nd" part will have $\boxed{01}$ in selected bits.
- "3rd" part will have $\boxed{10}$ in " "
- "4th" part will have $\boxed{11}$ in selected bits.

"3 bits Add. space" when "2 bits" are selected.

"Selected bits"

0	0	0	"first-part"	00
0	0	1		
0	1	0	"second-part"	01
0	1	1		
1	0	0	"Third-part"	10
1	0	1		
1	1	0	"fourth-part"	11
1	1	1		

- 1 bit $\rightarrow 2^1$ parts
- 2 bits $\rightarrow 2^2 = 4$ parts
- 3 bits $\rightarrow 2^3 = 8$ part
- n bits $\rightarrow 2^n$ parts

Note :=> If we have an "n-bit number" then total numbers placed in "Address space" of this "n-bit number" is (2^n).

Now if we choose $\boxed{\text{"k" bits}}$ out of $\boxed{\text{'n' bits.}}$ then we are basically dividing the entire address space into $\boxed{2^k}$ parts

P.T.O.

2^n nos. in total are present in the "address space",

K bits $\rightarrow$ $\boxed{2^K \text{ parts}}$ in "address space."

Now let us see the size of each part:

∴ "2^K parts" $\longrightarrow$ "2^n numbers"

1 part $\longrightarrow$ $\dfrac{2^n}{2^k}$

$= \left(2^{n-k}\right)$ ~~size~~ "size of each part".

Now Coming to "Computer Networks"

the $\boxed{\text{'n'}}$ bit is nothing but the no. of bits in "IP-address"

← "n bits" →

"K bits"	"n-k" bits

Now "K-bits" is nothing but the "Network-Id-part".

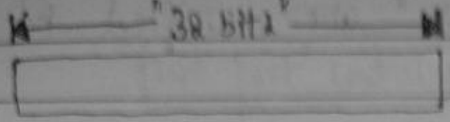
&

"n-k" bits is nothing but the "Host-Id-part".

So, if we choose $\boxed{\text{'K bits}}$ we will be getting "2^K-Networks".

& size of each network is $\left(2^{n-k}\right)$.

Coming to "Computer Networks": Size of "IP-address" is "32-bits".



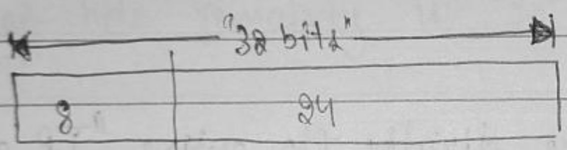
"IP-address" : with these "32 bits", No. of "IP addresses possible"

are $[2^{32}$ "IP-addresses"$]$

Initially in times of "ARPANET", The "IP-address" (32-bit IP address) is divided into "static parts".

→ "8-bits" of "Network-ID".

→ "24-bits" of "Host-ID".



∴ "No. of Networks" = "2^8 Networks" = "256 Networks"

& "Size of each Network" = "2^{24}" = "16 million Hosts".

Initially It want "256 Networks".

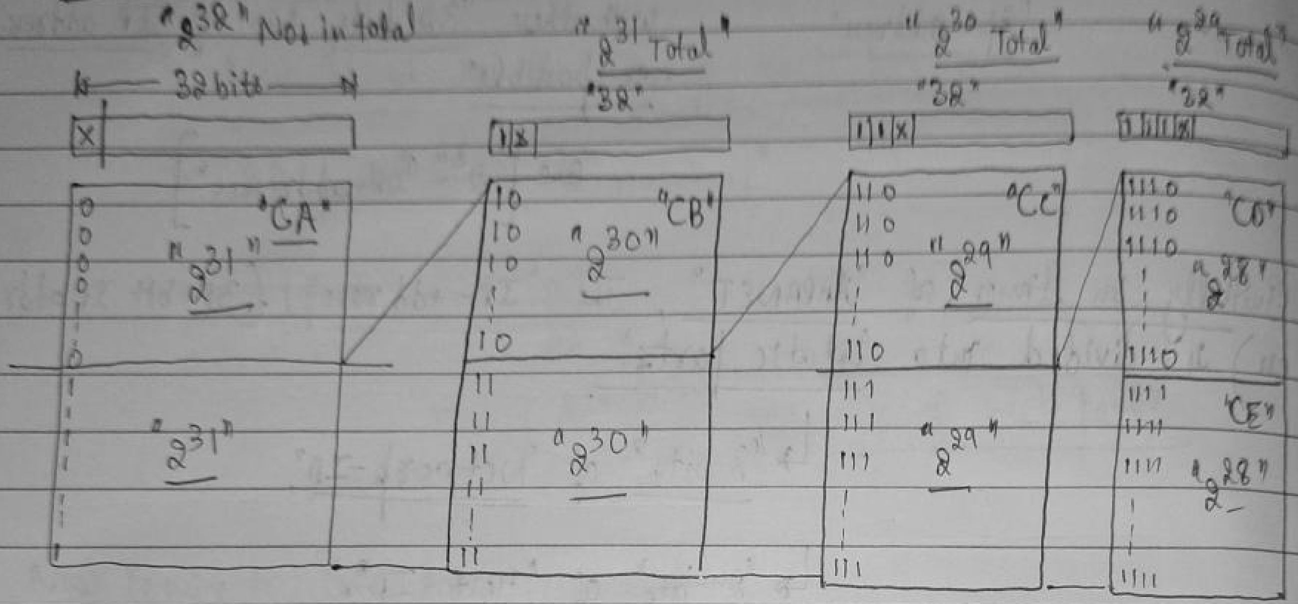
∴ This division is clearly not "scalable".

∴ we didnot use this at all & Moreover if we buy this netw (one network), we are going to get "16-Million" "IP-Addresses". & that is a very "big number".

P.T.

"CLASSFULL IP - ADDRESSING" :-

Now let us see how the "IP-Addresses" are classified in "Class-full manner."



Now if we choose "One bit" we are dividing the entire "IP addresses" into two parts.

"One part" will contain $\boxed{'0'}$ in the "first-bit".

&

"Second part" will contain $\boxed{'1'}$ in the "first-bit"

Now size of "first part" is → "2^{31}"

size of "second part" is → "2^{31}"

Now "first part" is "left out" like that & Now they have chosen the "second-part"

↳ Total "No. of Numbers" present in the second part is $\boxed{2^{31}}$, & our "IP add" is of "32-bits" out of which "first-bit" is already chosen & that bit is set as $\boxed{'1'}$ & In the remaining "2^{31}" part

If we choose the "second-bit" then again we will divide the "entire address-space" into "two parts".

"first part" containg :-> $\boxed{"10"}$ in the "leading-bits"

&

"second part" will contain $\boxed{"11"}$ in the "leading bits"

Now size of each part here is $\boxed{2^{30}}$.

Now leave the "first part" as it is & again divide the "second part".

Note:-> size of "IP-Address" doesn't change. It remains "32-bits"

Now in second part "first two bits" are already fixed as $\boxed{"11"}$
Now no. of bits remaining is "30" ∴ "total size" of this entire space is $\boxed{2^{30}}$

Now in this $\boxed{2^{30}}$, we will choose the "third-bit" as first "two bits" are "already fixed". Now becoz of this "third-bit" the entire space again gets divided into two parts, with

"first part containing" :-> $\boxed{"110"}$ as "leading-bits"

&

"second part containing" :-> $\boxed{"111"}$ as "leading-bits".

& size of each part is "2^{29}"

Now, again we will use "2nd part". Now here Total No. present is 2^{29}. Now ~~out~~ of "32-bits" first three bits are already fixed.
Now we will choose "4th bit" & we will divide again the entire

number space into "two halves."

first part containing :> "1110" as "leading bits"

&

Second part containing :> "1111" as "leading bits"

So, this is how they have classified the entire "IP space" into "different parts."

"First part which starts with zero" is called $\boxed{\text{Class "A"}}$

"Second part which starts with 10" is called $\boxed{\text{Class "B"}}$

"Third part which starts with 110" is called $\boxed{\text{Class "C"}}$

"Fourth part which starts with 1110" is called $\boxed{\text{Class "D"}}$

"Fifth part which starts with 1111" is called $\boxed{\text{Class "E"}}$

Now. in "Class A" Total No of "IP-Addresses" present is $\boxed{2^{31}}$.

"Class B" Total No of "IP-Addresses" present is $\boxed{2^{30}}$.

"Class C" Total No of "IP-Addresses" present" is $\boxed{2^{29}}$

"Class D" Total No of "IP-Addresses" present is $\boxed{2^{28}}$

"Class E" Total No of "IP-Addresses" present is $\boxed{2^{28}}$

•—•

Now let us discuss about all the "classes" "one by one" :->

1> "CLASS-A" :->

First let us discuss how to represent ["IP-addresses"]

"IP" address" is "32-bits" in length.

It can be represented in three ways :->

1> Represent it as a "32-bit" number consisting of $\boxed{0's}$ & $\boxed{1's}$.

2> Second way we can convert the "32-bit" binary no. into decimal. (This would be a hectic exercise)

3> Third way is to divide the entire "IP-address" of "32-bits" into four parts of "8-bits each" & convert these "8-bits" into "decimal numbers"

← "32-bits" →			
8 bits	8 bits	8 bits	8 bits

Now each of these "octets" are converted into "decimal Nos" separated by a "." ~~colon~~ , dot.

This is known as "dotted-decimal-representation". which is the "most popular representation".

Note :-> Given an ("IP-address") in binary, if it starts with $\boxed{0}$ it is of "class-A". If it starts with $\boxed{\text{"10"}}$ It belongs to "class-B" If it starts with $\boxed{\text{"110"}}$ belongs to "class-C" & so-on.

P.T.O

But if the "IP-address" is given in "dotted decimal representati-on".

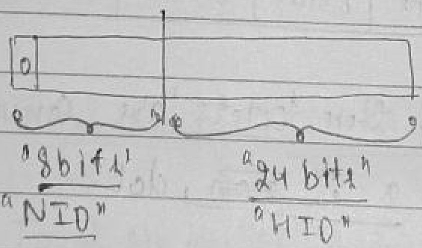
How to find out what is the class, to which the "IP-address" belongs to?

Take the "first octet" convert it to "binary" & examine the "leading bits" to find out the "class exactly".

Note :→ These are also called as "prefix codes"

"Prefix Code"	"Class"
'0'	"A"
'10'	'B'
"110"	'C'
'1110'	'D'
"1111"	'E'

1> "CLASS A" :→



"NID part" → "8 bits"

"HID part" → "24 bits"

out of these "8 bits" of 'NID' "first-bit" is already fixed as '0'

then only $\boxed{7}$ bits are remaining.

So using these "7 bits" No. of Networks present in "Class A" are

"2^7" = "128 Networks" are present in "class-A".

Now If we take any "Network" then No. of "IP-addresses" present In each "Network" is,

↳ "2^{24} = 16 Million."

∴ 16 Million "IP-addresses" are present per network in "class A"

"Range" fixed. ← "8 bits" →

0 _ _ _ _ _ _ _ N/w

0 0 0 0 0 0 0 → "0"

0 0 0 0 0 0 1 → "1"

0 0 0 0 0 1 0 → "2"

⋮

1 1 1 1 1 1 1 → "127"

∴ No. of "N/w's" in "class A" = 128
0 to 127

- But we are not going to use Network No. '0' & "127" practically.

∴ Number of Networks present in "Class A" = "126".
Practically

∴ Range of "Class A" = "1 to 126"

Each Network contains/consists of 2^{24} = "16 Millions" Hosts.

Note :→ Organizations like "NASA" & "Pentagon" uses these Networks.

2> "CLASS B" :→ "2^{30} IP Addresses" are present in "CLASS B"

P.T.O

"32 bits"

10	
"16 bits" "N ID"	"16 bits" "HID"

"first two bits are fixed"

Out of "16 bits" 2 bits are already fixed.

$\therefore$ No. of Networks in 'Class B' = $\boxed{2^{14} \text{ N/w's}}$

$[2^{14} \cong 16K$ Networks$]$

"No. of Hosts" present per 'Network' :→ (2^{16} Hosts / Network)

No. of IP-addresses in "Class B" = "2^{30}"

"$2^{14} * 2^{16}$" = 2^{30}

"No. of Networks" "No. of Hosts per Network"

Now let us see the range of "Class B"

"Just look at first octet".

"8 bits"

10 _ _ _ _ _ _

000000 → 128

000001 → 129

000010 → 130

⋮

111111 → 191

$\therefore$ Range is (128 — 191)

$(111 \rightarrow 2^3 - 1 = 7)$ $(n\ 1's \rightarrow 2^n - 1)$

Using "first octet" we get "64 No's" & there is second octet also which contains 8 bits

$\therefore$ No. of N/w's $= 64 * 2^8$

$= 2^{14}$ Networks

These "64 No's" (128 to 191) are capable of generating "2^{14} Networks".

"128" can be written as :- 128.0, 128.1, 128.2, ..., 128.255 — Each number is capable of generating "256 N/w's"

$\therefore$ "Every Number" is capable of generating "256 numbers or N/w's".

$\therefore$ Total = "64 x 256 N/w's"

128.0
128.1
128.2
128.3
128.4
...
128.255
129.0
129.1
...
129.255
...
191.0
191.1
...
191.255

"Networks in Class B"

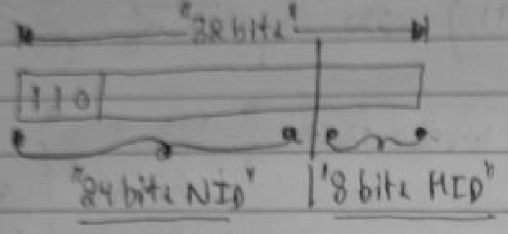
$\therefore$ We have [2^{14} N/w's] & Each Network consists of [2^{16} Hosts].

"Range of Class B" = (128 — 191)

"P.T.O"

⇒ "CLASS - C" :⇒ Total No. of "IP-addresses" present in "Class-C" are (2^{29}).

Now we need to divide this "2^{29} IP addresses" into "Networks & Hosts".



Now out of these "24 bits" in "N/ID part" "3 bits" are already fixed. ∴ Remaining bits are = 24 - 3 = '21 bits'.

∴ No. of N/w's in "Class C" = $[2^{21}$ Networks$]$

$2^{21} \simeq$ '2 Million"

& No. of "IP addresses" per network in "Class C" is 2^8

= "256 Hosts/NW"

Note :⇒ If we buy a Network of "class C" It consists of "256 IP-addresses".

These kind of "Networks" are used by "Small-organizations" / small colleges etc.

Let us now see the 'range of class "C" :⇒

|← 8 bits →|

110 _ _ _ _ _

0 0 0 0 0 → 192

0 0 0 0 1 → 193 $\therefore$ Range is (192 – 223)

0 0 0 1 0 → 194 "32"

⋮

1 1 1 1 1 → 223

Using first "eight bits" we get "32 Numbers" & still there are "16 bits" remaining.

$\therefore$ we get $\boxed{32 \times 2^{16}}$ = "2^{21} Networks"

Now "IP addresses of N/ws" are like ⇒

192.0.0
192.0.1
⋮
192.0.255
192.1.0
192.1.1
⋮
192.255.255
& so on

$\therefore$ One number actually represents 2^{16} "N/ws"

$\therefore \left[32 \times 2^{16} = 2^{21} \text{ N/ws}\right]$ in total in 'Class C'

4> "CLASS-D" & "CLASS-E" :⇒ Now coming to "Class D" & "Class E", IP address is not divided into "Network-Id" & "Host-Id" part.

The entire "IP-address" is left as such without dividing it into "N-ID" & "H-ID" parts.

P.T.O

Now, No. of "IP-addresses" present in "class D" = $\boxed{2^{28}}$.

Now Range of "class D"

|←— '8 bits' —→|

1110 _ _ _ _

0 0 0 0 → "224"

0 0 0 1 → "225"

0 0 1 0 → "226"

⋮

1 1 1 1 → "239"

"Range" = ("224 — 239")

Now Range of 'Class E"

|←— '8 bits' —→|

1111 _ _ _ _

0 0 0 0 → "240"

0 0 0 1 → "241"

0 0 1 0 → "242"

⋮

1 1 1 1 → "255"

Range = ("240 — 255")

These two Classes

↳ "Class D" & "Class E" are not used for practical purposes for assigning IP-addresses.

"CLASS D" is used for "Multicasting".

&

"CLASS E" is used for 'Reserved purposes"

↳ [generally used for "military applications"]

"CLASS A" → 1 to 126 | "CLASS C" → 192-223 | "CLASS E" → 240-255
"CLASS B" → 128 to 191 | "CLASS D" → 224-239

Note:→ Whenever we buy a "Network of any class" {A to C} we are not going to use two "IP-addresses" within this network

↳ The first "IP-address".

↳ The last "IP-address"

Note:→ Always the number of "Hosts" that can be configured is two less than the "total no. of IP addresses present".

"Class"	"Total IP address"	"No. of Network"	"No. of IPs per N/w"	"No. of Hosts per N/w"
"Class A"	2^{31}	126	2^{24}	$2^{24}-2$
"Class B"	2^{30}	2^{14}	2^{16}	$2^{16}-2$
"Class C"	2^{29}	2^{21}	2^{8}	$2^{8}-2$
"Class D"	2^{28}	—	—	—
"Class E"	2^{28}	—	—	—

"TYPES of CASTING → "Unicast", "Limited BroadCast", "Directed Broad-Cast"."

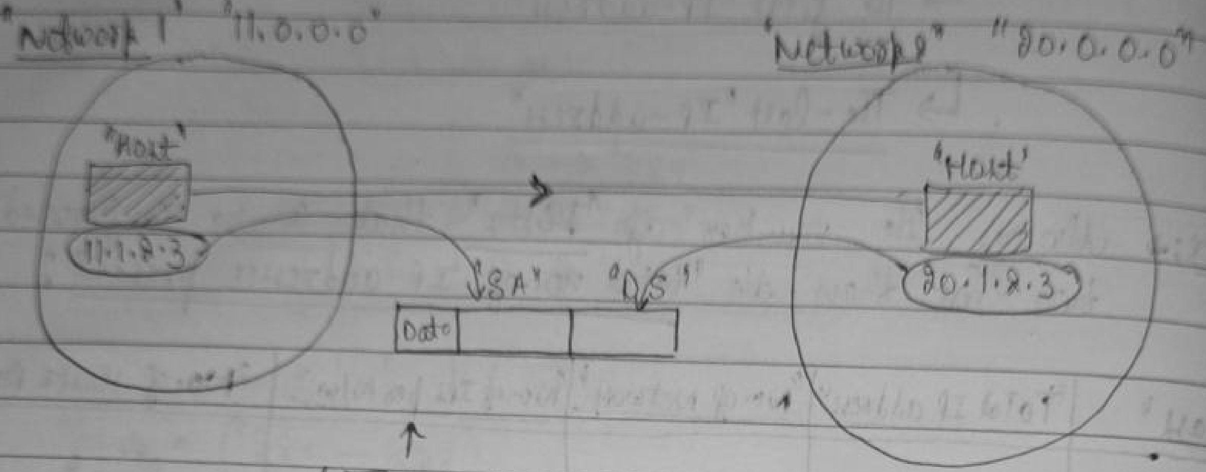
Let us now see how to send "packets" from "One Host" to "another Host"

There are two ways of doing so. This is also known as "Casting".

"Casting" is of two types":→

1> "Unicast":→ "Sending a packet from one host to only one particular host" "first IP-add. is reserved for Network IP address"

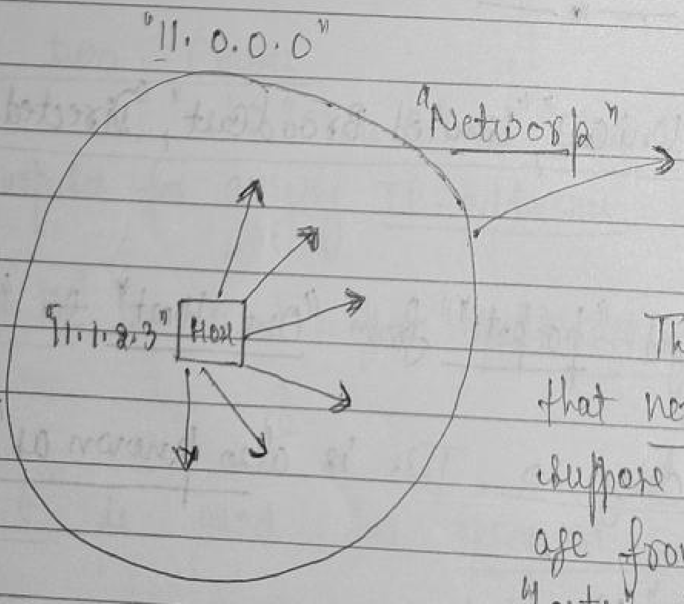
3> **Broadcast :-** Sending a "packet" from one host to many hosts is called broadcasting.

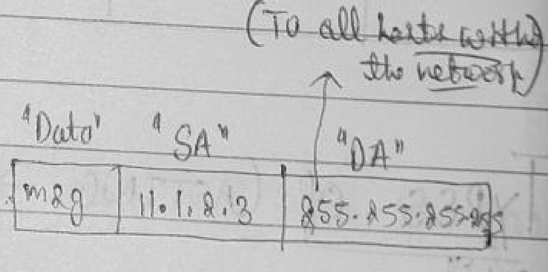


"This is known as Unicasting"
"Host-to-Host"

"Broadcasting"

- "Limited-Broadcasting"
- "Directed-Broadcasting"





There is a "Network" & within that network, there is a "Host". Now suppose we want to send a message from this host to all other "hosts" in the network. It is called as "Limited broadcasting"

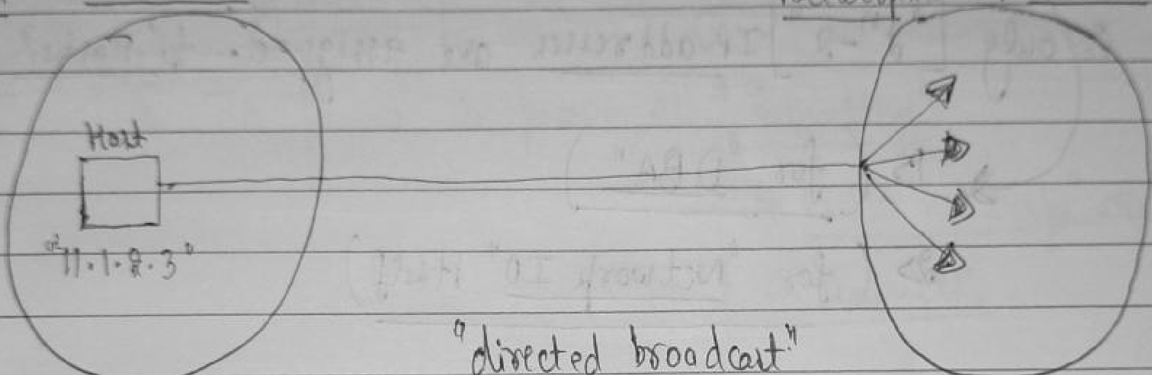
* No. "255.255.255.255" cannot be assigned to any "host", bcoz it is reserved for "Limited broadcasting".

If "DA" is "255.255.255.255" then such a packet will be received by all other hosts within the network (same network)

1) "Limited-broadcasting" :- Sending a message from "one host" to all other hosts within the same network is called 'Limited Broadcasting'.

2) "Directed-broadcasting" :- A ["host"] in a network wants to send a message to all other hosts in some other "network".

"Network 1" "11.0.0.0" "Network 2" "20.0.0.0"



"directed broadcast"

Msg	"SA"	"DA"
'data'	11.1.2.3	20.255.255.255

(20.255.255.255)

↓

if we use this "add." as "Destination address" then "packet" will be delivered to all the hosts in network ["20"].

∴ Note :- we will never assign an "IP add" which contains all [1's] in the "HOST-ID" part.

["LBA" :- 255.255.255.255]

["DBA" :- (NID), HID having all one's]

↓

"valid N/w id"

"N/w ID" In "N/w part" we have a valid "N/w" & in "Host ID" part we will have all 1's.

↓ This cannot be assigned to any "Host" as it is the "IP of the network".

∴ If we have "2^n" IP addresses per Network we are not going to use '2' out of these "2^n IP addresses"

∴ only ["$2^n - 2$"] IP addresses are assigned to "Hosts"

1) (for "DBA")

2) (for "Network ID" itself)

CA → $\boxed{1-126}$, CB → $\boxed{128-191}$, CC → $\boxed{192-223}$

"IP"	"NID"	"DBA"	"LBA"
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	255.255.255.255
130.1.2.3	130.1.0.0	130.1.255.255	"
150.0.150.150	150.0.0.0	150.0.255.255	"
200.1.10.100	200.1.10.0	200.1.10.255	"
220.15.1.10	220.15.1.0	220.15.1.255	"
250.0.1.2	X	X	X
300.1.2.3	X	X	X