**Introduction Of HTTP**

HTTP stands for Hypertext Transfer Protocol and it is a protocol which is used to transfer the data between the client and server which is connected to the network. Client computer send the request to the server and server sent response to the client.

Server

Request

Client

Computer

Response

**HTTP/1.1**

HTTP/1.1 is released in 1997 and is a top-level application protocol that exchanges information between a client computer and web server. In this process, a client sends a text-based request to a server by calling a method like GET or POST. In response, the server sends a resource like an HTML page back to the client.The first problem is HTTP/1.1 transfer all the requests & responses in the plain text message form. The second one is head of line blocking in which TCP connection is blocked all other requests until the response does not receive. All the information related to the header file is repeated in every request.

**HTTP/2.0**

HTTP/2 was released in 2015 and it’s works on the binary framing layer instead of textual that converts all the messages in binary format. it works on fully multiplexed that is one TCP connection is used for multiple requests. HTTP/2 uses HPACK which is used to split data from header. it compresses the header. The server sends all the other files like CSS & JS without the request of the client using the PUSH frame.

**Difference between HTTP/1.1 & HTTP/2.0**

|  |  |
| --- | --- |
| **HTTP/1.1** | **HTTP/2.0** |
| It will send all the request and response in text message. | It will send all the request and response in binary frame instead of text converting in to binary format. |
| Head of line blocking will block all the request until the response will received to the client. | It allows multiplexing so one TCP connection is used for multiple requests. |
| It uses requests resource Inline for use getting multiple pages. | It uses PUSH frame by server that collects all multiple pages. |
| It compresses data by itself. | It uses HPACK for data compression. |