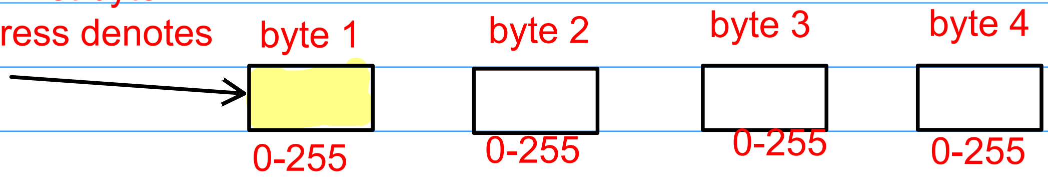
firat octet or first byte
of IPv4 address denotes
the CLASS



2^32
2^0
..
..
..
..
2^31

dotted
decimal
notation

minimum IP : 0.0.0.0

Maximum IP : 255.255.255.255

Binary
notation

Minimum IP : 00000000 00000000 00000000 00000000

Maximum IP : 11111111 11111111 11111111 11111111

2^8 = 0-255
= 256

class A	0 - 127	0.0.0.0	to	127.255.255.255
class B	128 - 191	128.0.0.0	to	191.255.255.255
class C	192 - 223	192.0.0.0	to	223.255.255.255
class D	224 - 239	224.0.0.0	to	239.255.255.255
class E	240 - 255	240.0.0.0	to	255.255.255.255

120.34.56.78	valid	class A
246.23.45.89	valid	class E

class A 0 - 127

class B 128 - 191

class C 192 - 223

class D 224 - 239

class E 240 - 255

class A 0 - 127

0	0000 0000
1	0000 0001
2	
3	
...	
..	
127	0111 1111

class B 128 - 191

128	1000 0000
129	1000 0001
...	
...	
..	
..	
191	1011 1111

class C 192 - 223

192	1100 0000
193	1100 0001
...	
...	
...	
..	
223	1101 1111

class D 224 - 239

224	1110 0000
225	1110 0001
...	
...	
...	
...	
239	1110 1111

class E 240 - 255

240	1111 0000
241	1111 0001
...	
...	
..	
...	
255	1111 1111

class A   0 -127

0.0.0.0

10.0.0.0   10.255.255.255

127.255.255.255

class B   128 - 191

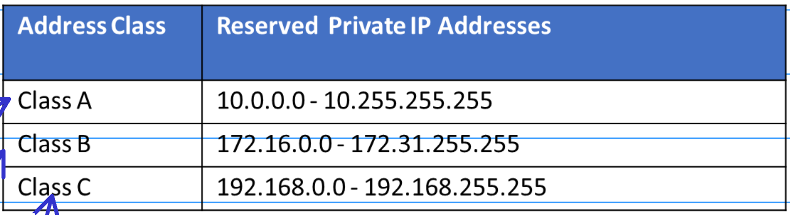
128.0.0.0

172.16.0.0   172.31.255.255

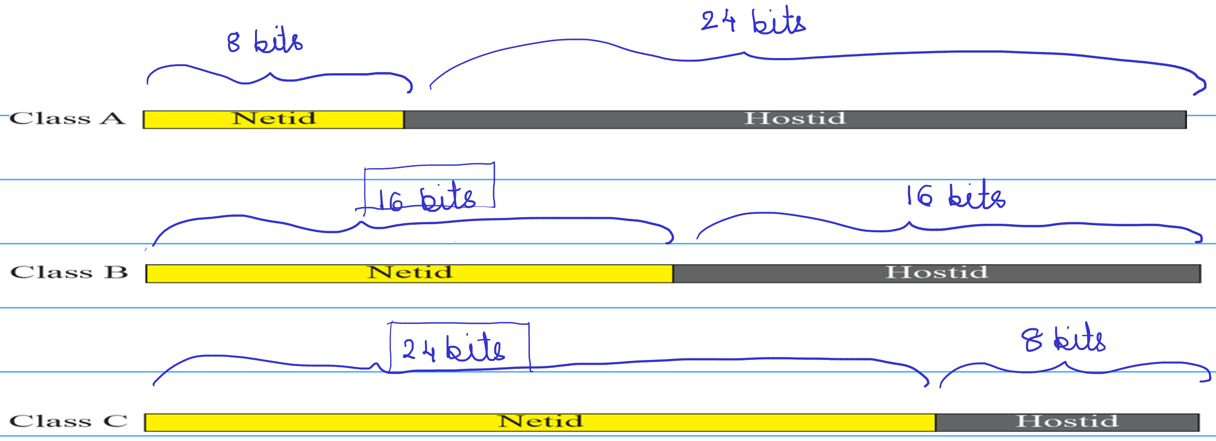
191.255.255.255

class C   192 - 223

192. 168.0.0   192.168.255.255



Address Class	Reserved Private IP Addresses
Class A	10.0.0.0 - 10.255.255.255
Class B	172.16.0.0 - 172.31.255.255
Class C	192.168.0.0 - 192.168.255.255



http request response model --

- the request sent from client to server
- server will process that request and send it back to client
- this request response model is followed in http protocol

request methods -

GET, - used to get web page from server
POST, - used to submit html form data to server
HEAD - tells server jsut sent response header dont send response body
PUT, - upload a page/document on server
DELETE - delete a doc/resource from server
TRACE - to trace http request
OPTIONS - tells you which request methods are supported by some url on server
CONNECT - some more request methods that establish 2 way communication channel with server which is used with proxy as well
PATCH. - updating partial data on server

Request method is not the only thing there also we have request body

request body - have data in different forms
like json format, xml format,
form data in key-value pair separated by ampersand (&) sign.

length - - this particular data is of how many bytes is given in length header

- data type is given in content type

it can also contain something like cookies
(which is one of state management mechanism)

request header also includes - client header information like IP address of client, or browser agent or some more details

this request is then send to the server
- the server is going to process the request and going to produce a response

RESPONSE INCLUDES -
again
http version 1.0
1.1

also has status code
which indicates request is handled succesfully or if there is any error

Response status codes:

1) 1xx (information),
in range of 100 indicates information
it tells if any switching of protocol is done in between

2) 2xx (success),
in range of 200 indicates success
200 - most important as it indicates status - success(OK)
201 - tells clients some resource is created on server

3) 3xx (redirection),
in range of 300 indicates redirection
302 - tells client to get resource from other url

4) 4xx (client error),
most commonly seen error
404 - not found error
403 - forbidden error(client trying to access something for which it does not have permission

5) 5xx (server error)
in servlet programming this error is commonly seen as interna server error 500 many a times

Along with status code & test , there some more information
content type is there - whatever response you are sending back to client,
you need to tell in header what type of response
it is?

when you are sending web pages to client which is dynamically filled
from server side processing
we send test/xml,also application json - popularly used in REST services

we have images ,jpg,png to send - also content types can be audio,videos or songs

#Traditionally for each request client opens new new socket connection to server

#send the request , request is processed on server

#the response is given back to client and the socket connection gets closed.

#for next request again a new connection is open through which request is sent,
processed at server and response is sent back at client and connection is closed.

so http is connectionless protocol

- inspite of being built on top of tcp protocol which is connection oriented

because of which each request is brand new request for web server

eg: just imagine that you just now logged in to a website

that was your first request.
after login you click on a link there
thats an entirely a new request for the server
again you logged in and you filled that form and submitted that form .

the server is now not able to recognize whether requests are comign from
same client or from different different client

of course if this is how a web applcation running then its going to be miserable

so you have to maitain information about client in order to give
better user experience
this user info/client info can be stored wither on server side or client side

if you store this info on server side its referred as server side state management
eg: session,application,etc

if you store client info on client side itself,then its referred as
client side state management
eg: cookies,hiddenfields, session storage,etc

Both has pros and cons because -
whatever you store on client side is visible to client
client can modify or manipulate that
and you putting everything on server side is more dangerous beacuse then
too much load is kept on server side

Java servlets will maintain this state management objects .
These objects are used into whatever learnt in http protocol for web programming.