**Evolution of Hyper Text Transfer Protocol - HTTP**

HTTP (HyperText Transfer Protocol) is one of the four building blocks of the World Wide Web developed by Tim Berners-Lee and his team during 1989 and 1991 while he was working at CERN. It is a simple protocol used to exchange hypertext documents made using HyperText Markup Language (HTML), hence called the HyperText Transfer Protocol (HTTP).

**HTTP/0.9 – The one-line protocol**

The first version of HTTP had no name and was a very simple version. It was latter called HTTP/0.9, and sometimes called as the one-line protocol. This was done so that it could be differentiated from the later versions. This version is extremely simple as the request only consists of a single line and start with the only possible method GET followed by the path to the resource. The response for the request was also simple as it consisted only the file itself. Only HTML files were transmitted through this version without any headers. There were also no error codes or status codes sent in case of a problem. A HTML file with the description of the problem was sent back for reference.

**HTTP/1.0**

As HTTP/0.9 was very limited, both browsers and servers extended it to be more versatile with HTTP/1.0. This version included more details such as the versioning and status code. This updated version of HTTP allowed the browser itself to understand the success or failure of the request and to adapt its behaviour accordingly. The version also allowed metadata to be transmitted, making the protocol extremely flexible and extensible as HTTP headers had been introduced, both for requests and responses. With the inclusion of HTTP headers, it was now able to transmit other documents instead of a plain HTML files.

**HTTP/1.1 – The standardized protocol**

This version was the first standardized version of HTTP, which was published in early 1997, only a few months after HTTP/1.0. This was done as HTTP/1.0 was used in many diverse implementations since 1995. The version introduced numerous improvements.

* A connection could be reused, saving time to reopen it numerous times to display the resources embedded into it.
* Pipelining was added, allowing a second request to be sent before the response for the first request was fully transmitted. This lowered the latency of the communication.
* Chunked responses were supported.
* Additional cache control mechanisms were introduced.
* Content negotiation which allowed a client and server to agree on the most adequate content to exchange was introduced.
* The ability to host different domains at the same IP address which allows server colocation was possible with the use of Host header.

More than 15 years of extensions were done on this version, thanks to its extensibility. One of the largest changes that happened to HTTP was in the end of 1994 when Netscape communications created an additional encrypted transmission layer on top of the basic TCP/IP stack called Secure Sockets Layer (SSL). This allowed the creation of e-commerce web sites by encrypting and guaranteeing the authenticity of the messages exchanged between the server and client. SSL was put on the standards track and eventually became TLS.

As the original version of HTTP was very well envisioned by Tim Berners-Lee it led to a wide range of extensions to be allowed on it, it even allowed authoring, thus creating a standard called WebDAV which later extended into various specific applications such as CardDAV and CalDAV. The flaw with these DAV extensions were that they had to be implemented by a server to be used. In 2000, a new pattern for using HTTP was designed called representational state transfer (REST). Actions induced by API could only be conveyed by accessing specific URIs with basic HTTP/1.1 methods. This allowed APIs to allow retrieval and modification of its data without having to update browsers and servers.

HTTP is independent of the security model of the Web, the same-origin policy. The model has been developed over the years after its creation. This proved useful and more lenient, allowing certain constraints to be lift some of the restrictions of this policy using a new bunch of HTTP headers defined as Cross-Origin Resource Sharing or Content Security Policy.

**HTTP/2 – A protocol for greater performance**

By the mid of 2010, Google demonstrated an alternative way of exchanging data between client and server by implementing an experimental protocol called SPDY. This showed an increase in responsiveness and solved the problem of duplication of data transmitted. This served as the foundations of the HTTP/2 protocol.

This increase in speed while using the Web was very much needed as the Web pages had become much more complex and large in size. They included large amounts of visual media and also saw a huge increase in the volumes of scripts written for pages, as pages were more interactive and large amounts of data were to be transmitted.

The HTTP/2 version was able to increase the responsiveness as the protocol was in binary rather than text. Parallel request could be handled over the same connection, and it also compressed the headers which removed the duplication and overhead data transmitted. It also introduced a new mechanism called the server push where it allowed a server to populate data in a client cache before being required. This version was officially standardized in May 2015. Since its standardization, high traffic Web sites the most rapid adoption as they saved considerably on data transfer overheads and subsequent budgets.

Like its previous version HTTP/2 also evolved over the years after its release. It could allow smarter Content Delivery Network (CDN) cache mechanism with the support of Alt-Svc. It introduced Client-Hints allowing the Browser to proactively communicate information about its requirements, or hardware constraints, to the server. This version also introduced security related prefixes in the Cookie header which helps to guarantee that a secure cookie has not been altered.

The evolution of HTTP proves its extensibility and simplicity, liberating creation of many applications and compelling the adoption of the protocol.

**HTTP/3 – HTTP over QUIC**

This is an experimental technology. This version of HTTP will use QUIC instead of TCP/TLS for the transport layer portion.