http 1 works as with the het and response between the client and the server,where the GET is used to send the request to the client and response is between the client and the server hence forth back until all the data like images and text are received.

Whereas the http 2 has been introduced lately after a long time and it also uses the same http1 verbs and all but the distinguish is that the binary framing layer,which can be thought as a part of application layer.as the http1 uses the get and response requests in the plain text format this uses the binary framing that encapsulates all the messages in binary format while maintain the http semantics.the api would still create message sin api formats but the underlying layer would convert these message sin binary.

Pipelining and head of line blocking in http1-

The http 1 that client receives is not fully rendered it requires the full rendering of the webpagehence it has to make seperate connections where the tcp connection has to be made separately and later in 1.1 it is been made that multiple tcp connections has been made to client and server and the tcp is said to closewhen it is directly told.

There is a bottleneck to this optimization strategy,since multiple data packets accesa to destination and at times where the head of the queue that is used to get the data got stucked so that it will block the request behind it this is called head of line blocking.

Advantages of binary framing layer in http 2-

http2 makes a single connection between 2 machines,in this there are multiple streams of data.each stream consists of multiple messages in response/request.each of these messages are splitted into smaller units called frames.whereas this rectifies the head of line blocking and The interleaved requests and responses can run in parallel without blocking the messages behind them, a process called multiplexing.

A single TCP connection also improves the performance of the HTTPS protocol, since the client and server can reuse the same secured session for multiple requests/responses. In HTTPS, during the TLS or SSL handshake, both parties agree on the use of a single key throughout the session. If the connection breaks, a new session starts, requiring a newly generated key for further communication. Thus, maintaining a single connection can greatly reduce the resources required for HTTPS performance. Note that, though HTTP/2 specifications do not make it mandatory to use the TLS layer, many major browsers only support HTTP/2 with HTTPS.