HTTP2

DD Owned by David Cherney DISH •••
Jun 21, 2023 • Initializing...

Here is a quick history of HTTP to introduce how HTTP2 is different than its predecessors.

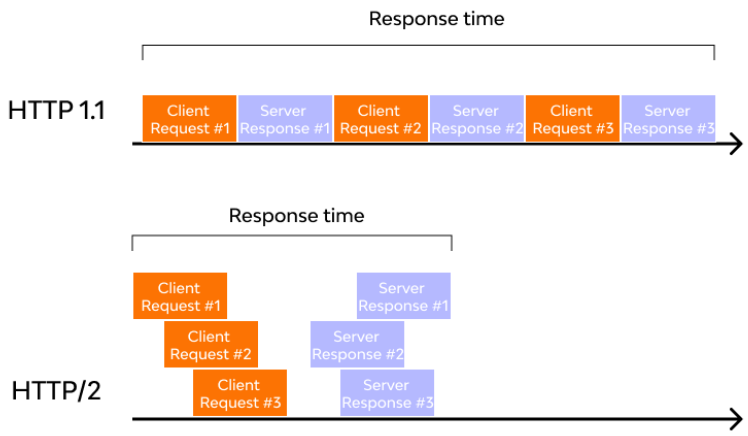
HTTP/0.9, the first version known to written history, used only an address in a single line to retrieve documents in HTML format; there were no responses.

HTTP/1.0 allowed multiple lines in requests, introduced response messages, and allowed files types other than HTML. It was not regulated, and there were many different flavors. By 1996 something had to be done.

HTTP/1.1 has been in continuous use since 1997. It introduced persistent connections via TCP, tags for character sets and languages, cookies, authentication, secure HTTP, compression, and more. By 2014, websites had become too big to be loaded in a timeframe comfortable for users because of HTTP/1.1's limitations; the internet was slow. The limitations included a maximum of 2 connections to servers, a need to wait for available connections to servers, repetitive headers and cookies, and the slow start time of TCP connections.

HTTP/2 was standardized in 2015, is based on Google's 2009 SPDY designed to make Chrome the top browser, and is an extension of HTTP/1.1. By introducing "binary headers" (aka binary framing) it enabled numerous enhancements;

- declare the size of the header fields before they are read,
- index frames as belonging to bi-directional streams within the connection,
- allowed client or server to initiate streams within a connection so long as the other allows it,
- allows client or server to close streams,
- gives parent-child relationships to streams with the rule that parents must be processed before children,
- gives relative weight to sibling streams with allocation of resources proportional to relative weight, and
- compresses headers by requiring client and server to remember previously seen headers and their frequencies from streams so that efficient Huffman encoding of headers is enabled.



Thus, HTTP/2 uses a TCP connection that lasts as long as the request response interaction takes which has sophisticated multiplexing via a hierarchy of streams, and compression where it matters.

Multiplexing