**Assignment: 1**

**(Day-1 Introduction to Browser & Web)**

**Question:1 Difference between HTTP1.1 vs HTTP2**

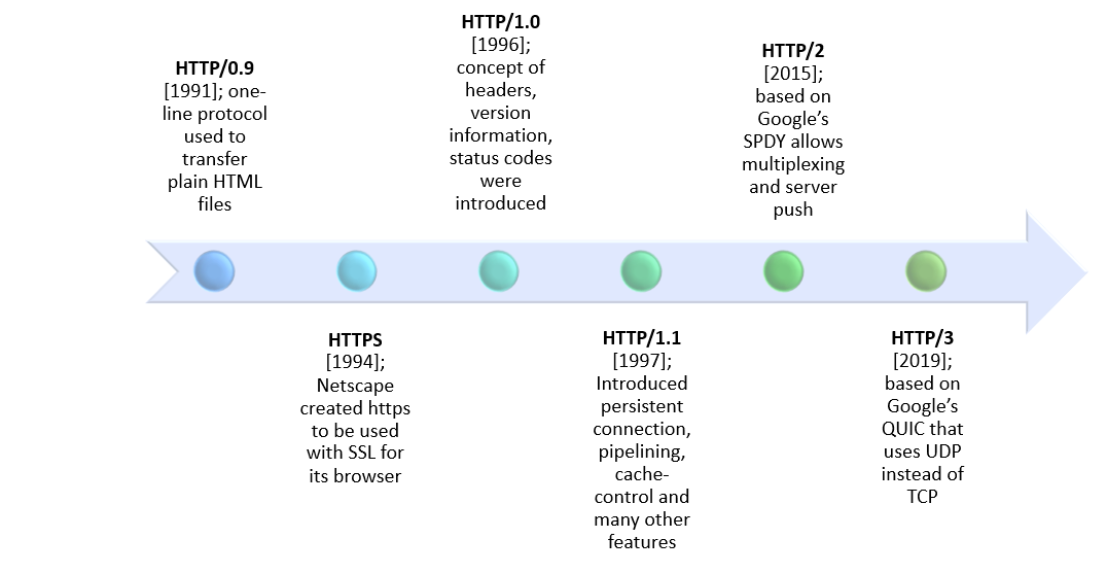
**What is Network Protocol?**

In networking, a protocol is a set of rules for formatting and processing data. Network protocols are like a common language for computers. The computers within a network may use vastly different software and hardware; however, the use of protocols enables them to communicate with each other regardless.

**Which Protocol Runs on the Network Layer?**

* [**IPsec:**](https://www.cloudflare.com/learning/network-layer/what-is-ipsec/) Internet Protocol Security (IPsec)
* [**ICMP:**](https://www.cloudflare.com/learning/ddos/glossary/internet-control-message-protocol-icmp/) The Internet Control Message Protocol (ICMP)
* **IGMP:** The Internet Group Management Protocol (IGMP)

**Road Map of HTTP:**



## Advancement from HTTP/1.1 To HTTP/2:

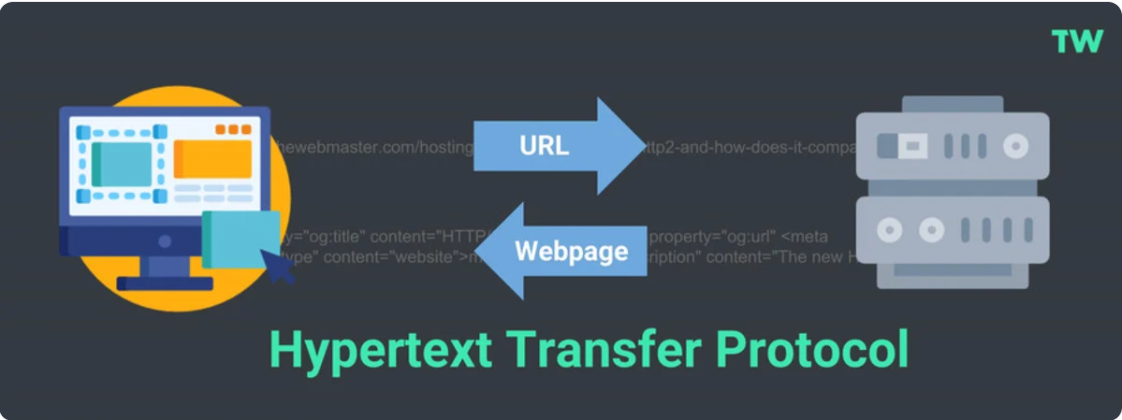
HTTP stands for hypertext transfer protocol, and it is the basis for almost all web applications. More specifically, HTTP is the method computers and servers use to request and send information.

The first usable version of HTTP was created in 1997. Because it went through several stages of development, this first version of HTTP was called HTTP/1.1. This version is still in use on the web. In 2015, a new version of HTTP called HTTP/2 was created.

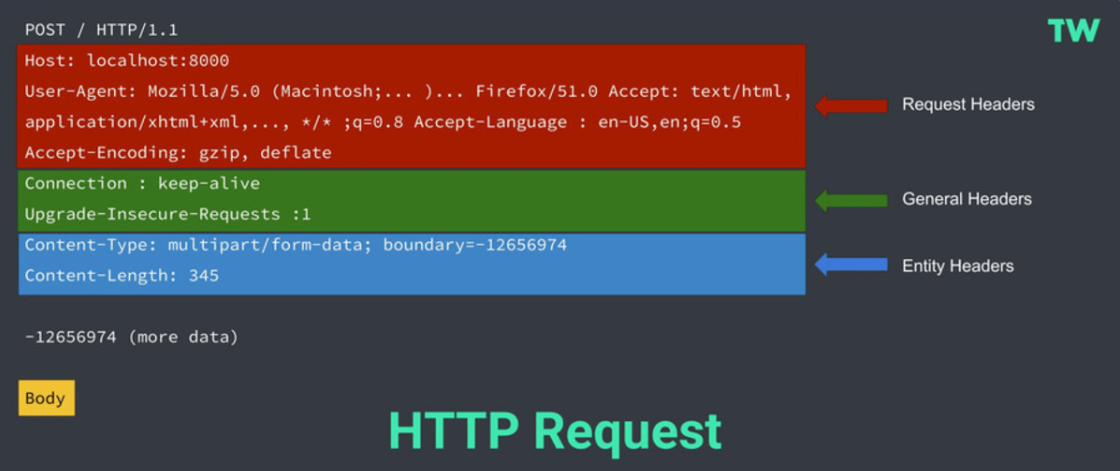
**What is HTTP?**

HTTP stands for Hypertext Transfer Protocol. It is the foundation of the World Wide Web and is used by browsers to load web pages.

A typical example is when your browser sends an HTTP request to a web server after entering an URL. The HTTP command then provides an HTTP response to the web server with the webpage’s contents.



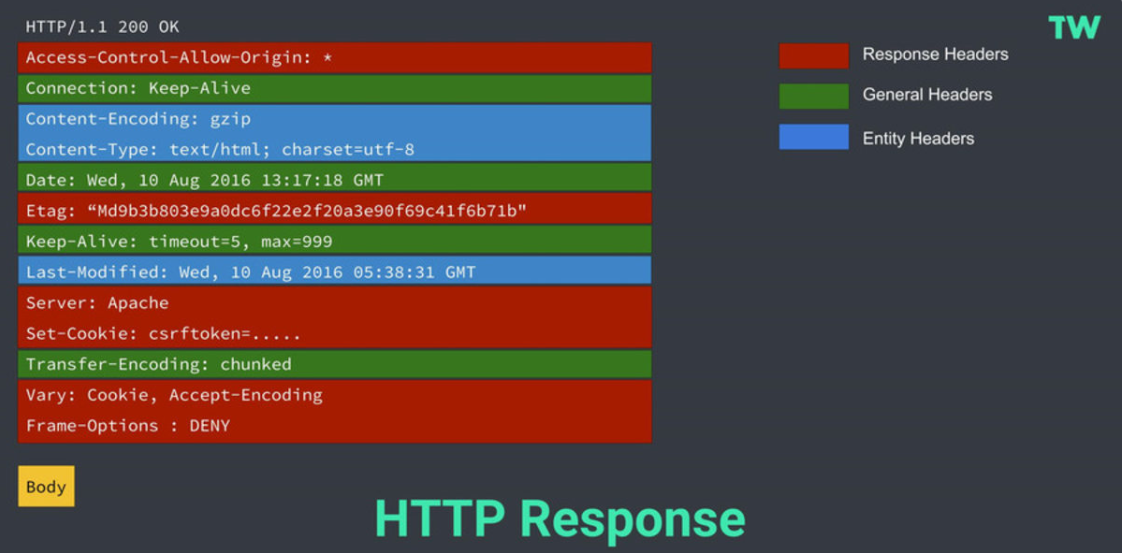
HTTP Request



HTTP requests typically comprise the following:

* Start-line - This describes the HTTP Method (such as Get, Put, or Post), the request target (such as an URL or Port), and the HTTP version (such as HTTP/1.1). This start-line is always a single line.
* Request Headers - An optional set of HTTP headers specifying the request. There are multiple different types of headers:
* Request Headers - Includes User-Agent, Accept-Type, Accept-Language.
* General Headers - Includes Connection.
* Entity Headers - Includes Content-Type or Content-Length.
* Blank Line - This confirms that all the request meta-data has been sent.
* Body (Optional) - This contains all the data associated with the request. There are typically two categories:
  + A body consisting of one single file, defined by the Content-Type and Content-Length Entity Headers.
  + A multipart body. One example could be a request containing information in an HTML form.

HTTP Response:



HTTP responses typically comprise the following:

1. **Start-line** - This usually includes the HTTP protocol version (HTTP/1.1), a status code (such as 200 or 404), and a textual description of the status code (such as “ok”).
2. **Response Headers** - An optional set of HTTP headers specifying the request. There are multiple different types of headers:

* **Response Headers** - such as Vary and Accept-Ranges, provide more information about the server.
* **General Headers** - such as Via, apply to the whole message.
* **Entity Headers** - such as Content-Length, or Last-Modified, apply to the response body.

1. **Body** - This contains all the data associated with the request. There are typically two categories:

* A body consisting of one single file, defined by the Content-Type and Content-Length Entity Headers.
* A body consisting of one single resource of unknown length encoded by chunks (Transfer-Encoding set to Chunked).
* A multipart body, with each part containing different information. These are not common.

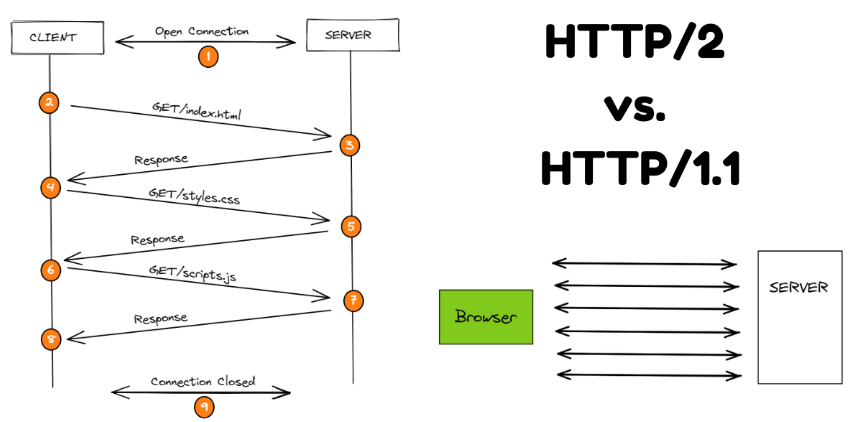
**What is HTTP/2?**

HTTP/2 is the next version of HTTP and is based on Google’s SPDY Protocol (originally designed to speed up the serving of web pages). It was released in 2015 by the Internet Engineering Task Force (IETF).

It is important to note that HTTP/2 is not a replacement for HTTP. It is merely an extension, with all the core concepts such as HTTP methods, Status Codes, URIs, and Header Fields remaining the same.

# **HTTP/2 vs. HTTP/1.1 - How their differences impact the performance?**

# HTTP/2 solves several problems that the creators of HTTP/1.1 did not anticipate. In particular, HTTP/2 is much faster and more efficient than HTTP/1.1.



# **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 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.

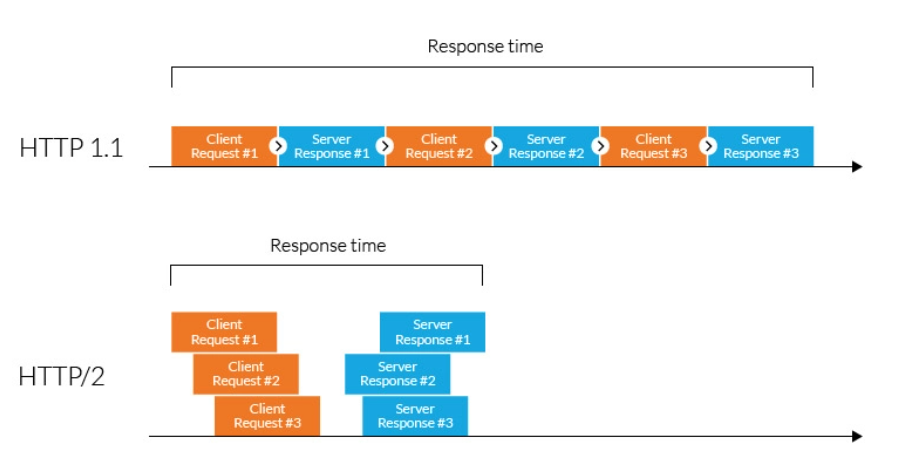
# **Header compression:** Small files load more quickly than large ones. To speed up web performance, both HTTP/1.1 and HTTP/2 compress HTTP messages to make them smaller. However, HTTP/2 uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets. 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.

# **Prioritization:** Prioritization refers to the order in which pieces of content are loaded. Prioritization affects a webpage's load time. For example, certain resources, like large JavaScript files, may block the rest of the page from loading if they have to load first. More of the page can load at once if these render-blocking resources load last. 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.

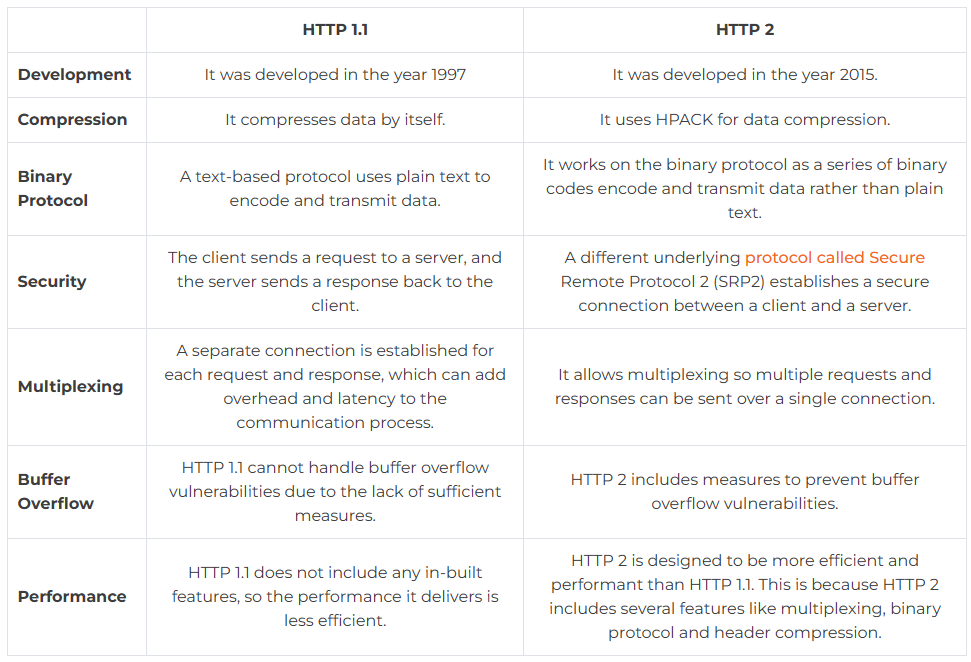
**Binary protocols:** Binary protocols consume less bandwidth, are more efficiently parsed and are less error-prone than the textual protocols used by HTTP/1.1. Additionally, they can better handle elements such as whitespace, capitalization and line endings.

**Increased security:** Web browsers only support HTTP/2 via encrypted connections, increasing user and application security.

## Pictorial Representation of HTTP/1.1 vs. HTTP/2 Protocol:



**Key Difference Between HTTP 1.1 vs HTTP 2:**



**Which Browser Supports HTTP 2 and HTTP 1?**

Most modern web browsers support HTTP 1.1 and  HTTP 2 including [Google Chrome](https://cheapsslweb.com/blog/how-to-fix-err-ssl-version-interference-error-in-chrome), Mozilla Firefox, Apple Safari, Microsoft Edge, and Opera.

However, the HTTP version might vary based on the version of the browser, and some older versions may not support it. So it all comes down to the operating system, as well as the specific configuration of the browser.

**Conclusion:**

HTTP is the backbone of the internet and enables the creation and sharing of various types of information and resources. Without HTTP, the internet would not be able to function in the way that it does today. However, with time HTTP versions kept changing.