**Hypertext Transfer Protocol (HTTP)**

HTTP (Hypertext Transfer Protocol) is the underlying protocol of the World Wide Web. Developed by Tim Berners-Lee and his team between 1989-1991. The Hypertext Transfer Protocol is an application protocol for distributed, collaborative, hypermedia information systems that allow users to communicate data on the World Wide Web.  
  
  
As a request-response protocol, HTTP gives users a way to interact with web resources such as HTML files by transmitting hypertext messages between clients and servers. HTTP clients generally use Transmission Control Protocol (TCP) connections to communicate with servers.

**HTTP utilizes some specific method in order to perform various tasks. All HTTP servers use the GET and HEAD methods, but not all support the rest of these request methods:**

* GET requests a specific resource in its entirety
* HEAD requests a specific resource without the body content
* POST adds content, messages, or data to a new page under an existing web resource
* PUT directly modifies an existing web resource or creates a new URI if need be
* DELETE gets rid of a specified resource
* TRACE shows users any changes or additions made to a web resource
* OPTIONS shows users which HTTP methods are available for a specific URL
* CONNECT converts the request connection to a transparent TCP/IP tunnel
* PATCH partially modifies a web resource

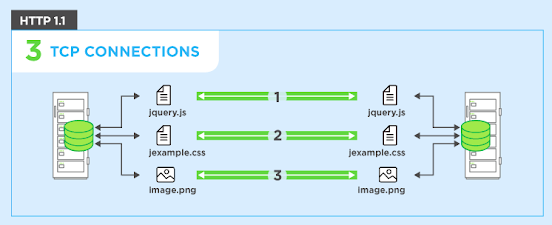
**The invention of the World Wide Web**

* HTTP/0.9 – The one-line protocol
* HTTP/1.0 – Building extensibility
* HTTP/1.1 – The standardized protocol
* More than 15 years of extensions
* HTTP/2 – A protocol for greater performance
* Post-HTTP/2 evolution
* HTTP/3 - HTTP over QUIC

**HTTP/1.1 – The standardized protocol**

HTTP 1.1 is the latest version of Hypertext Transfer Protocol (HTTP), the World Wide Web application protocol that runs on top of the Internet's TCP/IP suite of protocols. HTTP 1.1 provides faster delivery of Web pages than the original HTTP and reduces Web traffic. Developed by a committee of the Internet Engineering Task Force (IETF) that includes the Web's chief creator Tim Berners-Lee, HTTP 1.1 is supported by the latest Web servers and browsers.

* A connection could be reused, which saved time. It no longer needed to be opened multiple times to display the resources embedded in the single original document.
* Pipelining was added. This allowed a second request to be sent before the answer to the first one was fully transmitted. This lowered the latency of the communication.
* Chunked responses were also supported.
* Additional cache control mechanisms were introduced.
* Content negotiation, including language, encoding, and type, was introduced. A client and a server could now agree on which content to exchange.

[](https://www.blogger.com/u/1/blog/post/edit/2777797159233252408/3410384447796525651?hl=en)

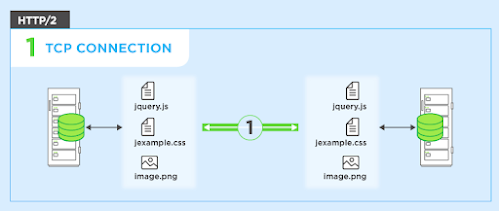
**HTTP/2 – A protocol for greater performance**

In 2015, Internet Engineering Task Force (IETF) release HTTP/2, the second major version of the most useful internet protocol, HTTP. It was derived from the earlier experimental SPDY protocol.

* Protocol negotiation mechanism — protocol electing, eg. HTTP/1.1, HTTP/2 or other.
* High-level compatibility with HTTP/1.1 — methods, status codes, URIs and header fields.
* Page load speed improvements trough:
* Compression of request headers
* Binary protocol
* HTTP/2 Server Push
* Request multiplexing over a single TCP connection
* Request pipelining
* HOL blocking (Head-of-line) — Package blocking

Over the years, web pages became more complex. Some of them were even applications in their own right. More visual media was displayed and the volume and size of scripts adding interactivity also increased. Much more data was transmitted over significantly more HTTP requests and this created more complexity and overhead for HTTP/1.1 connections. To account for this, Google implemented an experimental protocol SPDY in the early 2010s. This alternative way of exchanging data between client and server amassed interest from developers working on both browsers and servers. SPDY defined an increase in responsiveness and solved the problem of duplicate data transmission, serving as the foundation for the HTTP/2 protocol.

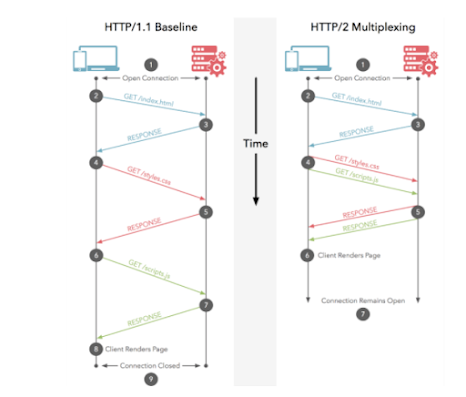
This evolution of HTTP has lead to the creation of many applications and has driven the adoption of the protocol. The environment in which HTTP is used today is quite different from that of the early 1990s. HTTP's original design proved to be scalable, allowing the web to evolve over a quarter of a century. By fixing flaws and retaining the flexibility and extensibility that made HTTP such a success, the adoption of HTTP/2 points to a bright future for the protocol.

[](https://www.blogger.com/u/1/blog/post/edit/2777797159233252408/3410384447796525651?hl=en)

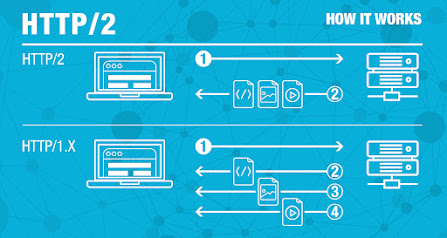
**Difference between HTTP1.1 vs HTTP2**

There are many advancement and the new feature was implemented in HTTP 2.0 , since HTTP 1.1 "practically" only allows one outstanding request per TCP connection. But with HTTP 2.0 we can have multiple request at a time.

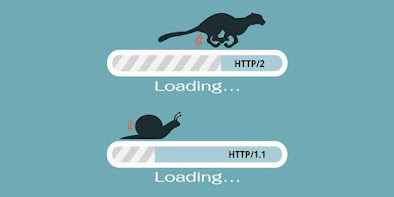
If you can see in example below, A web page comprising of two files:  
  
1) style.css (to enhance the UI display)  
  
2) Script.js (to make interactive web page)

[](https://www.blogger.com/u/1/blog/post/edit/2777797159233252408/3410384447796525651?hl=en)

When we use http 1.1 version, if you request web page at a one time only 1 file is getting loaded on your UI, but with HTTP 2.0 both file will get loaded.

[](https://www.blogger.com/u/1/blog/post/edit/2777797159233252408/3410384447796525651?hl=en)

This is multiplexing multiple requests over a single TCP connection an advantage out of many advantages in HTTP 2.0 together with HTTP headers compression, priority and more intelligent packet streaming management. This results in reduced latency and accelerates content download on modern web pages.

[](https://www.blogger.com/u/1/blog/post/edit/2777797159233252408/3410384447796525651?hl=en)

**HTTP/2 vs. HTTP/1.1: A Performance Analysis**

[](https://www.blogger.com/u/1/blog/post/edit/2777797159233252408/3410384447796525651?hl=en)

* HTTP1.x uses text-based commands to complete HTTP requests. If you were to view one of these requests they would be perfectly readable (to a system admin at least).
* HTTP2, on the other hand, uses binary commands (1s and 0s) to complete HTTP requests. It needs to be converted back from binary to read the request.

This conversion to Binary takes place in the Binary Framing Layer, so only binary commands are transmitted over the network.

* HTTP/2 introduces an extra step. It divides HTTP/1.x messages into frames, which in turn are embedded in a stream.

With frames, the Data and header frames are separated. Not only does this allow compression, but it also allows multiplexing, making the underlying TCP connections more efficient