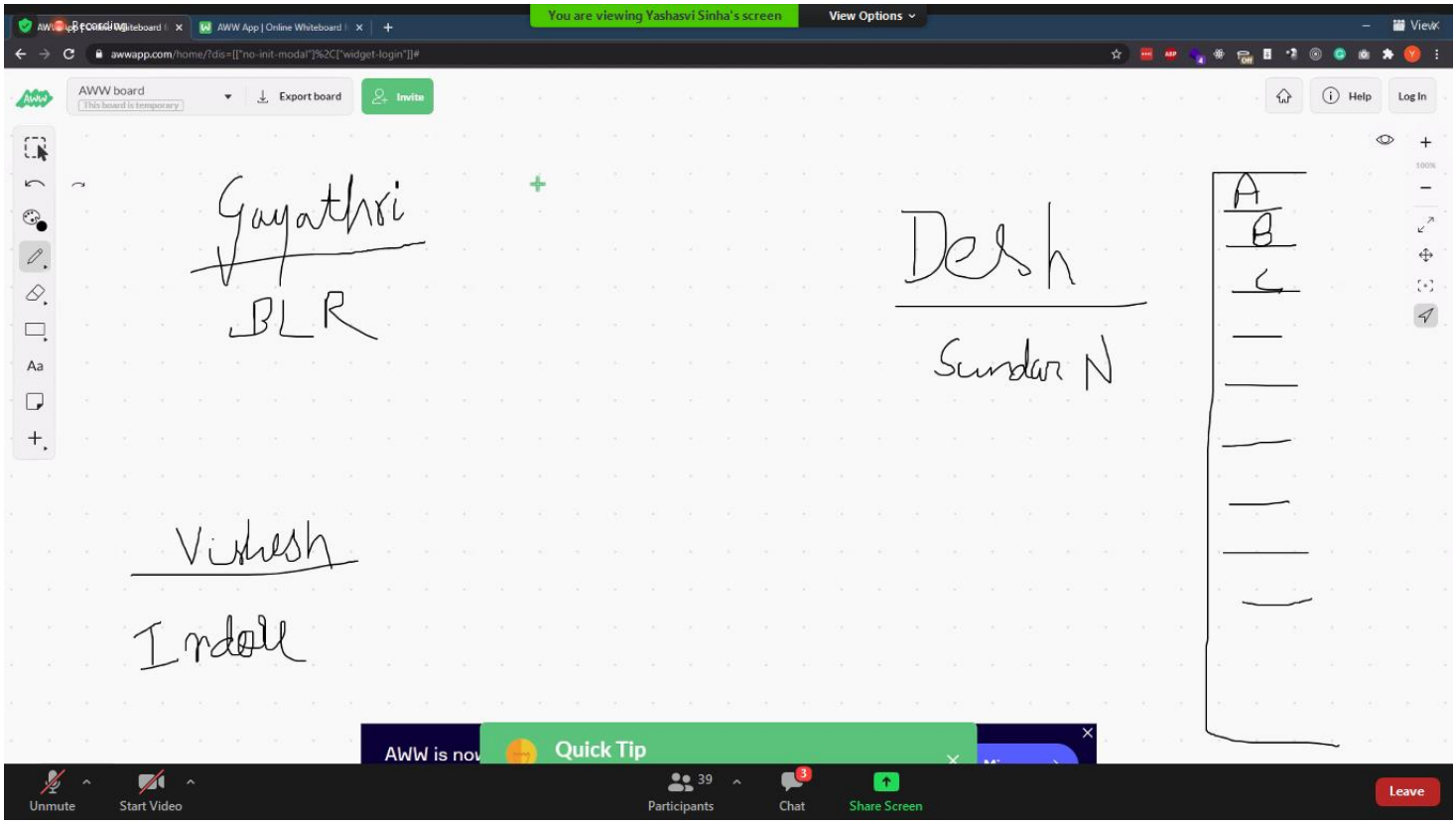
Client Server Model



Desh Raj has an abundance of books. Now, Gayathri wants a book A from Desh Raj. So, Gayathri will send a letter informing Desh Raj that she is wants book A and she posts it. The post office will check the address and deliver it to Desh Raj in Sundar Nagar.

But Vishesh is seeking book B, request the same from Desh Raj. Based on what is written in the letter, Desh Raj will fetch the book ‘A’, create a courier and will send it back again to the post office with Gayathri’s address on the package. The post office will figure out the path to be chosen to delivery the package to Gayathri.

The idea here is, this is the exact similar way, on how our client server model works.

Client Server Model:

The Client-server model is a distributed application structure that partitions task or workload between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and deliver the data packets requested back to the client.

Clients do not share any of their resources.

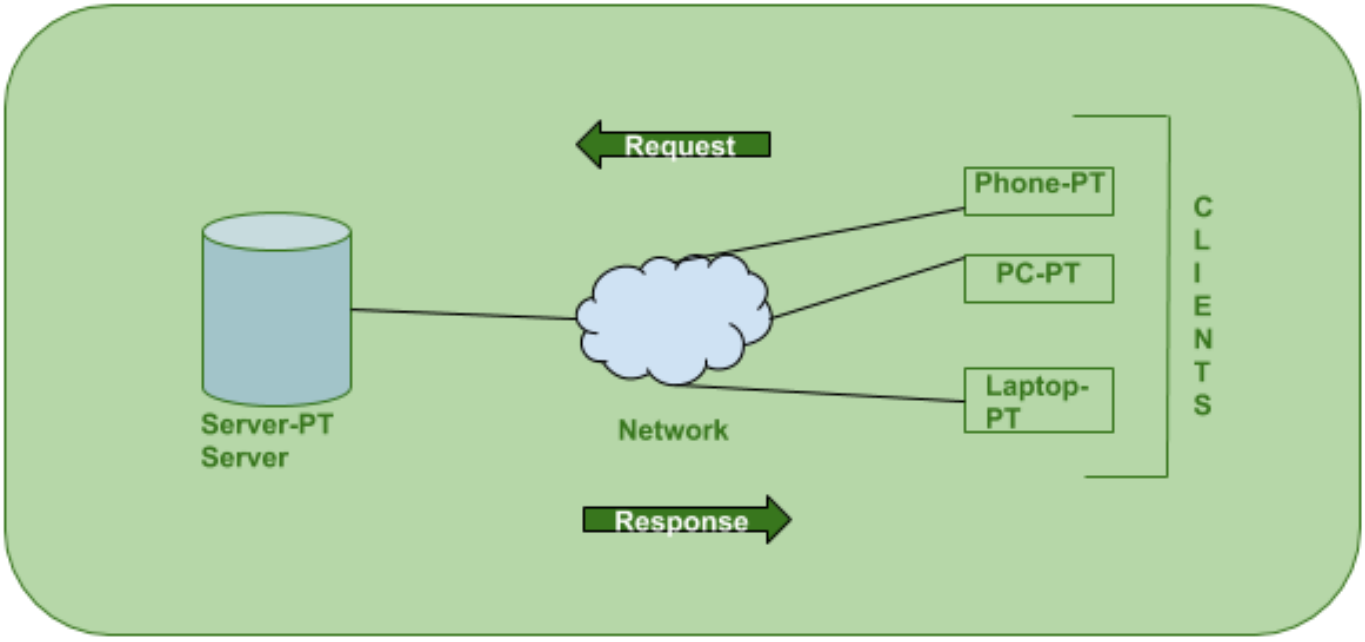
Examples of Client-Server Model are Email, World Wide Web, etc.

How the Client-Server Model works?

Client: When we talk the word **Client**, it means to talk of a person or an organization using a particular service. Similarly in the digital world a **Client** is a **computer** (**Host**) i.e., capable of receiving information or using a particular service from the service providers (**Servers**).

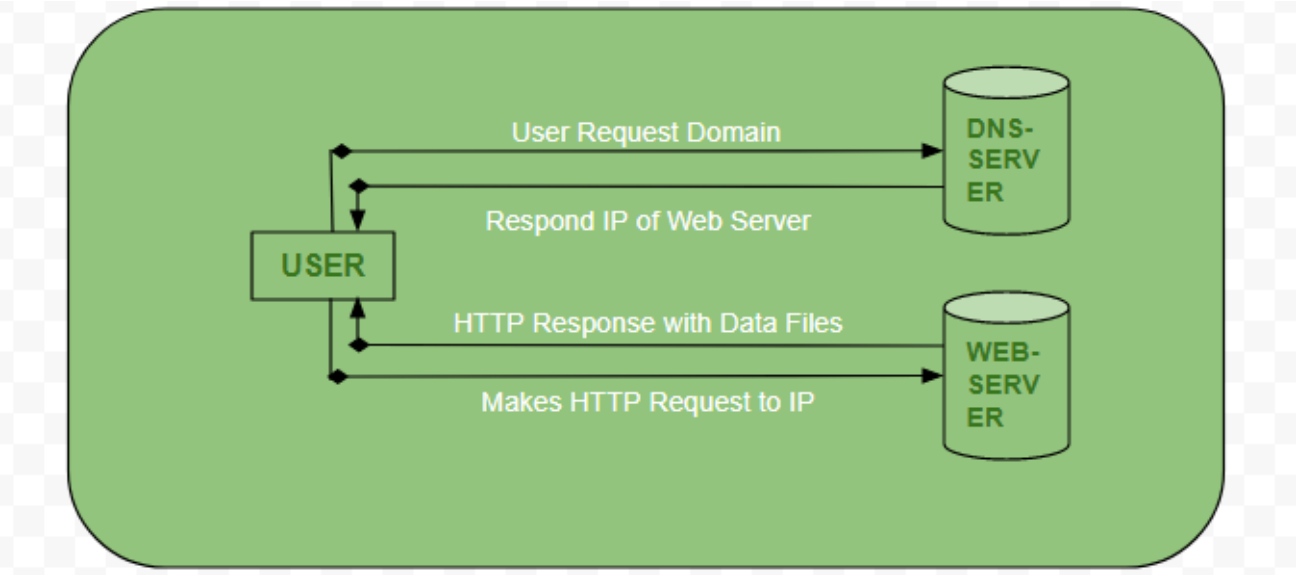
Servers: Similarly, when we talk the word **Servers**, it means a person or medium that serves something. Similarly in this digital world a **Server** is a remote computer which provides information (data) or access to particular services.

So, it's basically the **Client** requesting something and the **Server** serving it as long as its present in the database.



How the browser interacts with the servers?
There are few steps to follow to interacts with the servers a client.

- User enters the **URL** (Uniform Resource Locator) of the website or file. The Browser then requests the **DNS** (DOMAIN NAME SYSTEM) Server.
- **DNS Server** lookup for the address of the **WEB Server**.
- **DNS Server** responds with the **IP address** of the **WEB Server**.
- Browser sends over an **HTTP/HTTPS** request to **WEB Server's IP** (provided by **DNS server**).
- Server sends over the necessary files of the website.
- Browser then renders the files and the website is displayed. This rendering is done with the help of **DOM** (Document Object Model) interpreter, **CSS** interpreter and **JS Engine** collectively known as the **JIT** or (Just in Time) Compilers.



Advantages of Client-Server model:

- Centralized system with all data in a single place.
- Cost efficient requires less maintenance cost and Data recovery is possible.
- The capacity of the Client and Servers can be changed separately.

Disadvantages of Client-Server model:

- Clients are prone to viruses, Trojans and worms if present in the Server or uploaded into the Server.
- Server are prone to Denial of Service (DOS) attacks.
- Data packets may be spoofed or modified during transmission.
- Phishing or capturing login credentials or other useful information of the user are common and MITM (Man in the Middle) attacks are common.

HTTP

- HTTP stands for **HyperText Transfer Protocol.**
- It is a protocol used to access the data on the World Wide Web (www).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- This protocol is known as HyperText Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.
- HTTP is similar to the FTP as it also transfers the files from one host to another host. But, HTTP is simpler than FTP as HTTP uses only one connection, i.e., no control connection to transfer the files.
- HTTP is used to carry the data in the form of MIME-like format.
- HTTP is similar to SMTP as the data is transferred between client and server. The HTTP differs from the SMTP in the way the messages are sent from the client to the server and from server to the client. SMTP messages are stored and forwarded while HTTP messages are delivered immediately.

Features of HTTP:

- **Connectionless protocol**: HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the connection. The connection between client and server exist only during the current request and response time only.

- **Media independent**: HTTP protocol is a media independent as data can be sent as long as both the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME-type header.

- **Stateless**: HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.

HTTP	HTTPS
The full form of HTTP is the Hypertext Transfer Protocol.	The full form of HTTPS is Hypertext Transfer Protocol Secure.
It is written in the address bar as http://.	It is written in the address bar as https://.
The HTTP transmits the data over port number 80.	The HTTPS transmits the data over port number 443.
It is unsecured as the plain text is sent, which can be accessible by the hackers.	It is secure as it sends the encrypted data which hackers cannot understand.
It is mainly used for those websites that provide information like blog writing.	It is a secure protocol, so it is used for those websites that require to transmit the bank account details or credit card numbers.
It is an application layer protocol.	It is a transport layer protocol.
It does not use SSL.	It uses SSL that provides the encryption of the data.
Google does not give the preference to the HTTP websites.	Google gives preferences to the HTTPS as HTTPS websites are secure websites.
The page loading speed is fast.	The page loading speed is slow as compared to HTTP because of the additional feature that it supports, i.e., security.

HTTP is Stateless:

HTTP is stateless: there is no link between two requests being successively carried out on the same connection. This immediately has the prospect of being problematic for users attempting to interact with certain pages coherently, for example, using e-commerce shopping baskets. But while the core of HTTP itself is stateless, HTTP cookies allow the use of stateful sessions. Using header extensibility, HTTP Cookies are added to the workflow, allowing session creation on each HTTP request to share the same context, or the same state.

Most often used stateful part is to remember the user so that the user does not need to login again and again.

Stateless Protocol	Stateful Protocol
Stateless Protocol does not require the server to retain the server information or session details.	Stateful Protocol require server to save the status and session information.
In Stateless Protocol, there is no tight dependency between server and client.	In Stateful protocol, there is tight dependency between server and client
The Stateless protocol design simplify the server design.	The Stateful protocol design makes the design of server very complex and heavy.
Stateless Protocols works better at the time of crash because there is no state that must be restored, a failed server can simply restart after a crash.	Stateful Protocol does not work better at the time of crash because stateful server have to keep the information of the status and session details of the internal states.
Stateless Protocols handle the transaction very fast.	Stateful Protocols handle the transaction very slowly.
Stateless Protocols are easy to implement in Internet.	Stateful protocols are logically heavy to implement in Internet.

What is the URL?

A URL or **Uniform Resource Locator** is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc.

A URL uses a protocol for accessing the resource, which can be HTTP, HTTPS, FTP, etc.

It is mainly referred to as the address of the website, which a user can find in their address bars. An example of an URL is given below:

https://amazon.in/categories/books?name=Yash&email=yash@gmail.com

Syntax of URL

Each HTTP URL follow the syntax of its generic URI. Hence the syntax of the URL is also similar to the syntax of URI. It is given below:

```
scheme:[//authority]path[?query][#fragment]
```

The above URL is made up of the following components:

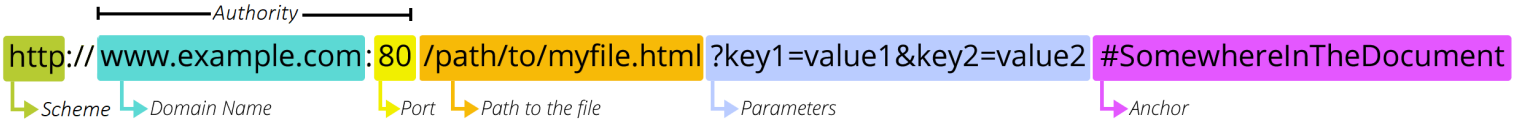
Scheme: The URL's first component is a scheme, which represents a protocol that a browser must need to use to request the resource. The commonly used protocols for websites are HTTP or HTTPS.

Authority: The authority includes two sub-components, **domain name and Port**, separated by a colon. The domain name can be anything, the registered name of the resource like javatpoint.com, and port is the technical gate to access the resource on a webserver. The port number **80 is used for HTTP** and **443 is used for HTTPS**.

Path: The path indicates the complete path to the resource on the webserver. It can be like /software/htp/index.html.

Query String: It is the string that contains the name and value pair. If it is used in a URL, it follows the path component and gives the information. Such as **"?key1=value1&key2=value2"**.

Fragment: It is also an optional component, preceded by a hash(#) symbol. It consists of a fragment identifier that provides direction to a secondary resource.



https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_URL

- Domain Name would be converted to a number 122.4.4.25. The range would be from 0 – 255. Each individual part are actually a 8 bit number. In binary there will be 8 parts.
- 232 = 4,294,967,296, so we can have so many unique values using a 32-bit system.
- So, when we enter a domain name, it gets converted into an IP address.

DNS Server

Domain Name Server (DNS) is a standard protocol that helps Internet users discover websites using human readable addresses. Like a phonebook which lets you look up the name of a person and discover their number, DNS lets you type the address of a website and automatically discover the Internet Protocol (IP) address for that website.

What is DNS Used For?

- Resolving names of World Wide Web (WWW) sites
- Routing messages to email servers and webmail services
- Connecting app servers, databases and middleware within a web application
- Virtual Private Networks (VPN)
- Peer-to-peer sharing programs
- Multiplayer games
- Instant messaging and online meeting services
- Communication between IoT devices, gateways and servers

https://ns1.com/resources/what-is-dns

IPV4 & IPV6

What is IP?

An IP stands for internet protocol. An IP address is assigned to each device connected to a network. Each device uses an IP address for communication. It also behaves as an identifier as this address is used to identify the device on a network. It defines the technical format of the packets. Mainly, both the networks, i.e., IP and TCP, are combined together, so together, they are referred to as a TCP/IP. It creates a virtual connection between the source and the destination.

We can also define an IP address as a numeric address assigned to each device on a network. An IP address is assigned to each device so that the device on a network can be identified uniquely. To facilitate the routing of packets, TCP/IP protocol uses a 32-bit logical address known as IPv4(Internet Protocol version 4).

An IP address consists of two parts, i.e., the first one is a network address, and the other one is a host address.

There are two types of IP addresses:

1. **IPv4**
2. **IPv6**

What is IPv4?

IPv4 is a version 4 of IP. It is a current version and the most commonly used IP address. It is a 32-bit address written in four numbers separated by 'dot', i.e., periods. This address is unique for each device.

For example, **66.94.29.13**

The above example represents the IP address in which each group of numbers separated by periods is called an Octet. Each number in an octet is in the range from 0-255. This address can produce 4,294,967,296 possible unique addresses.

In today's computer network world, computers do not understand the IP addresses in the standard numeric format as the computers understand the numbers in binary form only. The binary number can be either 1 or 0. The IPv4 consists of four sets, and these sets represent the octet. The bits in each octet represent a number.

Each bit in an octet can be either 1 or 0. If the bit the 1, then the number it represents will count, and if the bit is 0, then the number it represents does not count.

Representation of 8 Bit Octet

128	64	32	16	8	4	2	1

The above representation shows the structure of 8- bit octet.

Now, we will see how to obtain the binary representation of the above IP address, i.e., 66.94.29.13

Step 1: First, we find the binary number of 66.

128	64	32	16	8	4	2	1
0	1	0	0	0	0	1	0

To obtain 66, we put 1 under 64 and 2 as the sum of 64 and 2 is equal to 66 (64+2=66), and the remaining bits will be zero, as shown above. Therefore, the binary bit version of 66 is 01000010.

Step 2: Now, we calculate the binary number of 94.

128	64	32	16	8	4	2	1
0	1	0	1	1	1	1	0

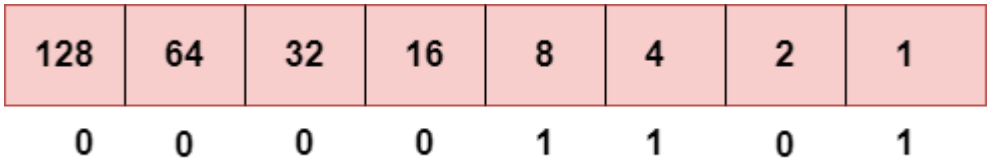
To obtain 94, we put 1 under 64, 16, 8, 4, and 2 as the sum of these numbers is equal to 94, and the remaining bits will be zero. Therefore, the binary bit version of 94 is 01011110.

Step 3: The next number is 29.

128	64	32	16	8	4	2	1
0	0	0	1	1	1	0	0

To obtain 29, we put 1 under 16, 8, 4, and 1 as the sum of these numbers is equal to 29, and the remaining bits will be zero. Therefore, the binary bit version of 29 is 00011101.

Step 4: The last number is 13.



To obtain 13, we put 1 under 8, 4, and 1 as the sum of these numbers is equal to 13, and the remaining bits will be zero. Therefore, the binary bit version of 13 is 00001101.

Drawback of IPv4

Currently, the population of the world is 7.6 billion. Every user is having more than one device connected with the internet, and private companies also rely on the internet. As we know that IPv4 produces 4 billion addresses, which are not enough for each device connected to the internet on a planet. Although the various techniques were invented, such as variable- length mask, network address translation, port address translation, classes, inter-domain translation, to conserve the bandwidth of IP address and slow down the depletion of an IP address. In these techniques, public IP is converted into a private IP due to which the user having public IP can also use the internet. But still, this was not so efficient, so it gave rise to the development of the next generation of IP addresses, i.e., IPv6.

What is IPv6?

IPv4 produces 4 billion addresses, and the developers think that these addresses are enough, but they were wrong. IPv6 is the next generation of IP addresses. The main difference between IPv4 and IPv6 is the address size of IP addresses. The IPv4 is a 32-bit address, whereas IPv6 is a 128-bit hexadecimal address. IPv6 provides a large address space, and it contains a simple header as compared to IPv4.

It provides transition strategies that convert IPv4 into IPv6, and these strategies are as follows:

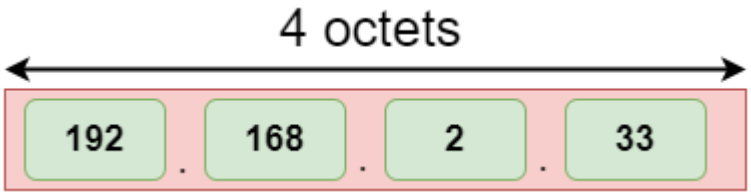
- **Dual stacking**: It allows us to have both the versions, i.e., IPv4 and IPv6, on the same device.
- **Tunneling**: In this approach, all the users have IPv6 communicates with an IPv4 network to reach IPv6.
- **Network Address Translation**: The translation allows the communication between the hosts having a different version of IP.

This hexadecimal address contains both numbers and alphabets. Due to the usage of both the numbers and alphabets, IPv6 is capable of producing over 340 undecillion (3.4*1038) addresses.

IPv6 is a 128-bit hexadecimal address made up of 8 sets of 16 bits each, and these 8 sets are separated by a colon. In IPv6, each hexadecimal character represents 4 bits. So, we need to convert 4 bits to a hexadecimal number at a time

Address format

The address format of IPv4:



The address format of IPv6:

The above diagram shows the address format of IPv4 and IPv6. An IPv4 is a 32-bit decimal address. It contains 4 octets or fields separated by 'dot', and each field is 8-bit in size. The number that each field contains should be in the range of 0-255. Whereas an IPv6 is a 128-bit hexadecimal address. It contains 8 fields separated by a colon, and each field is 16-bit in size.

Differences between IPv4 and IPv6

	Ipv4	Ipv6
Address length	IPv4 is a 32-bit address.	IPv6 is a 128-bit address.
Fields	IPv4 is a numeric address that consists of 4 fields which are separated by dot (.).	IPv6 is an alphanumeric address that consists of 8 fields, which are separated by colon.
Classes	IPv4 has 5 different classes of IP address that includes Class A, Class B, Class C, Class D, and Class E.	IPv6 does not contain classes of IP addresses.
Number of IP address	IPv4 has a limited number of IP addresses.	IPv6 has a large number of IP addresses.
VLSM	It supports VLSM (Virtual Length	It does not support VLSM.

	Subnet Mask). Here, VLSM means that Ipv4 converts IP addresses into a subnet of different sizes.	
Address configuration	It supports manual and DHCP configuration.	It supports manual, DHCP, auto-configuration, and renumbering.
Address space	It generates 4 billion unique addresses	It generates 340 undecillion unique addresses.
End-to-end connection integrity	In IPv4, end-to-end connection integrity is unachievable.	In the case of IPv6, end-to-end connection integrity is achievable.
Security features	In IPv4, security depends on the application. This IP address is not developed in keeping the security feature in mind.	In IPv6, IPSEC is developed for security purposes.
Address representation	In IPv4, the IP address is represented in decimal.	In IPv6, the representation of the IP address in hexadecimal.
Fragmentation	Fragmentation is done by the senders and the forwarding routers.	Fragmentation is done by the senders only.
Packet flow identification	It does not provide any mechanism for packet flow identification.	It uses flow label field in the header for the packet flow identification.
Checksum field	The checksum field is available in IPv4.	The checksum field is not available in IPv6.
Transmission scheme	IPv4 is broadcasting.	On the other hand, IPv6 is multicasting, which provides efficient network operations.
Encryption and Authentication	It does not provide encryption and authentication.	It provides encryption and authentication.
Number of octets	It consists of 4 octets.	It consists of 8 fields, and each field contains 2 octets. Therefore, the total number of octets in IPv6 is 16.