DIFFERNCE IN HTTP1.1 VS HTTP2

| **Feature** | **HTTP/1.1** | **HTTP/2** |
| --- | --- | --- |
| **Multiplexing** | No multiplexing; single request per connection | Multiplexing allows multiple requests and responses in parallel over a single connection. Each request/response is assigned a stream ID. |
| **Header Compression** | Headers are sent as plain text, leading to redundancy | Header compression reduces overhead by using a technique called HPACK, resulting in more efficient data transfer. |
| **Resource Prioritization** | No built-in support; all resources are treated equally | Supports priority levels for resources, allowing the client to indicate the importance of each resource. |
| **Server Push** | Not supported | Allows the server to push resources to the client before they are explicitly requested, improving performance. |
| **Connection Handling** | Requires multiple connections for parallelism | Uses a single connection per origin, reducing latency and improving efficiency. |
| **Error Handling** | Blocks execution until the error is resolved | Supports multiplexing, allowing other requests to proceed even if one encounters an error. |
| **Binary Protocol** | Text-based protocol (ASCII) | Binary framing layer for efficient data transfer. |
| **Backward Compatibility** | Fully backward compatible with HTTP/1.0 | Designed to be fully backward compatible with HTTP/1.1. Older browsers and servers can still work with HTTP/2. |
| **TLS Usage** | Optional (HTTP/1.1 can be used without TLS) | Encourages the use of TLS (HTTPS) for enhanced security, but not mandatory. |
| **Header Size Impact** | Large header size due to redundancy | Header compression significantly reduces the impact of header size on overall data transfer. |