Module 5

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web (i.e. internet) since 1990. HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers.

What is HTTP?

Basically, HTTP is a TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.) on the World Wide Web. The default port is TCP 80, but other ports can be used as well. It provides a standardized way for computers to communicate with each other. HTTP specification specifies how clients' request data will be constructed and sent to the server, and how the servers respond to these requests.

Advertisement

-[PS](https://playstream.media/)

PauseSkip backward 5 secondsSkip forward 5 seconds

Mute

Fullscreen

Basic Features of HTTP

The basic features that make HTTP a simple but powerful protocol are mentioned below:

* **Stateless:** Each request is treated as a new request i.e both client and server forget about each other after completion of a request unless cookies, tokens or sessions are used.
* **Connectionless:** The connection between client and server is closed after the completion of each request making it a connectionless protocol.
* **Media Independent:** Any type of data can be transferred over web using HTTP protocol, as long as both the client and server specify the format using MIME types.
* **HTTP Methods:** HTTP defines various methods for different actions such as GET, POST, PUT and more.
* **Caching Support:** HTTP provides support for caching which improves the performance by storing the copies of responses and reusing them later.

*HTTP/1.0 uses a new connection for each request/response exchange, where as HTTP/1.1 connection may be used for one or more request/response exchanges.*

Basic Architecture of HTTP

The following diagram shows a very basic architecture of a web application and depicts where HTTP sits:



The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.

Client

The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

Server

The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.

The Hypertext Transfer Protocol (HTTP) is an application-level protocol that uses TCP as an underlying transport and typically runs on port 80. HTTP is a stateless protocol i.e. server maintains no information about past client requests.

**HTTP Connections**

1. Non-Persistent
2. Persistent

**Basic Pre-Requisite**

The terminology which we must know before going deep into Persistent & Non-Persistent Connections is

1. [RTT(Round Trip Time)](https://www.geeksforgeeks.org/what-is-rttround-trip-time/)
2. [TCP 3-Way Handshake](https://www.geeksforgeeks.org/tcp-3-way-handshake-process/)

**1. RTT:** Time for a small packet to travel from client to server and back.

RTT = 2 X propagation time

1. For a connection Persistent or Non-persistent it is sure that to initiate a [TCP connection](https://www.geeksforgeeks.org/tcp-connection-establishment/) one RTT is used.

2. One RTT is used for the HTTP request and the first few bytes to the HTTP response to return. So to know the total file transmission time.

Total = 2RTT + transmit time

**2. TCP 3-Way Handshake:**TCP Connection establishes in 3 ways, that’s why it is called a 3-way Handshake.

* Requesting the server for the connection.
* The server responds to whether the connection can be established or not.
* Acknowledgment by the client on the response sent by the server.

**Difference between Persistent and Non-Persistent Connections**

| **Persistent HTTP** | **Non-Persistent HTTP** |
| --- | --- |
| The server leaves the connection open after sending a response. | Requires 2 RTTs per object. |
| Subsequent HTTP messages between the same client/server are sent over an open connection. | OS overhead for each TCP connection |
| The client sends requests as soon as it encounters a referenced object. | Browsers often open parallel TCP connections to fetch referenced objects. |
| As little as one RTT for all the referenced objects. | Here, at most one object can be sent over one TCP Connection. |

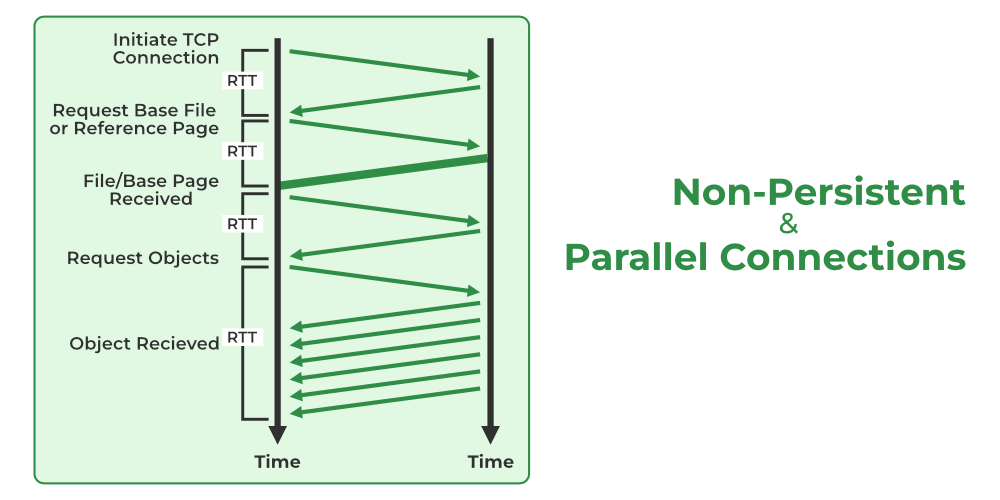
**Non-Persistent Connection**

Non-Persistent Connections are those connections in which for each object we have to create a new connection for sending that object from source to destination. Here, we can send a maximum of one object from one [TCP](https://www.geeksforgeeks.org/what-is-transmission-control-protocol-tcp/) connection.

There are two types:

**1. Non-Persistent-Without parallel connection:**Each objection takes two RTTs (assuming no window limit) one for TCP connection and the other for HTTP image/text file.

**2. Non-Persistent-With parallel connection:**Non-Persistent with a parallel connection requires extra overhead in transferring data.



*Non-Persistent & Parallel Connection*

**Advantages of Non-Persistent Connection**

1. Wastage of Resources is very less because the connection opens only when there is some data to be sent.
2. Non-Persistent Connection is more secure because after sending the data, the connection gets terminated and nothing can be shared thereafter.

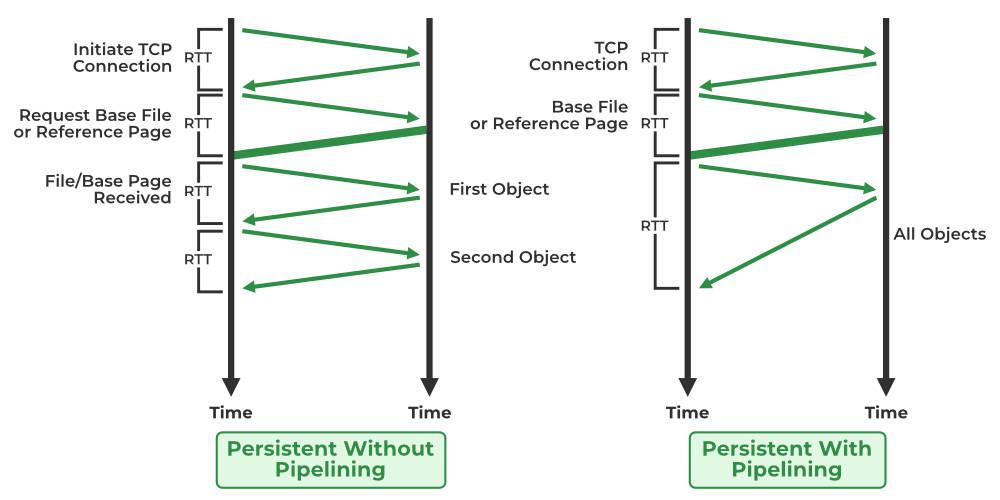
**Disadvantages of Non-Persistent Connection**

1. In Non-Persistent Connection, it requires a greater CPU overhead for the transmission of data

**Persistent Connection**

**1. Non-Pipelined Persistent Connection:** In a Non-pipeline connection, we first establish a connection that takes two RTTs then we send all the object’s images/text files which take 1 RTT each (TCP for each object is not required).

**2. Pipelined Persistent Connection:** In Pipelined connection, 2RTT is for connection establishment and then 1RTT(assuming no window limit) for all the objects i.e. images/text.



*Persistent Without Pipelining and with Pipelining*

**Advantages of Persistent Connections**

* Lower CPU and memory usage because there is less number of connections.
* Allows HTTP pipelining of requests and responses.
* Reduced network congestion (fewer TCP connections).
* Reduced latency in subsequent requests (no handshaking).
* Errors can be reported without the penalty of closing the TCP connection.

**Disadvantages of Persistent Connections**

* Resources may be kept occupied even when not needed and may not be available to others.
* Most modern browsers like Chrome, Firefox, and Internet Explorer use persistent connections.