

DIFFERENCE BETWEEN HTTP1.1 & HTTP2

INTRODUCTION:

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.

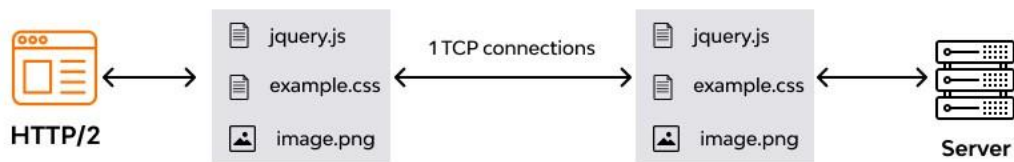
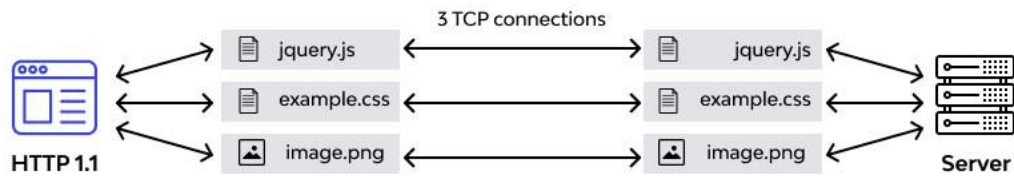
HTTP/1.1:

For better understanding, let's assume the situation when you make a request to the server for the Guvi.html page & server responds to you as a resource of 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 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 doesnot 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.

Multiplexing



Difference between HTTP/1.1 and HTTP/2 are:

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.