

1. Difference between HTTP1.1 vs HTTP2

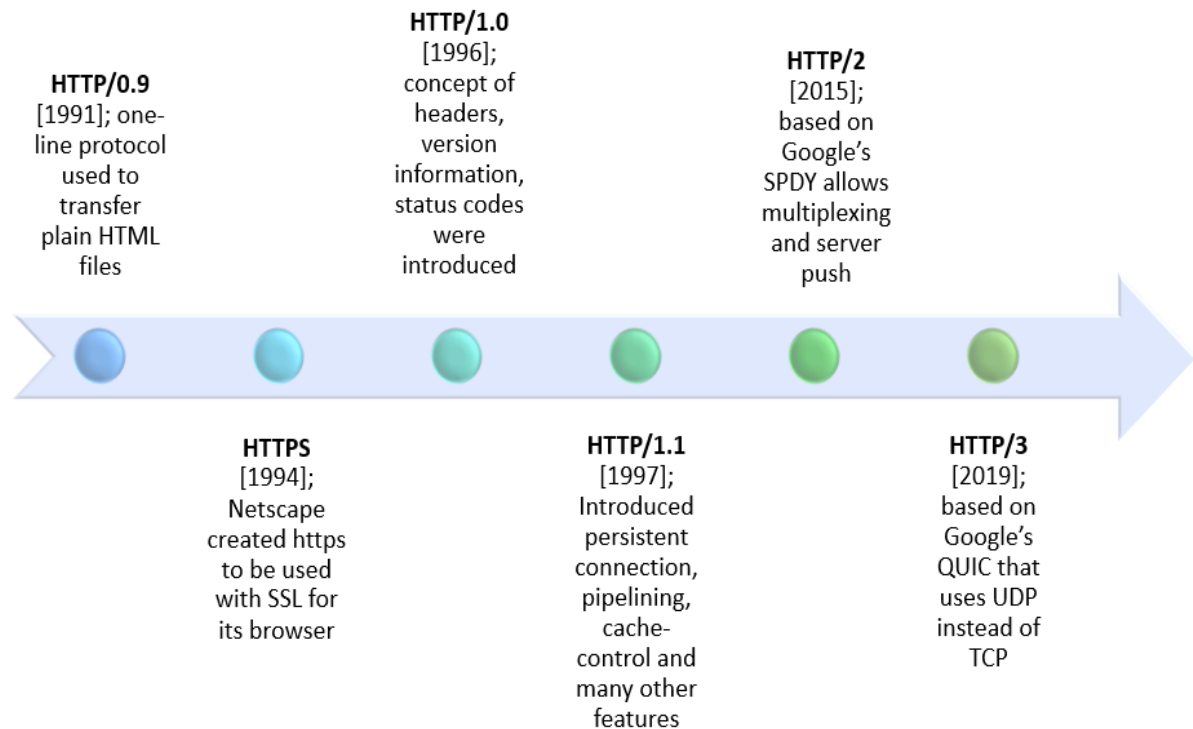
HTTP is based on the Client/Server model. Client/Server model can be explained as two computers, Client (receiver of service) and Server (provider of service) that are communicating via requests and responses.

A simple example would be a restaurant guest and a waiter. The guest (Client) asks (sends request) waiter (Server) for a meal, then the waiter gets the meal from the restaurant chef (your application logic) and brings the meal to the guest.

HTTP/1.1:	HTTP/2
It was no longer required for each connection to be terminated immediately after every request was served with a response; instead, with the keep-alive header, it was possible to have persistent connections. It allowed multiple requests/responses per TCP connection.	It introduces the concept of a server push where the server anticipates the resources that will be required by the client and pushes them prior to the client making requests. The client retains the authority to deny the server push; however, in most cases, this feature adds a lot of efficiency to the process.
The Upgrade header was used to indicate a preference from the client that made it possible to switch to a more preferred protocol if found appropriate by the server.	Introduces the concept of multiplexing that interleaves the requests and responses without head-of-line blocking and does so over a single TCP connection.
HTTP/1.1 provided support for chunk transfers that allowed streaming of content dynamically as chunks and for additional headers to be sent after the message body. This enhancement was particularly useful in cases where values of a field remained unknown until the content had been produced. For example, when the content had to be digitally signed, it was not possible to do so before the entire content gets generated.	It is a binary protocol i.e. only binary commands in the form of 0s and 1s are transmitted over the wire. The binary framing layer divides the message into frames that are segregated based on their type – Data or Header. This feature greatly increases efficiency in terms of security, compression and multiplexing.
Other features that reinforced its stability were introduced such as: <ul style="list-style-type: none">➤ pipelining (the second request is sent before the response to the first is adequately served)➤ content negotiation (an exchange between client and server to determine the media type, it also provides the provision to serve different versions of a resource at the same URI)➤ cache control (used to specify caching policies in both requests and responses)	HTTP/2 uses HPACK header compression algorithm that is resilient to attacks like CRIME and utilizes static Huffman encoding.

--	--

2. http version history



List 5 difference between Browser JS(console) vs Nodejs

JS(CONSOLE)	NODE JS
1. USES RENDERING AND JS ENGINE	1. USES JS ENGINE
2. Alert and prompt can be used	2. cannot be used
3. Runs Only in Browsers	3. runs as application like visual studio
4. execute <Script>filename.js</Script>	4. execute >node filename.js file
5. Front-end	5. Back-end

4.what happens when you type a URL in the address bar in the browser?

- User writes www.example.com in URL
- It goes to DNS from there it generates a unique DNS id for the website after that,
- It initiates TCP connection,
- Requests for HTTP,
- GET the required server response,
- Starts loading HTML/CSS/JS layout (includes all DOM & Parse trees),
- Hence User is showed the desired Website

DNS(Domain Name Server) contains Unique ID For Every Website