HTTP/1.1: The Old Workhorse

HTTP/1.1, the predecessor to HTTP/2, has been the workhorse of the web for over a decade. While it served the internet remarkably well during its time, it has its limitations that have become increasingly apparent as web content and user expectations have grown more complex. Here are some key characteristics of HTTP/1.1:

Sequential Processing: In HTTP/1.1, each request and response is processed sequentially. This means that if a browser wants to load multiple resources from a server, it has to send separate requests for each resource and wait for each response before proceeding to the next one. This can lead to "head-of-line blocking" issues, slowing down page loading times.
High Latency: Because of the serial nature of HTTP/1.1, it can suffer from high latency, especially for websites that require numerous requests to load, such as those with many images, stylesheets, and scripts. Each round-trip between the browser and server adds to the overall latency.
Resource Bundling: To mitigate the latency problem, web developers often use techniques like resource bundling (combining multiple resources into a single file) or domain sharding (spreading resources across multiple domains). While these workarounds help, they introduce complexities and can be inefficient.
Header Overhead: HTTP/1.1 has substantial header overhead. For each request, there's a significant amount of redundant header information that needs to be sent, increasing the size of the data transferred.

HTTP/2: The Modern Solution

HTTP/2 was developed to address the shortcomings of HTTP/1.1 and to improve web performance and user experience. It introduced several fundamental changes that revolutionized the way web content is delivered:

Multiplexing: One of the most significant improvements in HTTP/2 is multiplexing, which allows multiple requests and responses to be sent and received simultaneously over a single TCP connection. This eliminates the head-of-line blocking problem and reduces latency.
Header Compression: HTTP/2 uses header compression techniques like HPACK, significantly reducing the amount of redundant header data sent with each request, which was a major source of overhead in HTTP/1.1.
Server Push: HTTP/2 enables server push, a feature that allows the server to proactively send resources to the client before the client requests them. This can greatly improve page load times by eliminating the need for additional round-trip requests for critical resources.
Stream Prioritization: HTTP/2 allows for stream prioritization, meaning that the client can specify the order in which it wants resources to be delivered. This ensures that critical resources are prioritized, enhancing overall performance.
Binary Protocol: HTTP/2 uses a binary protocol instead of the text-based protocol used in HTTP/1.1. This makes it more efficient to parse, reducing processing overhead.
Backward Compatibility: HTTP/2 is designed to be backward compatible with HTTP/1.1, allowing websites to gradually transition to the new protocol without breaking compatibility with older browsers and servers.