**DIFFERENCE BETWEEN HTTP1.1 AND HTTP2**

**->INTRODUCTION**

**HTTP - Hyper Text Transfer Protocol**

* WWW is about communication between web clients and servers.
* Communication between client computers and web servers is done by sending HTTP Requests and receiving HTTP Responses

The Hypertext Transfer Protocol (HTTP) is an application-level

protocol for distributed, collaborative, hypermedia information

systems. HTTP has been in use by the World-Wide Web global

information initiative since 1990. The first version of HTTP,

referred to as HTTP/0.9, was a simple protocol for raw data transfer

across the Internet. HTTP/1.0, as defined by RFC 1945.

HTTP1.0 improved the protocol by allowing messages to be in the format of    MIME-like messages, containing metainformation about the data transferred and

modifiers on the request/response semantics. However, HTTP1.0 does

not sufficiently take into consideration the effects of hierarchical

proxies, caching, the need for persistent connections, or virtual

hosts.

**->HTTP1.1**

In 1997, when HTTP/1.1 was published, several enhancements were introduced. This brought new features, as well as clarification for ambiguities in the protocol.

The HTTP/1.1 standard was published only a few months after HTTP1.0, yet a version change was necessary because many applications pronounced themselves “HTTP1.0”, but they were not fully implemented. Therefore, the specification of a more recent version of the protocol was necessary for clients and servers to properly understand each other’s capabilities.

Among the features added to HTTP/1.1 were additional [cache controls](https://http.dev/caching), [Content Negotiation](https://http.dev/content-negotiation) that allowed for different languages, content encodings, and types. For example, a client and server can come to an agreement as to which representation is most suitable. New [HTTP request methods](https://http.dev/methods) were added, as well as support for many new [HTTP headers](https://http.dev/headers).

A notable improvement was that a [HTTP Connection](https://http.dev/http-connection) can be reused. In common instances where images are referenced in an *HTML* file, a subsequent [HTTP Connection](https://http.dev/http-connection) does not have to be made to download each one. This saves in terms of connection overhead, in particular, speeding up media-rich documents. Moreover, pipelining support allowed for a second HTTP request to be sent before the first was complete, reducing the overall latency of the exchange.

Yet another enrichment came from the ability to support server co-location. With the addition of the HTTP [Host](https://http.dev/host) header, the protocol was able to support servers hosted on different domains but located at the same IP address.

**->HTTP2**

HTTP/2 (originally named HTTP/2.0) is a major revision of the [HTTP](https://en.wikipedia.org/wiki/HTTP) network protocol used by the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). It was derived from the earlier experimental [SPDY](https://en.wikipedia.org/wiki/SPDY) protocol, originally developed by [Google](https://en.wikipedia.org/wiki/Google).HTTP/2 was developed by the HTTP Working Group (also called httpbis, where "[bis](https://en.wiktionary.org/wiki/bis)" means "twice") of the [Internet Engineering Task Force](https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force) (IETF). HTTP/2 is the first new version of HTTP since HTTP 1.1, which was standardised in [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2068](https://datatracker.ietf.org/doc/html/rfc2068) in 1997. The Working Group presented HTTP/2 to the [Internet Engineering Steering Group](https://en.wikipedia.org/wiki/Internet_Engineering_Steering_Group) (IESG) for consideration as a Proposed Standard in December 2014, and IESG approved it to publish as Proposed Standard on February 17, 2015 (and was updated in February 2020 in regard to [TLS 1.3](https://en.wikipedia.org/wiki/TLS_1.3)). The HTTP/2 specification was published as [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [7540](https://datatracker.ietf.org/doc/html/rfc7540) on May 14, 2015.

**MAJOR DIFFERENCES**

1. **PRIORITIZATION**

In HTTP/2, developers have hands-on, detailed control over prioritization. This allows them to maximize perceived and actual page load speed to a degree that was not possible in HTTP/1.1.

HTTP/2 offers a feature called weighted prioritization. This allows developers to decide which page resources will load first, every time. In HTTP/2, when a [client](https://www.cloudflare.com/learning/serverless/glossary/client-side-vs-server-side/) makes a request for a webpage, the server sends several streams of data to the client at once, instead of sending one thing after another. This method of data delivery is known as multiplexing. Developers can assign each of these data streams a different weighted value, and the value tells the client which data stream to render first.

**2.MULTIPLEXING**

HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. In contrast, HTTP/2 is able to use a single [TCP](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/) connection to send multiple streams of data at once so that no one resource blocks any other resource. HTTP/2 does this by splitting data into binary-code messages and numbering these messages so that the client knows which stream each binary message belongs to.

**3.SERVER PUSH**

 Typically, a server only serves content to a client device if the client asks for it. However, this approach is not always practical for modern web pages, which often involve several dozen separate resources that the client must request. HTTP/2 solves this problem by allowing a server to "push" content to a client before the client asks for it. The server also sends a message letting the client know what pushed content is.

**4.HEADER COMPRESSION**

 Small files load more quickly than large ones. To speed up web performance, both HTTP/1.1 and HTTP/2 compress HTTP messages to make them smaller. However, HTTP/2 uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets. This eliminates a few bytes from every HTTP packet. Given the volume of HTTP packets involved in loading even a single webpage, those bytes add up quickly, resulting in faster loading.