Difference between HTTP1.1 vs HTTP2?

**HTTP/1.1** - this was the prevalent format of HTTP until recently. It is a text-based protocol and has some inefficiencies in it - especially when requesting lots of resources like a typical web page. HTTP/1.1 messages can be unencrypted (where web site addresses start http://) or encrypted with HTTPS (where web site address start with https://). The client uses the start of the URL to decide which protocol to use, usually defaulting to http:// if not provided.

**HTTP/2** - a new version of HTTP released in 2015 which addresses some of the performance issues by moving away from a text based protocol to a binary protocol where each byte is clearly defined. This is easier to parse for clients and servers, leaves less room for errors and also allows multiplexing. HTTP/2, like HTTP/1.1, is available over unencrypted (http://) and encrypted (https://) channels but web browsers only support it over HTTPS, where it is decided whether to use HTTP/1.1 or HTTP/2 as part of the HTTPS negotiation at the start of the connection.

HTTP/2 is used by about a third of all websites at the time of writing (up to 50% of websites as of Jan 2020, and 67% of website requests). However not all clients support HTTP/2 so you should support HTTP/1.1 over HTTPS and HTTP/2 over HTTPS where possible (I believe node automatically does this for you when using the http module). I do not believe HTTP/1.1 will be retired any time soon. You should also consider supporting HTTP/1.1 over unencrypted HTTP and then redirect to HTTPS version (which will then use HTTP/1.1 or HTTP/2 as appropriate). A web server like Apache or Nginx in front of Node makes this easy.

**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.

**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 webpages, 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 to expect – like if Bob had sent Alice a Table of Contents of his novel before sending the whole thing.

**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.