**Difference between HTTP1.1 vs HTTP2**

**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.

**HTTP 2:**

HTTP/2 began as the SPDY protocol, developed primarily at Google with the intention of reducing web page load latency by using techniques such as compression, multiplexing, and prioritization.

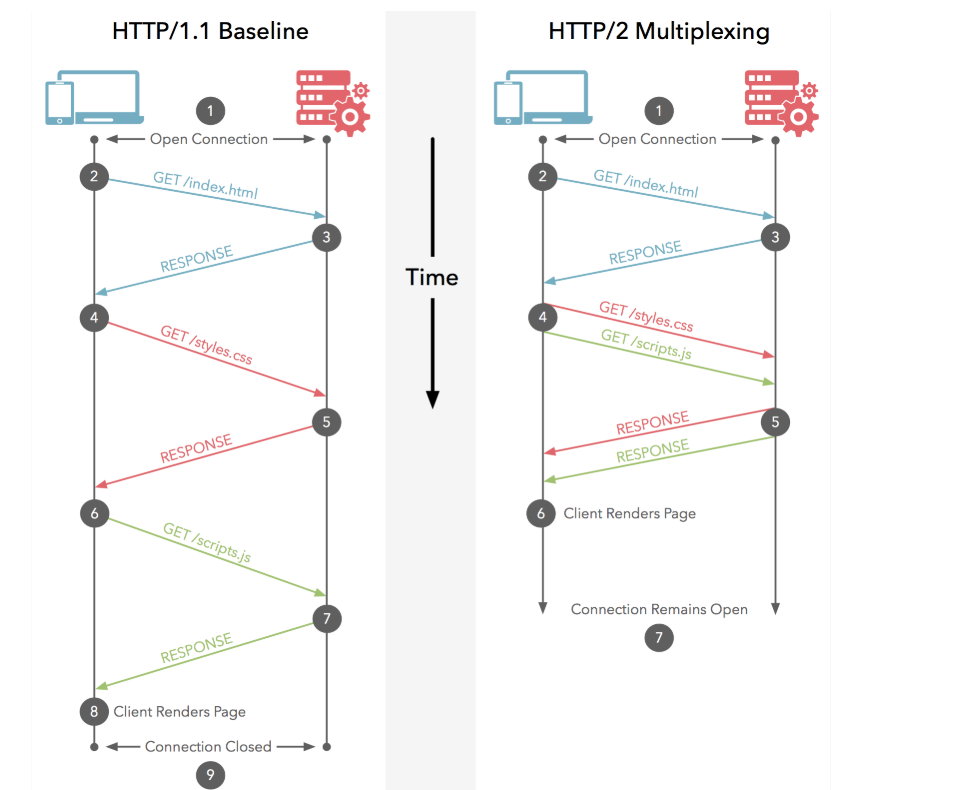
**HTTP1.1 VS HTTP1.2:**

**Https 1.1**

1. Connection Multiplexing: In HTTP/1.1, only one request-response is allowed on a given connection at a time. This means that if a browser wants to download multiple resources (like images, scripts, and stylesheets) from a server, it needs to open multiple connections, leading to increased latency and slower page loading times.
2. Header Compression: Headers are not compressed in HTTP/1.1, leading to higher overhead, especially when multiple requests are made.
3. No Prioritization: Requests are processed in the order they are received, without any priority given to more critical resources.

**Https 2.0**

1. Connection Multiplexing: One of the significant improvements in HTTP/2 is the ability to send multiple requests for data in parallel over a single TCP connection. This is achieved through multiplexing, which allows for more efficient use of network resources.
2. Header Compression: HTTP/2 uses header compression techniques, reducing the amount of data transmitted between the client and the server.
3. Prioritization: HTTP/2 allows for the prioritization of requests, meaning more important resources can be assigned a higher priority, leading to faster page loading times.
4. Binary Protocol: HTTP/2 uses a binary protocol as opposed to the text-based protocol of HTTP/1.1. This makes it more efficient to parse and reduces the chance of errors.
5. Server Push: HTTP/2 introduces server push, allowing the server to push resources to the client before the client requests them. This can improve page load times by anticipating and pushing critical resources.



**HTTP/2 and HTTP/1.1 that impact performance?**

**Multiplexing:**

 HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. In contrast, HTTP/2 is able to use a single [TCP](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/) 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.

**Server push:**

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.

**Header compression:**

 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. However, 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.

**Conclusion:**

HTTP/2 brings several improvements over HTTP/1.1, such as connection multiplexing, header compression, prioritization, and server push, all aimed at making web applications faster and more efficient. HTTPS is the secure version of HTTP, and both HTTP/1.1 and HTTP/2 can be used with either HTTP or HTTPS.