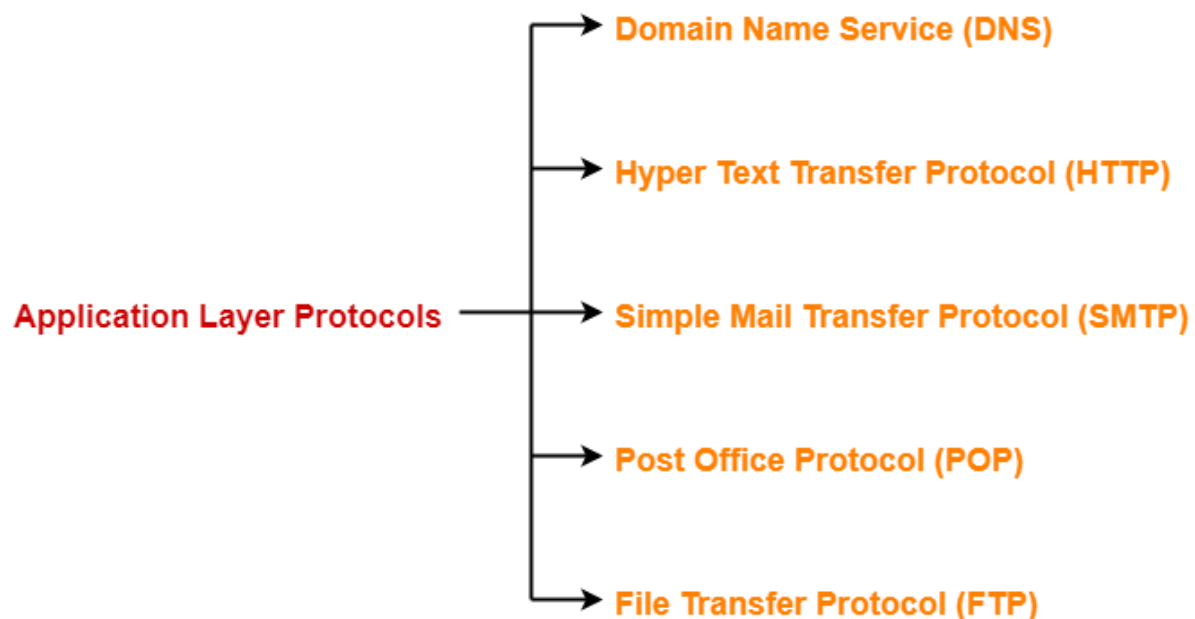


## Application Layer Protocols-



### DNS in Networking-

- DNS is short for Domain Name Service or Domain Name System.
- It is an application layer protocol.

#### Purpose-

- 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-

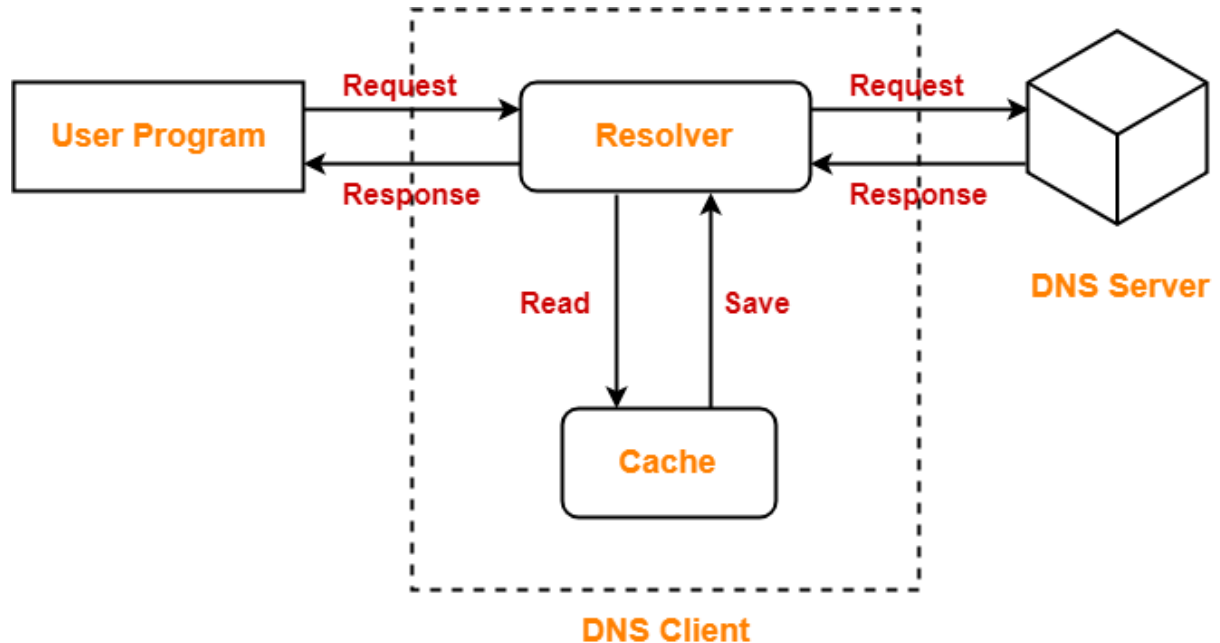
The need for Domain Name Service arises due to the following reasons-

1. IP Addresses are not static and may change dynamically.  
So, a mapping is required which maps the domain names to the IP Addresses of their web servers.
2. 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-



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-

### 1. DNS uses UDP (port 53) at the transport layer.

- DNS uses UDP at the transport layer due to the following reasons-
  - UDP is much faster than TCP.
  - TCP is slow as it uses **Three-way handshake** to start the data transfer.
- DNS requests are very small. So, they fit well within UDP segments.
- Although UDP is not reliable but reliability can be added on application layer.
- Reliability can be added by using timeouts and resend at the application layer.
- Thus, in the end both speed and protection are achieved.

### 2. DNS is a connection less protocol.

- DNS uses UDP at the transport layer for replying to the DNS queries of clients.
- Therefore, it is a connection less protocol.

### 3. DNS is non-persistent.

### 4. DNS is a stateless protocol.

- This is because-
  - DNS server accepts the requests, process them, resolves the query and forget about them.
  - It does not make any assumption how long this will be.

### 5. Mapping an IP Address onto a domain name is referred to as Inverse domain.

- It is important to note-
  - DNS can translate a domain name onto an IP Address.
  - Also, it can translate an IP Address onto a domain name.

### 6. For the first time, there is more delay in translating the domain name onto an IP Address.

- Converting a domain name onto an IP Address is an extra overhead.
- This overhead is called as DNS Overhead.
- It causes an unnecessary delay in serving the request.
- So, there is more delay for the first time.
- To reduce the delay next time, IP Addresses are stored in the computer using log.
- This avoids the DNS overhead next time and takes less time in serving the request.
- When it gets expired, the request is again served through DNS.

## Hyper Text Transfer Protocol-

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

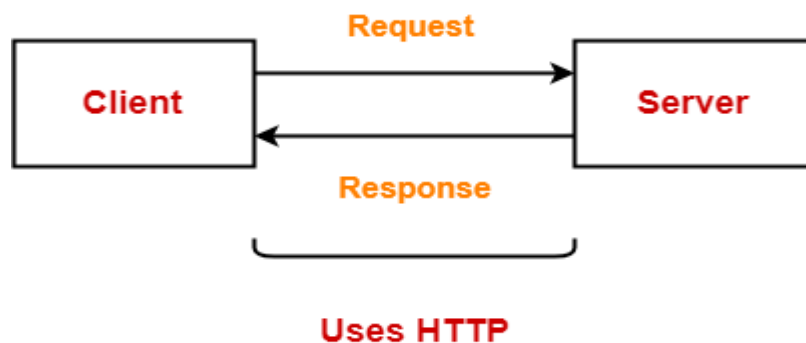
### Purpose-

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

## Important Notes-

### 1. HTTP uses TCP at the transport layer.

- This is because-
  - Unlike UDP, it guarantees the delivery of data via a Three-way handshake.
  - It ensures the re transmission of lost packets.
  - HTTP does not have any inbuilt facility for providing reliability.
  - So, if HTTP uses UDP, then it will have to maintain or handle the session on its own.
  - For example- If a packet gets lost, then HTTP will have to re-transmit the packet.

**2. Any service which does not use TCP should have the inbuilt facility for providing reliability.**

**3. HTTP uses port number 80.**

- HTTP clients use port 80 to send and receive requested web pages from a HTTP server.
- Similarly, HTTP server responds to all the requests at port 80.

**4. HTTP 1.0 is non-persistent and HTTP 1.1 is persistent.**

- Persistent connections improve the performance by 20%.

**5. HTTP 1.0 is a connectionless protocol.**

- This is because-
  - After serving the single HTTP request, the connection is closed and it is not used again.
  - So, HTTP 1.0 without connection keep alive is connectionless.

**6. HTTP is an in-band protocol.**

- This is because-
  - HTTP passes the control data (commands) and main data over the same connection.
  - Both control data and main data are processed in the same way without any distinction.
  - No high priority is given to the control data (commands).

**7. HTTP is a stateless protocol.**

- This is because-
  - HTTP server does not maintain any state.
  - It forgets about the client after sending the response.
  - It treats every new request independently.
  - HTTP closes the connection automatically after generating the response for each request.
  - This ensures that no client can engage connection with web server for a long time.

## **What If HTTP Is Stateful Protocol?**

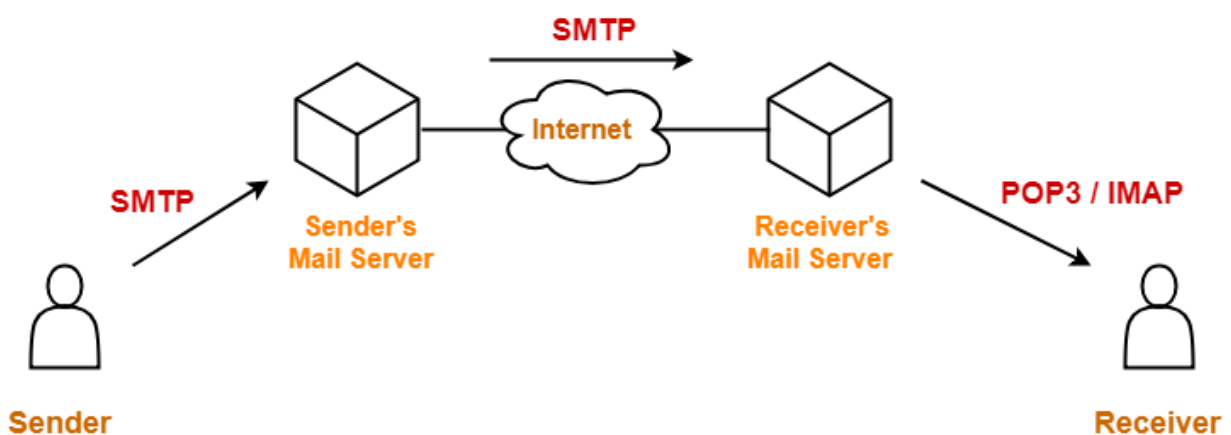
- If HTTP is a stateful protocol, then-
  - It will give a chance to the browser window to engage the connection with the web server for a long time.
  - This may unnecessarily create a situation of reaching to maximum connections of a web server even though most of the connections are idle.

## 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-

- Between the sender and the sender's mail server
- 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.
- SMTP is an in-band protocol.
- SMTP is a stateless protocol.

## Important Points-

### 1. SMTP is a pure text-based protocol.

- SMTP can only handle the messages containing 7-bit ASCII text.
- SMTP cannot transfer other types of data like images, video, audio etc.
- SMTP cannot transfer executable files and binary objects.
- SMTP cannot transfer the text data of other languages like French, Japanese, Chinese etc. (since they are represented in 8-bit codes)

### 2. MIME extends the limited capabilities of email.

- Multipurpose Internet Email Extension (MIME) is an extension to the internet email protocol.
- It extends the limited capabilities of email by enabling the users to send and receive graphics, audio files, video files etc in the message.
- MIME was specially designed for SMTP.

### 3. SMTP is a stateless protocol.

- This is because-
  - It does not maintain the state of its clients.
  - If an email is asked to be sent twice, then SMTP server resends it without saying that the email has already been sent.

### 4. We cannot use SMTP at the receiver's side.

- This is because-
  - SMTP is a push protocol.
  - At receiver's side, a pull protocol like POP3, IMAP is needed.
  - Receiver periodically checks if he has any mail from his mail server.



**5. Sender and receiver cannot run SMTP between their machines.**

- This is because-
  - Machines cannot always be ON.
  - So, the functionality has been divided between the client and the mail server.
  - The mail server receives the mail on behalf of its client and manages the mail box of the client.

**6. SMTP is not suitable for client authentication.**

- This is because-
  - SMTP does not require authentication.
  - It allows anyone on the Internet to send emails to anyone or even to a large group of people.
  - **SMTP Auth** short for SMTP Authentication has been provided for authentication.

## Post Office Protocol-

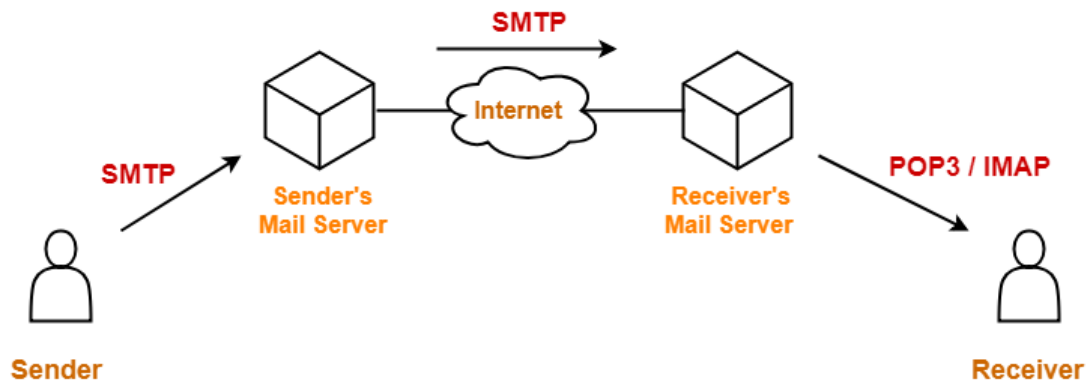
- POP is short for Post Office Protocol.
- It is an application layer protocol.

### Purpose-

- It is a message access protocol.
- It enables the clients to receive or download the emails from their remote mail server.
- POP version 3 (POP3) is the most popularly used version.

### Working-

- To retrieve a message from the mail server-
  - Client establishes a TCP connection using port 110.
  - Client identifies itself to the server.
  - Client issues a series of **POP3** commands.



### Characteristics of POP-

- POP is a pull protocol.
- POP uses TCP at the transport layer.
- POP uses port number 110.
- POP uses persistent TCP connections.
- POP is a connection-oriented protocol.
- POP is an in-band protocol.
- POP is a stateful protocol until the mail is downloaded as well as stateless across sessions.

### Internet Message Access Protocol-

- IMAP is short for Internet Message Access Protocol.
- It is an application layer protocol.
- It also enables the clients to receive or download the emails from their remote mail server.

POP has been largely superseded by Internet Message Access Protocol (IMAP).

## Characteristics of IMAP-

- IMAP is a pull protocol.
- IMAP uses TCP at the transport layer.
- IMAP uses port number 143.
- IMAP uses persistent TCP connections.
- IMAP is a connection-oriented protocol.
- IMAP is an in-band protocol.
- IMAP is a stateful protocol.
- IMAP distributes mail boxes across multiple servers.

## POP Vs IMAP-

The following table lists the differences between POP and IMAP-

Post Office Protocol	Internet Message Access Protocol
POP allows you to read the mail only after downloading it.	IMAP allows you to check the mail content before downloading.  So, with IMAP you can choose to download your messages or just delete them.
The email message is downloaded to the local computer and the copy at the server is deleted.	The email message is stored on the mail server itself.
The user can not organize mails in the mail box of the mail server.	The user can organize mails on the server.
Mails can only be accessed from a single device.	Mails can be accessed from multiple devices which is the biggest advantage.
In POP, the mail server and client's mail account are not synced.  So, changes made in the client's mail account are not visible on the web mail inbox.	In IMAP, the mail server and the client's mail account are synced.  So, changes made in the client's mail account are instantly visible on the web mail inbox.
<b><u>Example-</u></b>  If you are using POP and marks a mail as read, it does not get marked as read in the web mail inbox because the mails are downloaded to the computer and so the changes won't be visible on the server.	<b><u>Example-</u></b>  If you are using IMAP and marks a mail as read, it gets marked as read in the web mail inbox too because the changes are taking place on the server.
POP is a stateful protocol until the mail is downloaded as well as stateless across sessions.	IMAP is a stateful protocol because the IMAP server has to maintain a folder hierarchy for each of its users.
POP is a better choice for those who hardly checks their mail on any other computer.	IMAP is a better choice for those who frequently check their mails on other computers.

## File Transfer Protocol-

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

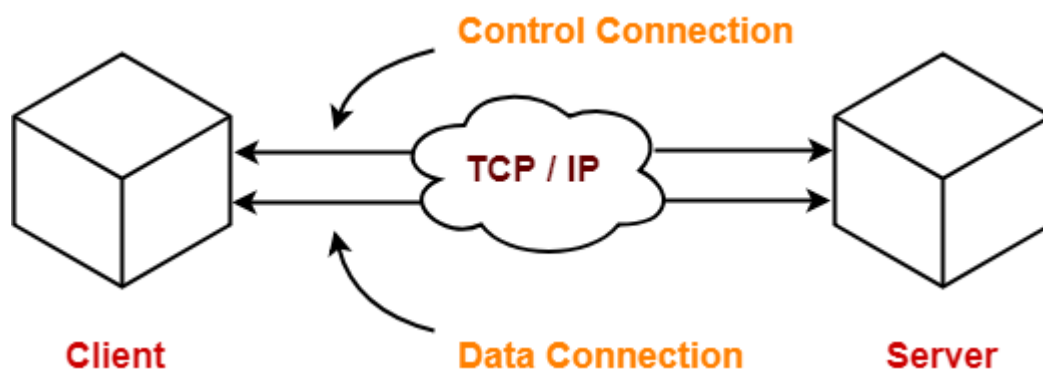
### Purpose-

- 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.
- SMTP is a stateful protocol.

### Important Notes-

#### 1. Emails can't be sent using FTP.

- This is because-
  - FTP requires the connection establishment between the client and server before transferring the files.
  - So, both have to be online at the same time.
  - That is why, emails are not sent using FTP.

**2. FTP can transfer one file at a time.**

- FTP is used for transferring one file at a time in either direction between the client and the server.

**3. FTP is a stateful protocol.**

- This is because-
  - The client establishes control connection for the duration of an FTP session.
  - It typically spans multiple data transfers.
  - So, FTP is a stateful protocol.

**Comparison of Application Layer Protocols-**

The following table compares all the important application layer protocols-

	DNS	HTTP	SMTP	POP	FTP
<b>Stateful / Stateless</b>	Stateless	Stateless	Stateless	Stateful	Stateful
<b>Transport Protocol Used</b>	UDP	TCP	TCP	TCP	TCP
<b>Connectionless / Connection Oriented</b>	Connectionless	Connectionless	Connection Oriented	Connection Oriented	Connection Oriented
<b>Persistent / Non-persistent</b>	Non-persistent	HTTP 1.0 is non-persistent. HTTP 1.1 is persistent.	Persistent	Persistent	Control connection is persistent. Data connection is non-persistent.
<b>Port Number Used</b>	53	80	25	110	20 for data connection. 21 for control connection.
<b>In band / Out-of-band</b>	In band	In band	In band	In band	Out-of-band