**Difference between HTTP1.1 vs HTTP2**

**HTTP** stands for **Hypertext Transfer protocol** & it is client -server communication.By using HTTP user sends the request to the server & the server sends the response to the user.There are several stages of development of HTTP but we will focus mainly on HTTP1.1 was created on 1997 & the new one is HTTP 2 which is created in 2015.

**HTTP 1.1:**

Developed by Timothy Berners-Lee in 1989 as a communication standard for the World Wide Web, HTTP is a top-level application protocol that exchanges information between a client computer and a local or remote 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.

For better understanding, let’s assume the situation when you make a request to the server for the google.com html page & server responds to you as a resource google.com 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 is the part of the request so there is an open connection between the server & client. there is a persistent connection which means several requests & 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 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:**

HTTP/2 was developed over the SPDY protocol. HTTP/2 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.

From a technical point of view, one of the most significant features that distinguishes HTTP/1.1 and HTTP/2 is the binary framing layer, which can be thought of as a part of the application layer in the internet protocol stack. As opposed to HTTP/1.1, which keeps all requests and responses in plain text format, HTTP/2 uses the binary framing layer to encapsulate all messages in binary format, while still maintaining HTTP semantics, such as verbs, methods, and headers. An application level API would still create messages in the conventional HTTP formats, but the underlying layer would then convert these messages into binary. This ensures that web applications created before HTTP/2 can continue functioning as normal when interacting with the new protocol.

|  |  |
| --- | --- |
| **HTTP 1.1** | **HTTP 2** |
| 1.It works on the textual format. | 1.It works on the binary protocol. |
| 2.There is head of line blocking that blocks all the requests behind it until it doesn’t get its all resources. | 2.It allows multiplexing so one TCP connection is required for multiple requests. |
| 3.It uses requests resource Inlining for use getting multiple pages | 3.It uses PUSH frame by server that collects all multiple pages |
| 4.It compresses data by itself. | 4.It uses HPACK for data compression. |

**CONCLUSION:**

As you can see from this point-by-point analysis, HTTP/2 differs from HTTP/1.1 in many ways, with some features providing greater levels of control that can be used to better optimize web application performance and other features simply improving upon the previous protocol. Now that you have gained a high-level perspective on the variations between the two protocols, you can consider how such factors as multiplexing, stream prioritization, flow control, server push, and compression in HTTP/2 will affect the changing landscape of web development.