Unit - 8
Application Layer Protocols:

DNS :
- DNS is short for **Domain Name Service** or **Domain Name System**.
- It is an application layer protocol.
- DNS is a host name to IP Address translation service.
- It converts the names we type in our web browser address bar to the IP Address of web servers hosting those sites.
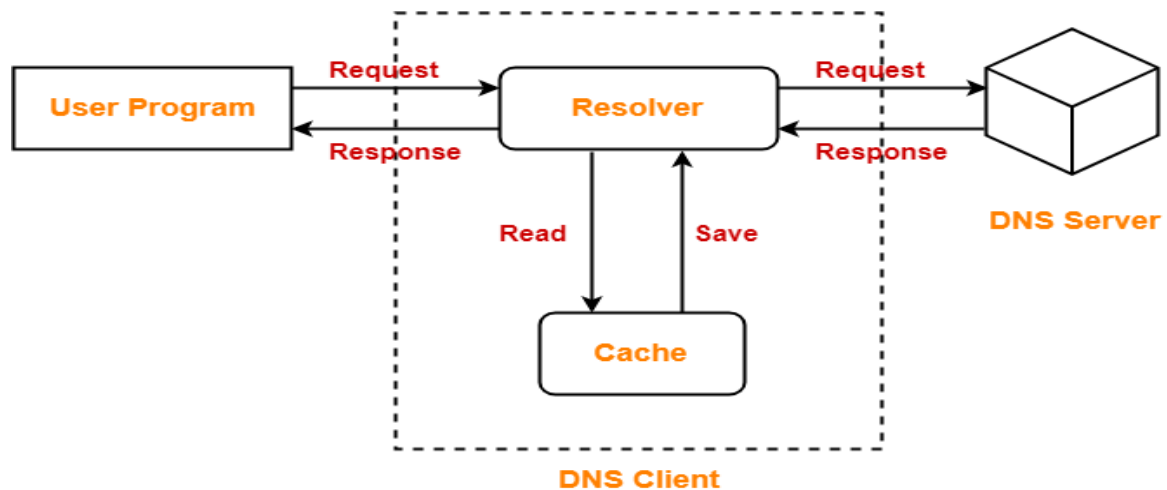

Need-
The need for Domain Name Service arises due to the following reasons-

- IP Addresses are a complex series of numbers.
- So, it is difficult to remember IP Addresses directly while it is easy to remember names.

DNS Resolution-

DNS Resolution is a process of resolving a domain name onto an IP Address.

The following diagram illustrates the process of DNS resolution-

User Program — Request → Resolver — Request → DNS Server
User Program ← Response — Resolver ← Response — DNS Server

Read / Save → Cache

DNS Client

The steps involved in DNS Resolution are-

Step-01:

- A user program sends a name query to a library procedure called the resolver.

Step-02:

Resolver looks up the local domain name cache for a match.

- If a match is found, it sends the corresponding IP Address back.
- If no match is found, it sends a query to the local DNS server.

Step-03:

DNS server looks up the name.

- If a match is found, it returns the corresponding IP Address to the resolver.
- If no match is found, the local DNS server sends a query to a higher level DNS server.
- This process is continued until a result is returned.

Step-04:

- After receiving a response, the DNS client returns the resolution result to the application.

Important Notes-

- DNS uses UDP (port 53) at the transport layer.
- DNS is non-persistent.
- DNS is a stateless protocol.
- Mapping an IP Address onto a domain name is referred to as **Inverse domain**.

Hyper Text Transfer Protocol-

- HTTP is short for **Hyper Text Transfer Protocol**.
- It is an application layer protocol.

- It is **mainly used for the retrieval of data from websites throughout the internet**.
- It works on the top of TCP/IP suite of protocols.

Working-

HTTP uses a client-server model where-

- Web browser is the client.
- Client communicates with the web server hosting the website.
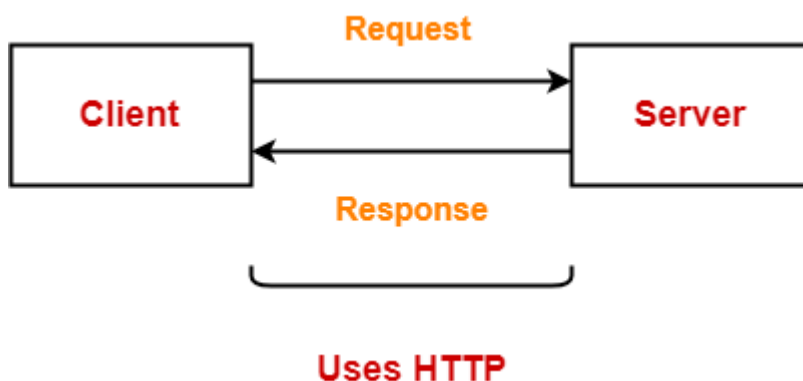
Whenever a client requests some information (say clicks on a hyperlink) to the website server.

The browser sends a request message to the HTTP server for the requested objects.
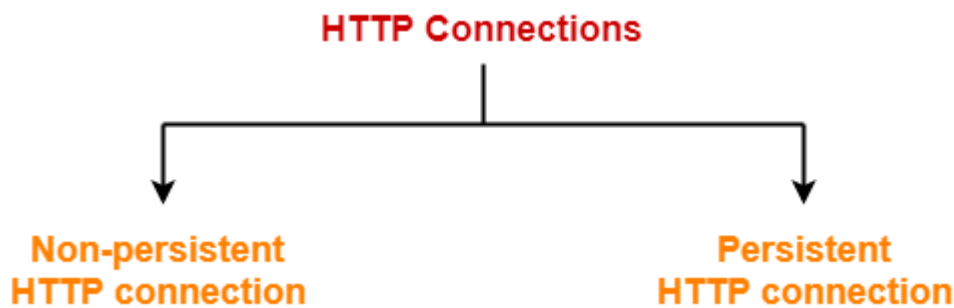
Then-

- HTTP opens a connection between the client and server through **TCP**.
- HTTP sends a request to the server which collects the requested data.
- HTTP sends the response with the objects back to the client.
- HTTP closes the connection.

HTTP Connections-

HTTP connections can be of two types-

1. Non-persistent HTTP connection
2. Persistent HTTP connection

Non-persistent HTTP connection	Persistent HTTP connection
Non-persistent HTTP connection is one that is used for serving exactly one request and sending one response.	Persistent HTTP connection is one that can be used for serving multiple requests.
HTTP server closes the TCP connection automatically after sending a HTTP response.	HTTP server closes the TCP connection only when it is not used for a certain configurable amount of time.
A new separate TCP connection is used for each object.	A single TCP connection is used for sending multiple objects one after the other.
HTTP 1.0 supports non-persistent connections by default.	HTTP 1.1 supports persistent connections by default.

Example-	Example-
Suppose a request has been made for a HTML page that contains 10 images (called objects). Then, With non-persistent connection, all the 11 objects (1 page + 10 images) will be sent one by one. For **getting each object, a new separate connection** will be opened and used.	Suppose a request has been made for a HTML page that contains 10 images (called objects). Then, With persistent connection, all the 11 objects (1 page + 10 images) will be sent one after the other **using a single TCP connection**.

Important Notes-

- HTTP uses TCP at the transport layer.
- HTTP uses port number 80.
- HTTP 1.0 is non-persistent and HTTP 1.1 is persistent.
- HTTP 1.0 is a connectionless protocol.
- HTTP is a stateless protocol.

What If HTTP Is Stateful Protocol?

If HTTP is a stateful protocol, then-

- It will give a chance to the browser window to engage the connection with the web server for a long time.
- This may unnecessarily create a situation of reaching to maximum connections of a web server even though most of the connections are idle.

Simple Mail Transfer Protocol-

- SMTP is short for **Simple Mail Transfer Protocol.**
- It is an application layer protocol.
- It is used for sending the emails efficiently and reliably over the internet.

Working-

- SMTP server is always on a listening mode.
- Client initiates a TCP connection with the SMTP server.
- SMTP server listens for a connection and initiates a connection on that port.
- The connection is established.
- Client informs the SMTP server that it would like to send a mail.
- Assuming the server is OK, client sends the mail to its mail server.
- Client's mail server use DNS to get the IP Address of receiver's mail server.
- Then, SMTP transfers the mail from sender's mail server to the receiver's mail server.
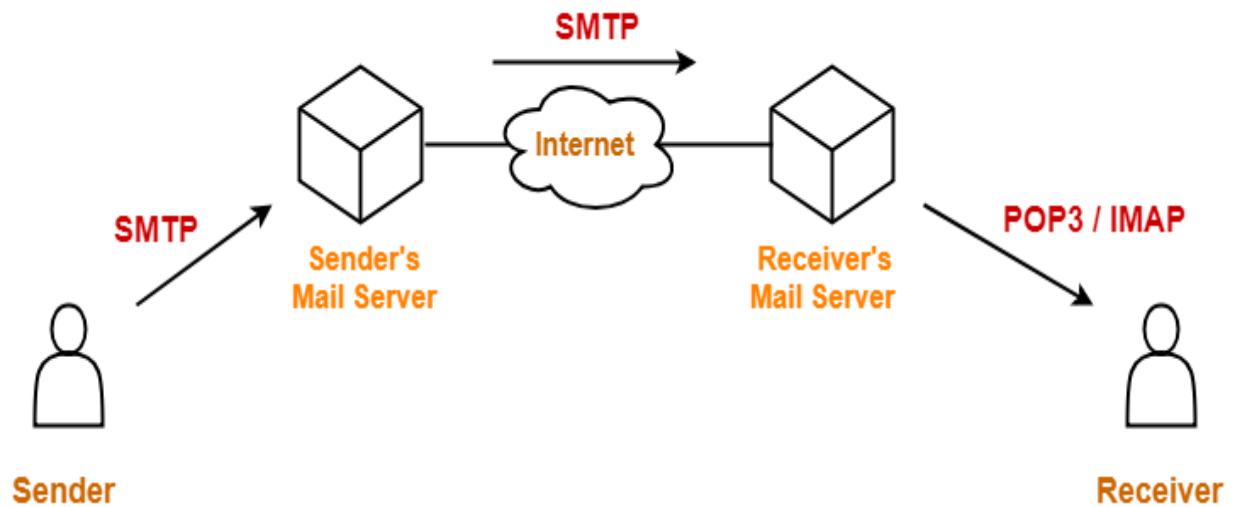
While sending the mail, SMTP is used two times-

1. Between the sender and the sender's mail server
2. Between the sender's mail server and the receiver's mail server

NOTE-

To receive or download the email,

- Another protocol is needed between the receiver's mail server and the receiver.
- The most commonly used protocols are POP3 and IMAP.

Characteristics of SMTP-

- SMTP is a push protocol.
- SMTP uses TCP at the transport layer.
- SMTP uses port number 25.

- SMTP uses persistent TCP connections, so it can send multiple emails at once.
- SMTP is a connection oriented protocol.

Important Points-

- SMTP is a pure text based protocol.

- MIME(Multipurpose Internet Email Extension) extends the limited capabilities of email.

- We can not use SMTP at the receiver's side.

- Sender and receiver can not run SMTP between their machines.

- SMTP is not suitable for client authentication.

File Transfer Protocol-

- FTP is short for **File Transfer Protocol**.
- It is an application layer protocol.

- It is used for exchanging files over the internet.
- It enables the users to upload and download the files from the internet.

Working-

FTP establishes two TCP connections between the client and the server.

- One connection is used for transferring data.
- Other connection is used for transferring control information.
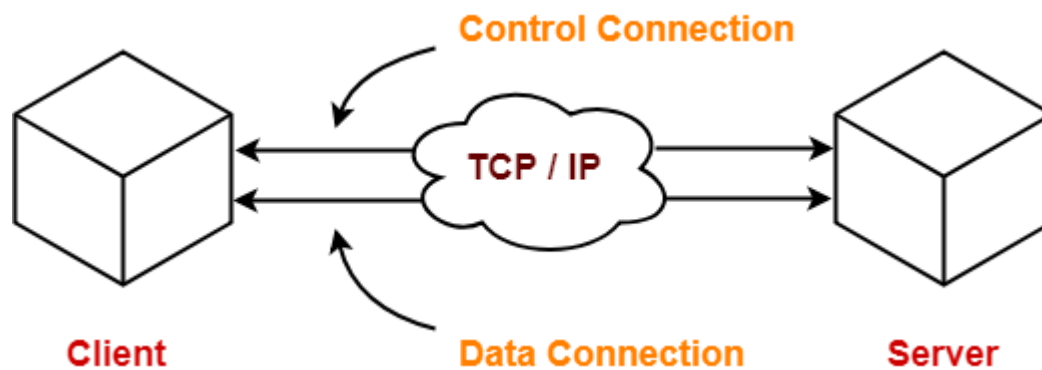
Characteristics of FTP-

- FTP uses TCP at the transport layer.
- FTP uses port number 21 for control connection.
- FTP uses port number 20 for data connection.
- FTP uses persistent TCP connections for control connection.
- FTP uses non-persistent connections for data connection.
- FTP is a connection oriented protocol.
- FTP is an out-of-band protocol as data and control information flow over different connections.

Important Notes-

- Emails can't be sent using FTP.

- FTP can transfer one file at a time.

- FTP is a stateful protocol.