|  |  |
| --- | --- |
| **HTTP 1** | **HTTP 2** |
| **1.HTTP/1.1: Uses a single connection for each request, leading to potential bottlenecks and increased latency for multiple simultaneous requests.** | **1.HTTP/2: Supports multiplexing, allowing multiple requests and responses to be sent concurrently over a single connection, improving efficiency and reducing latency.** |
| **2.HTTP/1.1: Headers are not compressed, leading to increased overhead and slower performance, especially for larger websites.** | **2.HTTP/2: Implements header compression, reducing the amount of data transmitted and improving overall efficiency.** |
| **3HTTP/1.1: Text-based protocol, which can be less efficient due to parsing overhead.** | **3.HTTP/2: Utilizes a binary protocol, reducing parsing complexity and improving the speed of communication between clients and servers.** |
| **4.HTTP/1.1: Relies on the client to request each resource individually, limiting server-initiated communication.** | **4.HTTP/2: Introduces server push, allowing servers to proactively send resources to clients before they are explicitly requested, improving page loading times.** |
| **5.HTTP/1.1: Does not provide explicit support for prioritizing requests, potentially leading to sub optimal resource loading.** | **5HTTP/2: Supports request prioritization, allowing more important resources to be loaded first for an improved user experience.** |
| **6.HTTP/1.1: Requires opening and closing connections for each request, incurring additional latency.** | **6.HTTP/2: Facilitates connection reuse, reducing the need to open and close connections for subsequent requests, leading to faster performance.** |
| **7.HTTP/1.1: Each request may require a separate round-trip, contributing to increased latency.** | **7.HTTP/2: Reduces the number of round-trips through features like multiplexing, improving overall responsiveness.** |
| **8.HTTP/1.1: Headers are sent with each request and response, contributing to larger overall header sizes.** | **8.HTTP/2: Header compression and the use of binary encoding reduce the size of headers, decreasing the amount of data transmitted.** |
| **9.HTTP/1.1: Widely supported and compatible with older systems.** | **9.HTTP/2: Maintains backward compatibility with HTTP/1.1 and can be used over the same ports, although some older systems and proxies may not fully support it.** |
| **10.HTTP/1.1: Inefficient handling of resources, with potential for redundant data transfer and slower loading times.** | **10.HTTP/2: Optimizes resource usage through features like multiplexing and header compression, resulting in more efficient communication and faster page loading.** |