**Difference between HTTP1.1 and HTTP 2**

The following are the difference between HTTP1.1 and HTTP 2

|  |  |  |
| --- | --- | --- |
| S.No | HTTP1.1 | HTTP 2 |
| 1 | It works on the textual format. | 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. | It allows multiplexing so one TCP connection is required for multiple requests. |
| 3 | It uses requests resource Inlining for use getting multiple pages | It uses PUSH frame by server that collects all multiple pages |
| 4 | It compresses data by itself. | It uses HPACK for data compression. |
| 5 | HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. | HTTP/2 is able to use a single TCP connection to send multiple streams of data at once so that no one resource blocks any other resource. HTTP/2 does this by splitting data into binary-code messages and numbering these messages so that the client knows which stream each binary message belongs to. |
| 6 | Typically, a server only serves content to a client device if the client asks for it. However, this approach is not always practical for modern webpages, which often involve several dozen separate resources that the client must request. | HTTP/2 solves this problem by allowing a server to "push" content to a client before the client asks for it. The server also sends a message letting the client know what pushed content to expect – like if Bob had  sent Alice a Table of Contents of his novel before sending the whole thing. |
| 7 | Small files load more quickly than large ones. To speed up web performance, both HTTP/1.1 and HTTP/2 compress HTTP messages to make them smaller. | HTTP/2 uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets. This eliminates a few bytes from every HTTP packet. Given the volume of HTTP packets involved in loading even a single webpage, those bytes add up quickly, resulting in faster loading. |