**HTTP1.1 vs HTTP2**

**HTTP:**

* HTTP stands for ***H****yper****T****ext* ***T****ransfer* ***P****rotocol*.
* HTTP is the method computers and servers use to request and send information.
* The HTTP protocol did not use headers and only transmitted plain HTML files.
* It was a one-line protocol only supporting the GET method.
* As the need to exchange more than just plain HTML emerged along with the client and server applications becoming more mature, HTTP/1.0

**HTTPS:**

* In 1994, Netscape Communications created HTTPS (*Hypertext Transfer Protocol Secure*) to be used with SSL for its web browser, Netscape Navigator.
* The need for encrypted transmission channels emerged as the applications being designed shifted towards a more commercial market where advertisers, unknown individuals, and cybercriminals could have easy access to personal data

**HTTP/1.1:**

Version 1.1 was released in 1997 and became the Internet Standard. This version added many performance enhancements, including keepalive connections, caching mechanisms, request pipelining, transfer encodings, and byte-range requests.

**HTTP/2:**

* HTTP 2 is the second major version of the HTTP network protocol used for transmitting data over the Internet.
* It was developed to improve the performance and efficiency of web applications by reducing the amount of data sent over the wire.

**HTTP 2.0 vs. 1.1:**

|  |  |  |
| --- | --- | --- |
|  | **HTTP 1.1** | **HTTP 2** |
| **Development** | It was developed in the year 1997 | It was developed in the year 2015. |
| **Compression** | It compresses data by itself. | It uses HPACK for data compression. |
| **Security** | A text-based protocol uses plain text to encode and transmit data. | It works on the binary protocol as a series of binary codes encode and transmit data rather than plain text. |
| **Security** | The client sends a request to a server, and the server sends a response back to the client. | A different underlying protocol called Secure Remote Protocol 2 (SRP2) establishes a secure connection between a client and a server |
| **Multiplexing** | A separate connection is established for each request and response, which can add overhead and latency to the communication process. | It allows multiplexing so multiple requests and responses can be sent over a single connection. |
| **Buffer Overflow** | HTTP 1.1 cannot handle buffer overflow vulnerabilities due to the lack of sufficient measures. | HTTP 2 includes measures to prevent buffer overflow vulnerabilities. |
| **Performance** | HTTP 1.1 does not include any in-built features, so the performance it delivers is less efficient. | HTTP 2 is designed to be more efficient and performant than HTTP 1.1. This is because HTTP 2 includes several features like multiplexing, binary protocol and header compression. |

**Conclusion:**

* HTTP/2 differs from HTTP/1.1 in many ways, with some features providing greater levels of control that can be used to better optimize web application performance and other features simply improving upon the previous protocol.
* Considering how such factors as multiplexing, stream prioritization, flow control, server push, and compression in HTTP/2 will affect the changing landscape of web development.