**BLOG ON DIFFERENCE BETWEEN HTTP1.1 vs HTTP2**

***HTTP (hyper text transfer protocol)*** *is a protocol for fetching resources such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. A complete document is reconstructed from the different sub-documents fetched, for instance, text, layout description, images, videos, scripts, and more.*

*From the release of HTTP/1.1 in 1997 until recently, there have been few revisions to the protocol.*

* ***HTTP/0.9 - The one-line protocol – GET/mypage.html***
* ***HTTP/1.0 - Building extensibility – HTTP headers***
* ***HTTP/1.1 - Standardized protocol***
* ***HTTP/2 - A protocol for greater performance***
* ***HTTP/3 - HTTP over QUIC***

***THE DIFFERENCE’S***

* *To put it simple http/2 is faster than http/1.1 hence the amount of time to fully establish the loaded data/page/video...etc. is faster hence the security is also higher compared to http/2, this is the prime difference between http/2 and http/1.1.*
* *Though http/1.1 is a good version which withstood 15 years which had the advantage of withstanding the changes that happened parallelly in web and also has some really good features such as CORS, it has some flaws and it definitely can’t withstand more than 15 years, which are:*
* *It leads to additional HTTP requests, usually repeating some of the information sent in each request. It also requires one request finish before the next one begins.*

***HTTP/2 aims to correct these issues:***

The [key differences with HTTP/1.x](https://http2.github.io/faq/#what-are-the-key-differences-to-http1x) are that HTTP/2:

* is binary, instead of textual.
* is fully multiplexed, instead of ordered and blocking.
* can serve multiple files in parallel over a single connection.
* uses header compression to reduce overhead.
* allows servers to “push” responses proactively into client caches.

***Binary protocols*** *are more efficient to parse and less error prone than textual protocols. They make it faster to transfer data and they’re machine friendly. This is good for performance and not something you need to configure. It’s a free optimization you get for switching to HTTP/2.*

***Multiplexing*** *means multiple files and requests can be transferred at the same time as opposed to HTTP/1.1, which only accepts a single request per connection. In other words, it allows requests to be transferred in parallel instead of in series, which means it doesn’t lead to the overhead of establishing multiple connections. Multiplexing eliminates the issues associated with head-of-line blocking.*

***Header compression*** *reduces the overhead of each request making the requests smaller and allowing more requests to fit into a single IP packet. You don’t have to send the same cookies, referrers, and other headers with every request. It leads to less data being transferred and less requests overall.*

*With* ***server push****, a server can anticipate future requests and send information before it’s requested. For example, instead of sending an HTML file and then waiting for requests for CSS, JavaScript, images, etc., the server can send these resources knowing in advance that the client is going to request them.*

*The downside to server push is the server is sending resources that may already live in cache. A solution is cache digests which lets the client tell the server what it already has cached so the server only needs to push the resources that are needed.*

*Another advantage of* [*HTTP/2*](https://http2.github.io/faq/) *is stream prioritization, which allows the client to specify the order in which it wants to receive resources.* [*HTTP/2*](https://kinsta.com/learn/what-is-http2/) *is also backwards compatible. Your browser likely uses it already for sites that are delivered over HTTP/2 and falls back for those delivered over HTTP/1.1.*

*There are some potential downsides with HTTP/2. Since the optimization strategies are different, if we only optimize for HTTP/2 we potentially penalize anyone using a browser that doesn’t yet support it. This problem gets smaller every day and will continue to become less of an issue in the days ahead.*

*Still not advertising for http/2 version, both 1.1 and 2 has different use cases and both should be used based on the requirement, though the future is towards http/2 and http/3.*