### The Differences: HTTP/1.1 vs. HTTP/2

1. **Multiplexing**: HTTP/2 introduces a significant improvement in multiplexing, allowing multiple requests and responses to be exchanged concurrently over a single TCP connection. In contrast, HTTP/1.1 uses multiple connections to achieve a similar result. With HTTP/2, a single connection can carry multiple streams, avoiding the bottleneck issues associated with numerous connections.
2. **Header Compression**: HTTP/2 uses header compression, which reduces the overhead of sending HTTP headers with each request and response. In HTTP/1.1, headers are not compressed, and this overhead can be substantial, especially for mobile devices or slower connections.
3. **Binary vs. Textual**: HTTP/1.1 uses plain text for its protocol, making it more human-readable but less efficient for machines to process. HTTP/2, on the other hand, uses a binary format, which is more compact and easier for machines to parse, leading to faster data transmission.
4. **Server Push**: HTTP/2 introduces a feature called server push, allowing the server to send additional resources to the client before they are requested. This preemptive loading of resources can significantly improve page load times, as the client doesn't have to wait for requests to fetch additional assets.
5. **Flow Control**: HTTP/2 includes built-in flow control mechanisms, ensuring that one slow-loading resource does not block the loading of other resources. This feature is especially helpful in cases where a high-latency connection might hinder performance.
6. **Prioritization**: HTTP/2 allows for stream prioritization, enabling the client and server to specify the importance of different resources. This ensures that critical assets are delivered and loaded first, improving the user experience.
7. **Header Fields**: HTTP/2 reduces the number of header fields required, making it more efficient. In contrast, HTTP/1.1's numerous header fields add to the overall overhead, resulting in slower load times.
8. **HOL Blocking**: HTTP/1.1 is susceptible to head-of-line (HOL) blocking, which occurs when a slow request blocks the delivery of subsequent requests. HTTP/2 mitigates HOL blocking by multiplexing requests and responses.