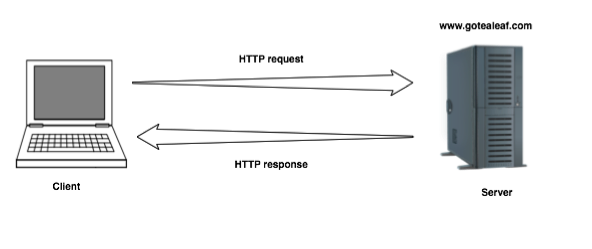
**HTTP 1.1 vs HTTP 2**

**Introduction**

HTTP stands for Hyper Text Transfer Protocol. It is a synonym of “Accessing Internet” for most of us. It was invented by Tim Berners-Lee in 1989. It is used in client-server communication. The communication protocol is your aid in reaching out to World Wide Web. Technically, it sends the request to the server and the server sends the response to user.



**HTTP 1.1**

It processes,

1. The client (browser) has to send a request to the server using the method (GET/POST).
2. Server responds with the requested resource, for example – image, alongside the status of what it did to the client’s request.

Keep in mind that this is not a one-time process. Such requests and responses needs to be transferred between both these machines until the client receives all the resources, essential to load a web page on the end-user’s (your) screen.

This request-response exchange can be regarded as an IP stack being handled by transfer layer and networking layers before finally reaching to the application layer.

For Example,

In the situation when you make a request to the server for the Google.html page & server responds to you as a resource Google.html page.

Before sending the request and the response there is a TCP connection established between client & server.

Again you make a request to the server for image img.jpg & the server gives a response as an image img.jpg.

the connection was not lost here after the first request because we add a keep-alive header which is the part of the request so there is an open connection between the server & client.

There is a persistent connection which means several requests & responses are merged in a single connection.

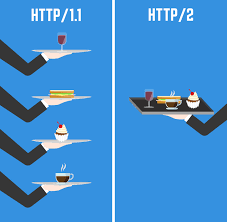
**Problem faced in HTTP 1.1,**

No.1

HTTP/1.1 transfer all the requests & responses in the plain text message form.

No.2

Head of line blocking in which TCP connection is blocked all other requests until the response does not receive. All the information related to the header file is repeated in every request.



**HTTP 2.0**

HTTP/2 was developed over the SPDY(“Speedy”) protocol.

HTTP/2 works on the binary framing layer instead of textual that converts all the messages in binary format. it works on fully multiplexed that is one TCP connection is used for multiple requests.

HTTP/2 uses HPACK algorithm which is used to split data from header. it compresses the header. The server sends all the other files like CSS & JS without the request of the client using the PUSH frame.

HTTP/2 supports full multiplexing for requests as well as responses over a single TCP connection. Due to these capabilities, lower page load times are achieved by removing needless latency and improving the overall capacity of network.

Resource usage has increased dramatically for machines processing requests to deliver media-rich content and complex web designs. Developers worked hard around optimization hacks, and as a result, the robust solution of HTTP/2 was obtained.



|  |  |
| --- | --- |
| HTTP 1.1 | HTTP 2.0 |
| It uses Textual format. | It uses Binary format |
| It uses simple TCP | It uses SPDY Protocol |
| Header file is repeated in every request | HPACK is used to split header from the data |
| GET/POST methods are used for request and response | It supports full multiplexing |
| Long performance timing | Short performance timing |