**Difference between HTTP1.1 vs HTTP2**

**HTTP/1.1 – *For******better understanding***, let’s assume the situation when you make a request to the server for the Guvi.html page & server responds to you as a resource Guvi.html page. Before sending the request and the response there is a TCP connection established between client & server. Again, you make a request to the server for image img.jpg & the server gives a response as an image img.jpg the connection was not lost here after the first request because we add a keep-alive header which the part of the request so there is an open connection between the server and the client there is a persistent connection which means several request & responses are merged in a single connection. These are the drawbacks that lead to the creation of HTTP/2: The first problem is HTTP/1.1 transfer all the request & 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.

**HTTP2 -** HTTP2 was developed over the SPDY protocol. HTTP2 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. HTTP2 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.

| **HTTP/1.1** | **HTTP/2** |
| --- | --- |
| It works on the textual format. | It works on the binary protocol. |
| There is head of line blocking that blocks all the requests behind it until it doesn’t get its all resources. | It allows multiplexing so one TCP connection is required for multiple requests. |
| It uses requests resource Inlining 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. |