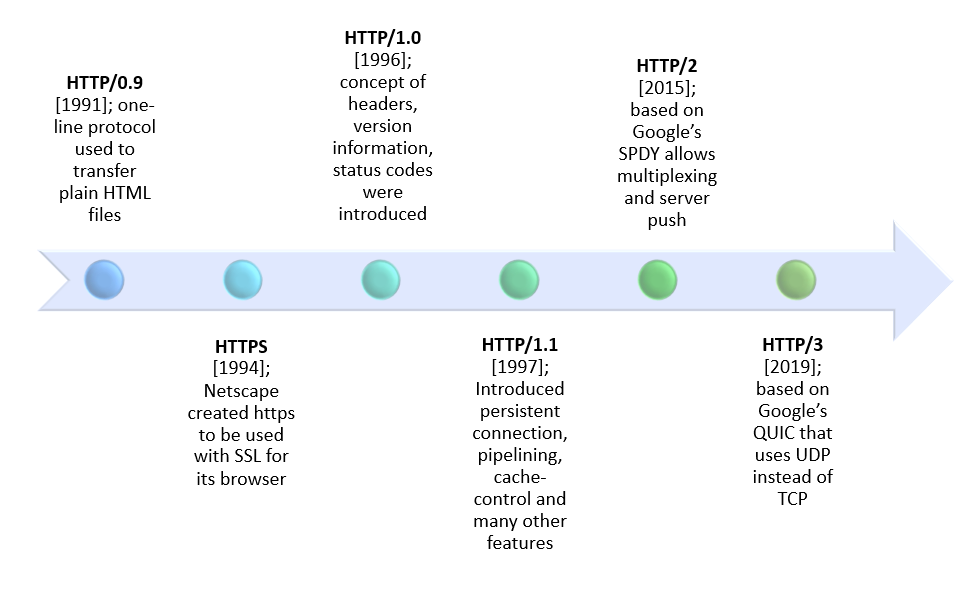
**HTTP - Hypertext Transfer Protocol**

HTTP is a protocol to transfer web content over internet.

HTTP is the set of rules or Protocol for transferring files such as text, images, sound, video and other multimedia files over the web. As soon as a user opens their web browser, they are indirectly using HTTP.

HTTP assigns rules and regulations to transfer a web content.

**History of HTTP**

In 1989, while working at CERN, Tim Berners-Lee wrote a proposal to build a hypertext system over the internet. Initially called the Mesh, it was later renamed the World Wide Web during its implementation in 1990. Built over the existing TCP and IP protocols, it consisted of 4 building blocks

**HTTP methods**

* **GET –** request the server to send the data (eg. Request for a web page)
* **POST –**  request the server to store the data (eg. Username, password..)

**HTTP Features**

* **Connectionless:** One request and one response.

when the request made by a client or browser to the server will get disconnected, till when the server get response and re establishing the connection again.

* **MIME type:**

HTTP can send any sort of data as long as both the client and server can read HTTP.

* **Stateless:**

Connection between the client and the server remains as current request only, server will not store any request.

**HTTP Request / Response**

Communication between clients and servers is done by requests and responses:

**HTTP 1.1 vs HTTP 2.0**

HTTP2.0 Vs. HTTP1.1 is not a debate at all. HTTP2.0 is much faster and more reliable than HTTP1.1. HTTP1.1 loads a single request for every TCP connection, while HTTP2.0 avoids network delay by using multiplexing.

HTTP is a network delay sensitive protocol in the sense that if there is less network delay, then the page loads faster. However, an impressive increase in network bandwidth only slightly improves page load time. This is key to understanding the differences in performance efficiencies between the different versions of HTTP. Back in the day when people used dial up modems web pages were simple and it was the actual data transfer between the server and the client that contributed towards the largest chunk of the page load time. Today the actual downloading of resources from server takes a negligible portion of the total page load time due to the tremendous increase in bandwidth availability. It is the time taken to establish the TCP connection and making requests that impacts performance. It was initially recommended to use only two connections per hostname but today most browsers use six connections per hostname. When we talk about http vs http2.0 in terms of performance it is important to note that a lot of performance optimizations adopted by HTTP 1.1 introduced complexities in terms of developmental efforts as well as network congestion that HTTP 2.0 attempts to address.

The table below points out the differentiating factors between http2.0 vs http1.1:

Header Compression Headers are sent on every request leading to a lot of duplicate data being sent uncompressed across the wire. Header compression is included by default in HTTP2.0 using HPACK. Performance Optimization Provides support for caching to deliver pages faster. Spriting, concatenating, inlining, domain sharding are some of the optimizations used as a workaround to the ‘six connections per host’ rule. Removes the need for unnecessary optimization hacks. Protocol Type Text based protocol that is in the readable form. It is a binary protocol (HTTP requests are sent in the form of 0s and 1s). Needs to be converted back from binary in order to read it. SecuritySSL is not required but recommended. Digest authentication used in HTTP1.1 is an improvement over HTTP1.0. HTTPS uses SSL/TLS for secure encrypted communication. Though security is still not mandatory, it is mostly encrypted (though it is not enforced) since almost all clients require traffic to be encrypted. It also has some minimum standards, such as minimum key size for encryption. TLS 1.2 etc.

Header Compression Headers are sent on every request leading to a lot of duplicate data being sent uncompressed across the wire. Header compression is included by default in HTTP2.0 using HPACK. Performance Optimization Provides support for caching to deliver pages faster. Spriting, concatenating, inlining, domain sharding are some of the optimizations used as a workaround to the ‘six connections per host’ rule. Removes the need for unnecessary optimization hacks. Protocol Type Text based protocol that is in the readable form. It is a binary protocol (HTTP requests are sent in the form of 0s and 1s). Needs to be converted back from binary in order to read it.SecuritySSL is not required but recommended. Digest authentication used in HTTP1.1 is an improvement over HTTP1.0. HTTPS uses SSL/TLS for secure encrypted communication. Though security is still not mandatory, it is mostly encrypted (though it is not enforced) since almost all clients require traffic to be encrypted. It also has some minimum standards, such as minimum key size for encryption. TLS 1.2 etc.

|  |  |  |
| --- | --- | --- |
| Differentiator | HTTP 1.1 | HTTP2.0 |
| year | 1997 | 2015 |
| Key Features | It supports connection reuse i.e. for every TCP connection there could be multiple requests and responses, and pipelining where the client can request several resources from the server at once. However, pipelining was hard to implement due to issues such as head-of-line blocking and was not a feasible solution. | Uses multiplexing, where over a single TCP connection resources to be delivered are interleaved and arrive at the client almost at the same time. It is done using streams which can be prioritized, can have dependencies and individual flow control. It also provides a feature called server push that allows the server to send data that the client will need but has not yet requested |
| Status code | Introduces a warning header field to carry additional information about the status of a message. Can define 24 status codes, error reporting is quicker and more efficient. | Underlying semantics of HTTP such as headers, status codes remains the same. |
| Authentication mechanism | It is relatively secure since it uses digest authentication, NTLM authentication. | Security concerns from previous versions will continue to be seen in HTTP2.0.  However, it is better equipped to deal with them due to new TLS features like connection error of type Inadequate\_Security. |
| caching | Expands on the caching support by using additional headers like cache-control, conditional headers like If-Match and by using entity tags | HTTP2.0 does not change much in terms of caching. With the server push feature if the client finds the resources are already present in the cache, it can cancel the pushed stream. |
| Web traffic | HTTP1.1 provides faster delivery of web pages and reduces web traffic as compared to HTTP1.0. However, TCP starts slowly and with domain sharding (resources can be downloaded simultaneously by using multiple domains), connection reuse and pipelining, there is an increased risk of network congestion. | HTTP2.0 utilizes multiplexing and server push to effectively reduce the page load time by a greater margin along with being less sensitive to network delays. |