

# World Wide Web

## HTTP / 2



STUDIEREN  
AUF HÖCHSTEM  
NIVEAU

Prof. Dr. Jürgen Anders, Hochschule Furtwangen  
Fakultät Digitale Medien

Weiterentwicklung von HTTP/1.1 führte 2015 zur Verabschiedung von **HTTP/2** als neuem Internet Standard (RFC 7540)

Nachfolger von HTTP/1.1 bringt viele Verbesserungen, wobei Hauptfokus auf der **Verbesserung der wahrgenommenen Geschwindigkeit** für den Endnutzer liegt

- **HTTP/2**
  - ist anders als Vorgänger ein **binäres Protokoll**
  - ermöglicht **Request Multiplexing**
  - kann auch **Header-Informationen komprimieren**
  - spezifiziert **Server Push**
  - ist **rückwärtskompatibel** zum HTTP/1.1-Vokabular
  - Methoden und Status Codes von HTTP/1.1 bleiben erhalten
  - kann das **Web sicherer** machen

## Binäres Protokoll:

HTTP/2 ist ein binäres Protokoll – alle früheren Versionen basierten auf Klartext (und waren menschenlesbar)

HTTP/2 führt neue Konzepte ein:

- **Stream:**
  - Virtueller Kanal innerhalb einer Verbindung, der bi-direktionale Nachrichten (Messages) transportiert
- **Message:**
  - Logische HTTP-Nachricht wie Request, Response, etc., die aus einem oder mehr Frames besteht
- **Frame:**
  - Kleinste Kommunikationseinheit, die einen speziellen Datentyp transportiert, z.B. HTTP-Header, Payload, etc.

Bekanntes Problem bei HTTP/1.1

- Download vieler Ressourcen von einem Host kann nur nacheinander, aber nicht parallel erfolgen

Behelfslösung bei HTTP/1.1: Domain Sharing

- Verschiedene Ressourcen einer Webseite werden unter verschiedenen Subdomains veröffentlicht (z.B. Bilder, JS, CSS), denn verschiedene Domains erlauben zusätzliche TCP-Verbindungen

**HTTP/2 führt Multiplexing ein**

- Erlaubt verflochtenen Abruf und Empfang multipler Ressourcen innerhalb einer TCP/IP-Verbindung
- Requests blockieren nicht mehr -> keine Notwendigkeit mehr, multiple TCP-Verbindungen zu mehreren Domainnamen aufzubauen

**HTTP/2 führt Header-Kompression ein (RFC 7541)**

- HTTP/1.1-Header müssen unkomprimiert sein; Kompression ist nur beim Nachrichtenteil erlaubt
- HTTP/2 erlaubt nun auch Header-Kompression. Dabei wird die Kompressionsmethode HPACK verwendet -> besonders interessant für Seiten mit großen Cookies (>1 MB), was heutzutage nicht mehr so utopisch ist
- HTTP/2 empfiehlt, den Nachrichtenteil nicht mehr zu komprimieren
  - Bei gzip und deflate gibt es Sicherheitsbedenken
  - Das Binärformat sorgt bereits für eine effiziente Komprimierung, weshalb in RFC 7540 von einer Komprimierung im Zuge der TLS-Verschlüsselung abgeraten wird.

## HTTP/2 führt Server-Side-Push ein

In HTTP/1.1 konnten Ressourcen nur durch **vollständige Request-Response-Zyklen** ausgeliefert werden

- Bekanntes Problem: Client macht ersten Request, parst das erhaltene HTML-Dokument und muss danach die (vielen) **eingebetteten Ressourcen** anfordern
- HTTP/2 führt **Server Push** ein: Auf ersten HTTP-Request hin kann Server gleich weitere Ressourcen in der Antwort mitschicken  
-> Vermeidung zusätzlicher Request-Response-Zyklen

Wie kann HTTP-Server die zu pushenden Ressourcen erkennen? **Genaue Implementierung ist hier noch unklar:**

- **Idee 1:** Autoren der Webanwendungen können Server Push explizit oder durch zusätzliche HTTP-Header veranlassen
- **Idee 2:** HTTP-Server können „lernen“, welche Ressourcen gepusht werden müssen

## HTTP/2 unterstützt Verschlüsselung

Im ursprünglichen Entwurf war **zwingend TLS-Verschlüsselung** vorgesehen

- HTTP/2-Spezifikation erzwingt Nutzung von TLS nicht
- Theoretisch sind auch HTTP/2-Nachrichten ohne Verschlüsselung erlaubt
- Aber die großen Browserhersteller (Google, Mozilla, Microsoft) erzwingen Benutzung von TLS in ihrer Implementierung...  
*„Firefox will only be implementing HTTP/2 over TLS – and so far that means for https:// schemed URLs. It does enforce the protocol's >= TLS 1.2 requirement – if a server negotiates HTTP/2 with a lower TLS version it is treated as a protocol error.“*

# World Wide Web

## HTTP / 2



STUDIERN  
AUF HÖCHSTEM  
NIVEAU

Prof. Dr. Jürgen Anders, Hochschule Furtwangen  
Fakultät Digitale Medien