**Difference between HTTP1.1 vs HTTP2**

HTTP stands for hypertext transfer protocol & it is used in 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 HTTP/1.1 which was created in 1997 & the new one is HTTP/2 which was created in 2015.

HTTP/1.1: For better understanding, let’s assume the situation when you make a request to the server for the geeksforgeeks.html page & server responds to you as a resource geeksforgeeks.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/1.1 was the third version of HTTP and the standard protocol for over 15 years. It introduced persistent connections for improved performance and laid the foundation for standard requests, such as GET, HEAD, PUT, and POST.As websites became more resource-intensive, however, HTTP/1.1’s limitations began to show. Specifically, its use of one outstanding request per TCP connection created significant overhead, slowing down page load times.

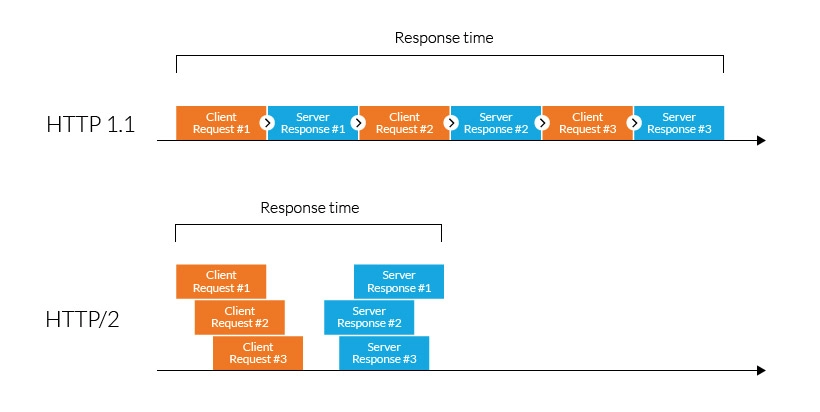
HTTP/2: HTTP/2 was developed over the SPDY protocol. HTTP/2 works on the binary framing layer instead of textual that converts all the messages in binary format. it works on fully multiplexed that is one TCP connection is used for multiple requests. HTTP/2 uses HPACK which is used to split data from 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.

In 2010, Google released the SPDY protocol as a way of modifying how HTTP handles requests and responses. Its focus was on reducing latency via TCP pipelining and providing mandatory compression, amongst other features. While HTTP/2 was initially modeled after SPDY, it was soon modified to include unique features, including a fixed header compression algorithm, (in contrast to SPDY’s dynamic stream-based compression). Following its release, Google announced that it would remove support for SPDY in favor of HTTP/2.

HTTP/1.1 vs. HTTP/2 Protocol

HTTP/2 improved on HTTP/1.1 in a number of ways that allowed for speedier content delivery and improved user experience, including:

**HTTP/1.1 vs. HTTP/2 Protocol**



**Binary protocols** – Binary protocols consume less bandwidth, are more efficiently parsed and are less error-prone than the textual protocols used by HTTP/1.1. Additionally, they can better handle elements such as whitespace, capitalization and line endings.

**Multiplexing** – HTTP/2 is multiplexed, i.e., it can initiate multiple requests in parallel over a single TCP connection. As a result, web pages containing several elements are delivered over one TCP connection. These capabilities solve the head-of-line blocking problem in HTTP/1.1, in which a packet at the front of the line blocks others from being transmitted.

**Header compression** – HTTP/2 uses header compression to reduce the overhead caused by TCP’s slow-start mechanism.

**Server push** – HTTP/2 servers push likely-to-be-used resources into a browser’s cache, even before they’re requested. This allows browsers to display content without additional request cycles.

**Increased security** – Web browsers only support HTTP/2 via encrypted connections, increasing user and application security.