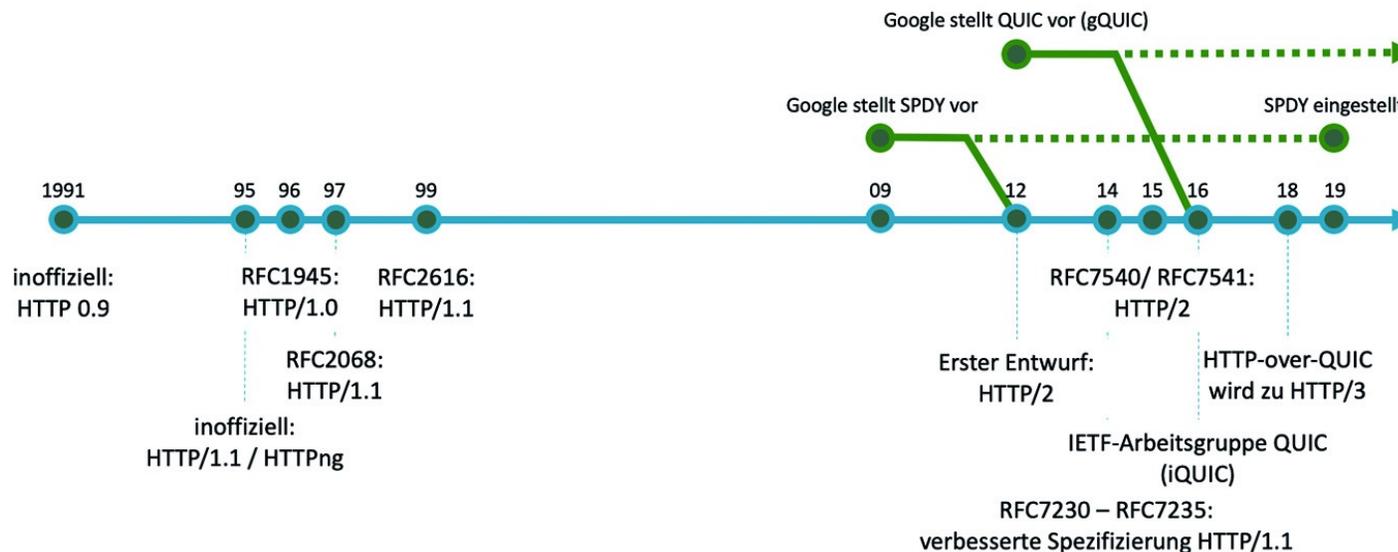


# HTTP/2 .... HTML5

Machen wir das Web schnell

# HTTP/2-Ziele

- Verbesserung der wahrgenommenen Latenzzeit
- Adressiert den "head of line blocking"
- Benötigt nicht mehrere Verbindungen
- Beibehaltung der Semantik<sup>1)</sup> von HTTP/1.1



<sup>1)</sup> Bedeutung, Inhalt

# Bedienbarkeit

| VERZÖGERUNG   | BENUTZERREAKTION           |
|---------------|----------------------------|
| 0 - 100 ms    | Sofort                     |
| 100 - 300 ms  | <i>Fühlt sich träge an</i> |
| 300 - 1000 ms | Die Maschine arbeitet...   |
| 1 s+          | Mentaler Kontextwechsel    |
| 10 s          | Ich komme später wieder... |

*Jakob Nielsen*

# Wie schnell sind Websites rund um die Welt?

## First Contentful Paint

The number of seconds from the time the navigation started until the page's primary content appears on the screen.

MEDIAN DESKTOP  
**2.5 seconds**  
▼10.7%

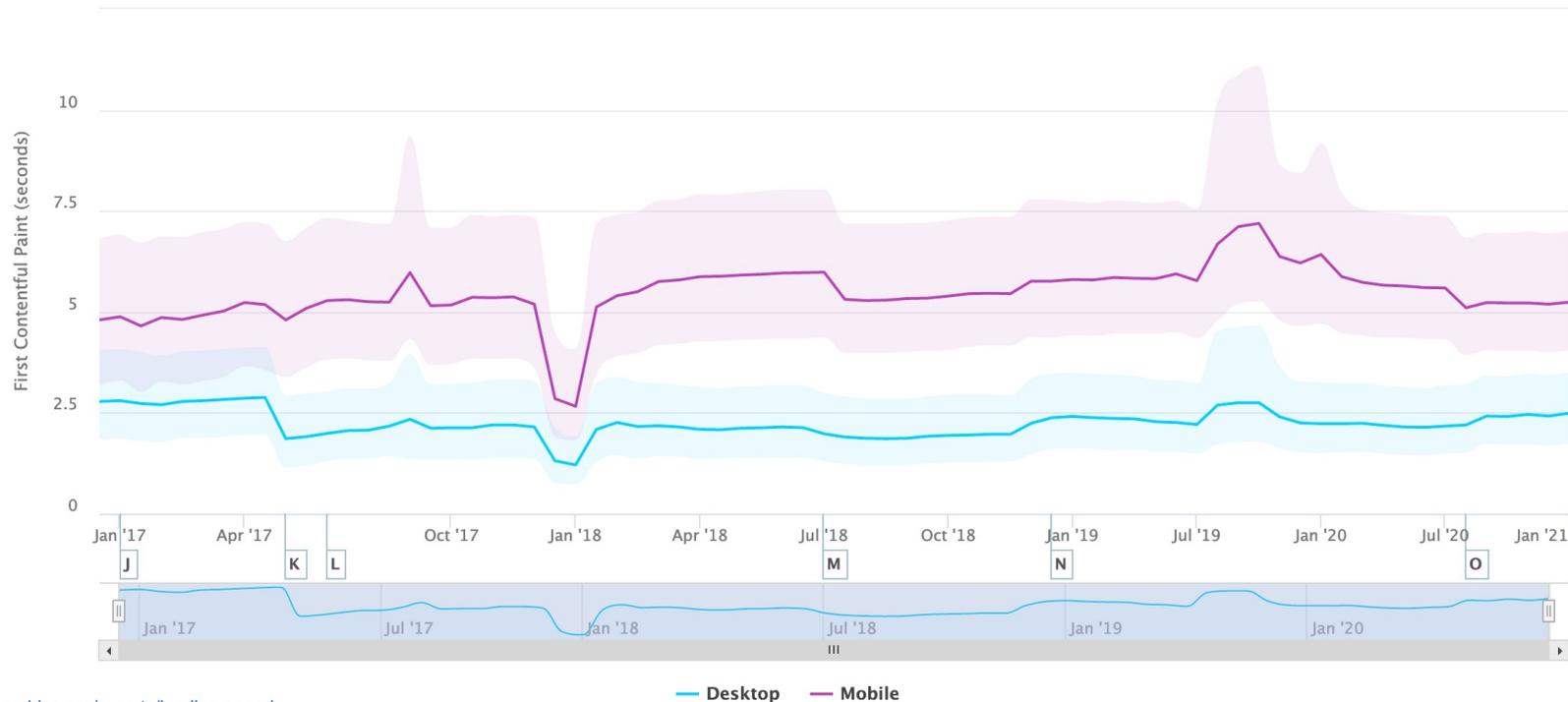
MEDIAN MOBILE  
**5.2 seconds**  
▲8.3%

### Timeseries of First Contentful Paint

Source: [httparchive.org](http://httparchive.org)

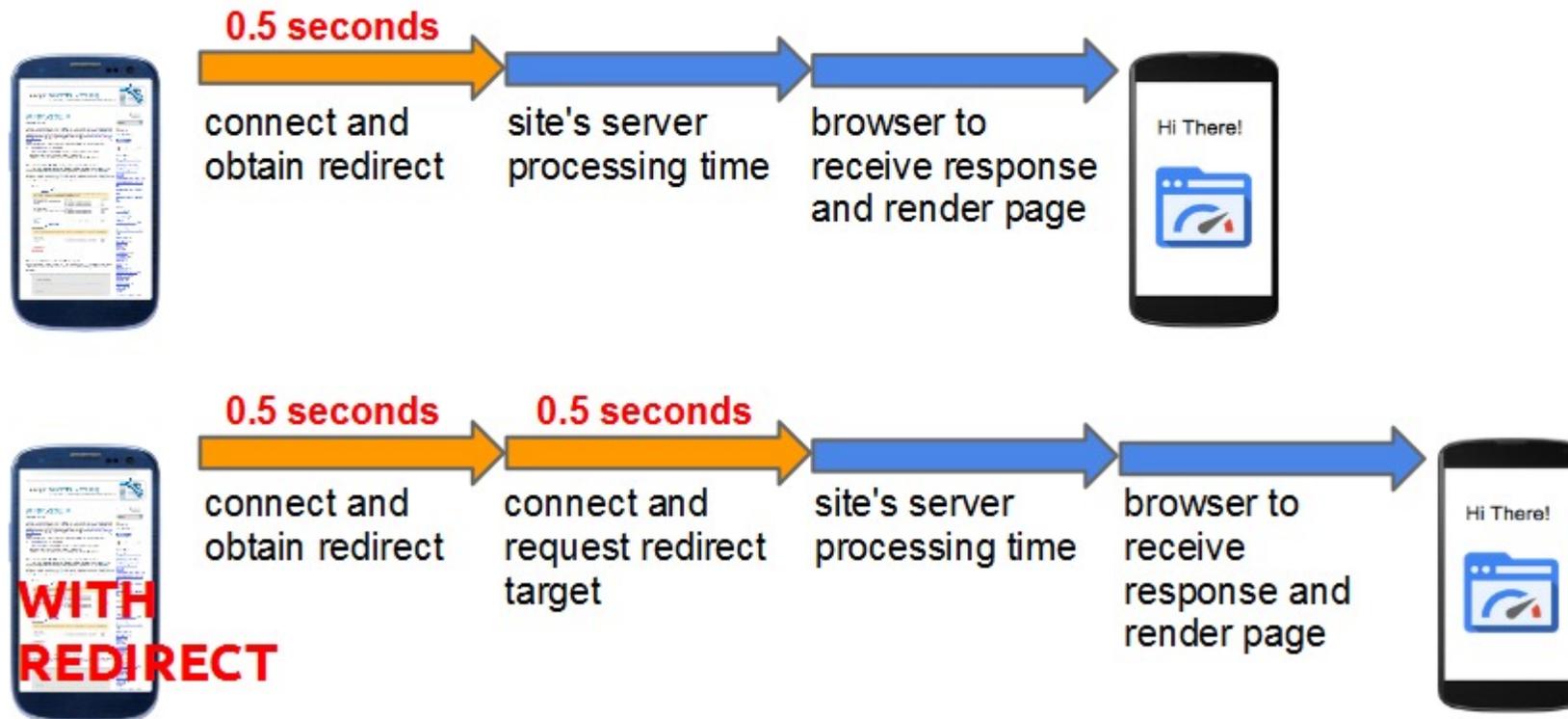
Zoom 1m 3m 6m YTD 1y 3y **All**

From Mar 15, 2012 To Jan 1, 2021



# Avoid landing page redirects (each costs 0.5 seconds minimum)

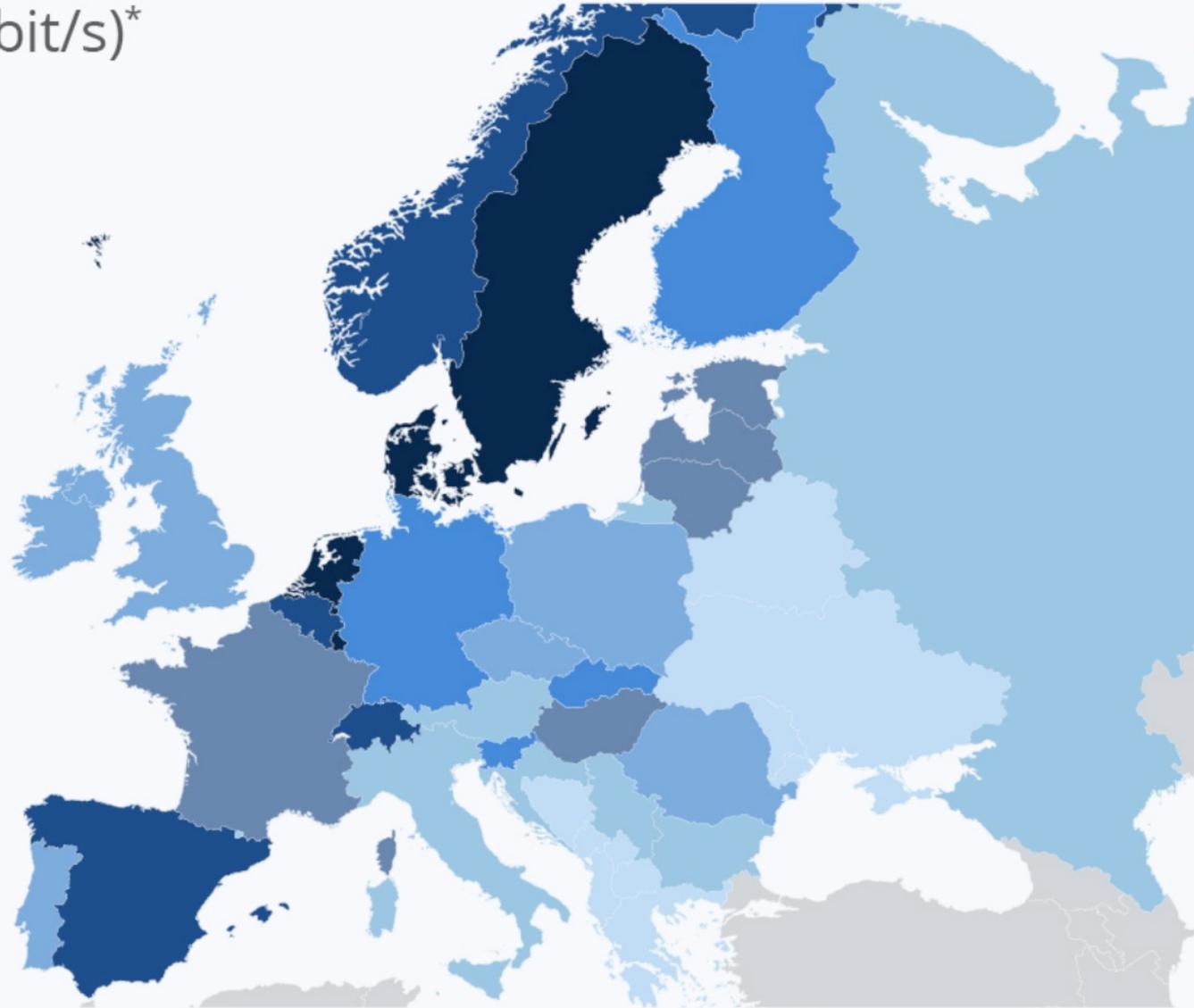
z. B.: [mysite.com](http://mysite.com) -> [m.mysite.com](http://m.mysite.com)



# Die Netzwerkbandbreite wird uns retten?

Richtig, oder? Oder vielleicht auch nicht...

# Durchschnittliche Breitband-Internetgeschwindigkeit in Europa (in Mbit/s)\*



\* Datenerhebung durch M-Lab, zwischen 9. Mai 2018 und 8. Mai 2019

Quellen: cable.co.uk, M-Lab

# Bandbreite

In Computernetzwerken ist die **Bandbreite** ein Maß für die Bitrate der verfügbaren oder verbrauchten Datenkommunikationsressourcen, ausgedrückt in Bits pro Sekunde oder einem Vielfachen davon (bit/s, kbit/s, Mbit/s, Gbit/s usw.).

Dies steht im Gegensatz zur Verwendung des Begriffs *Bandbreite* im Bereich der Signalverarbeitung für analoge Signalbandbreite, gemessen in Hertz.

# Latenz (Laufzeit)

Unter der **Laufzeit** versteht man die zeitliche Differenz zwischen dem Eintritt eines Signals in ein System und dem Austritt.

# RTT

Die **Round Trip Time** bzw. **Paketumlaufzeit** gibt die Zeit an, die ein Datenpaket in einem Rechnernetz benötigt, um von der Quelle zum Ziel und wieder zurück zu reisen

# Durchschnittliche RTT zu Google ist...

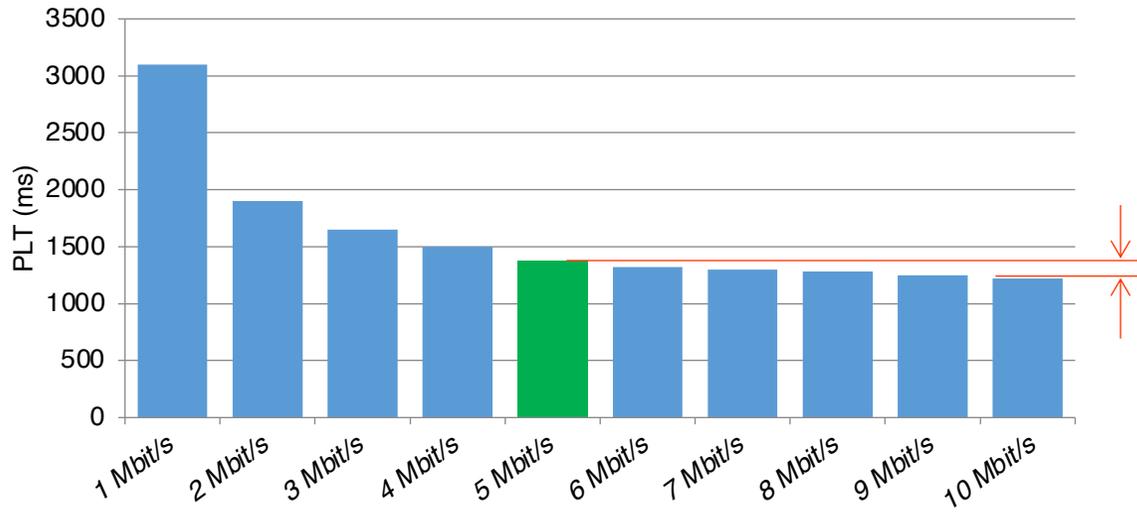
- Weltweit: **~100ms**
- US: **~50-60ms**

# Die Bandbreite spielt keine Rolle

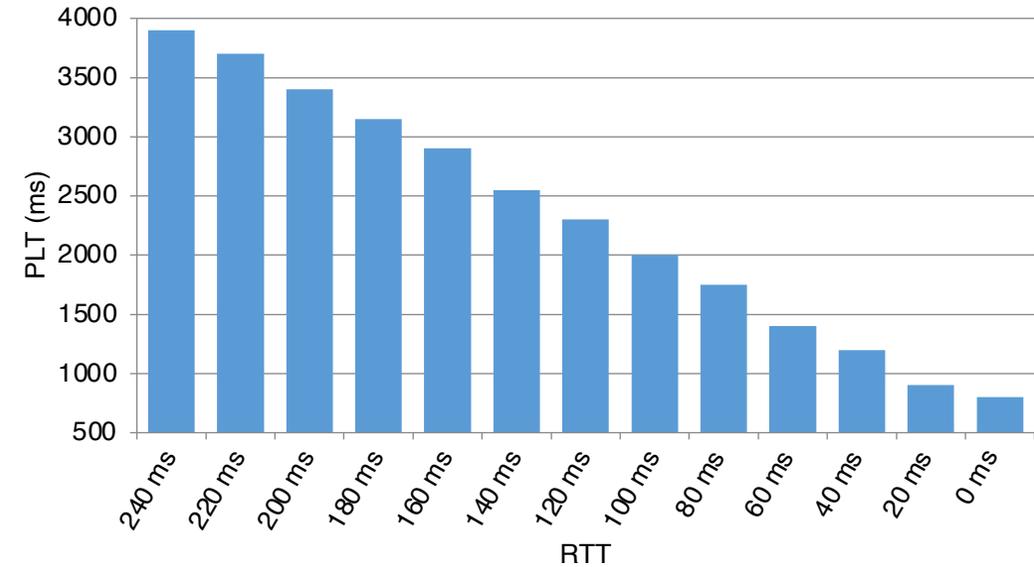
Es ist die Latenz, verdammt!

# PLT: Latenz vs. Bandbreite

## Latenz pro Band mit



## Seitenladezeit bei abnehmender RTT



Der durchschnittliche Haushalt in den USA verfügt über eine Verbindung mit **20 Mbit/s+**.

Ergo **würde der** Durchschnittsverbraucher in den USA durch ein Upgrade seines Anschlusses **keine** verbesserte PLT **sehen**.

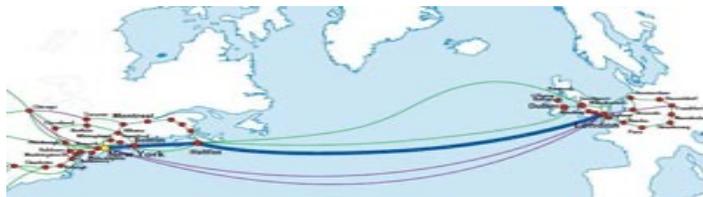
**Latenz ist der neue Leistungseingpass**

# Die Verbesserung der Bandbreite ist einfach ....

- Noch viele dunkle Fasern verfügbar
- 60 % der neuen Kapazität durch Upgrades
- "Einfach mehr Kabel verlegen" ...

# Die Verbesserung der Latenzzeit ist teuer... unmöglich?

- Begrenzt durch die Lichtgeschwindigkeit
- Wir sind bereits innerhalb eines kleinen konstanten Faktors
- Verlegen von **kürzeren** Leitungen!



**\$ 80.000.000 / ms**

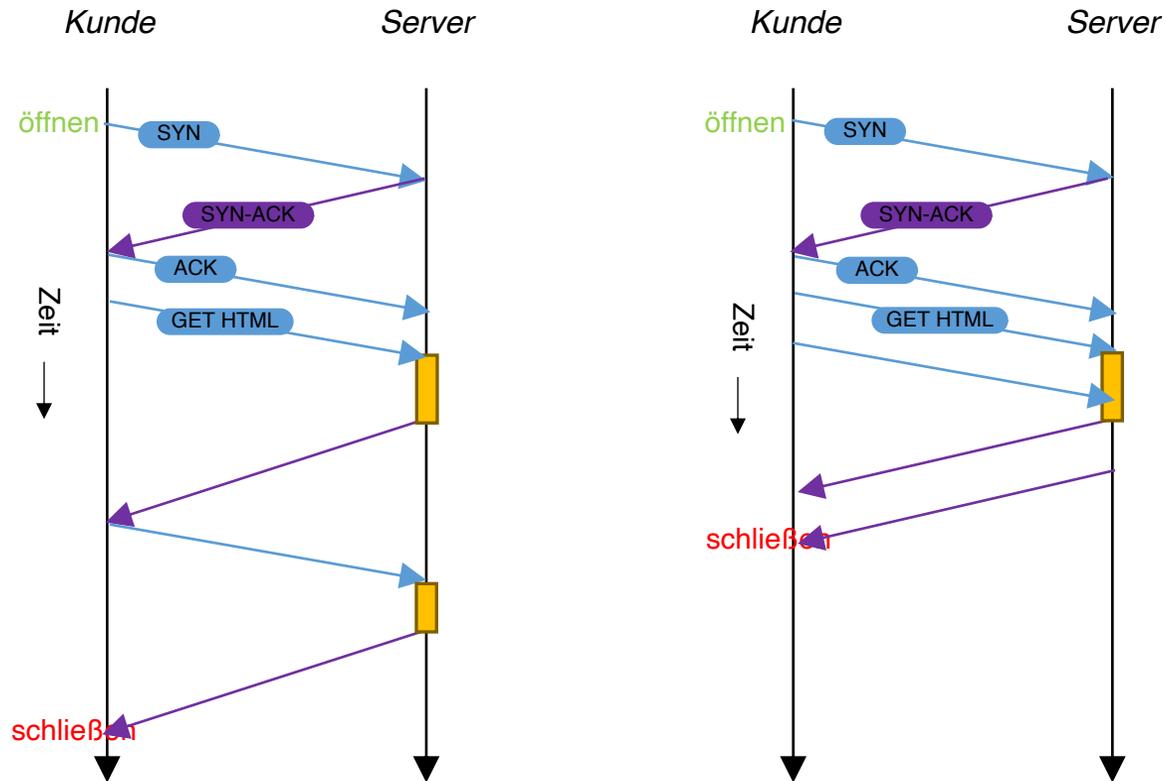
**Hibernian (2011): \$ 400.000.000**  
**Neues transatlantisches Kabel,**  
**das den Aktienhandel um**  
**5 Millisekunden verkürzt**

# Warum ist die Latenz das Problem?

*Erinnern wir uns an diese HTTP-Sache...*

# HTTP hat kein Multiplexing!

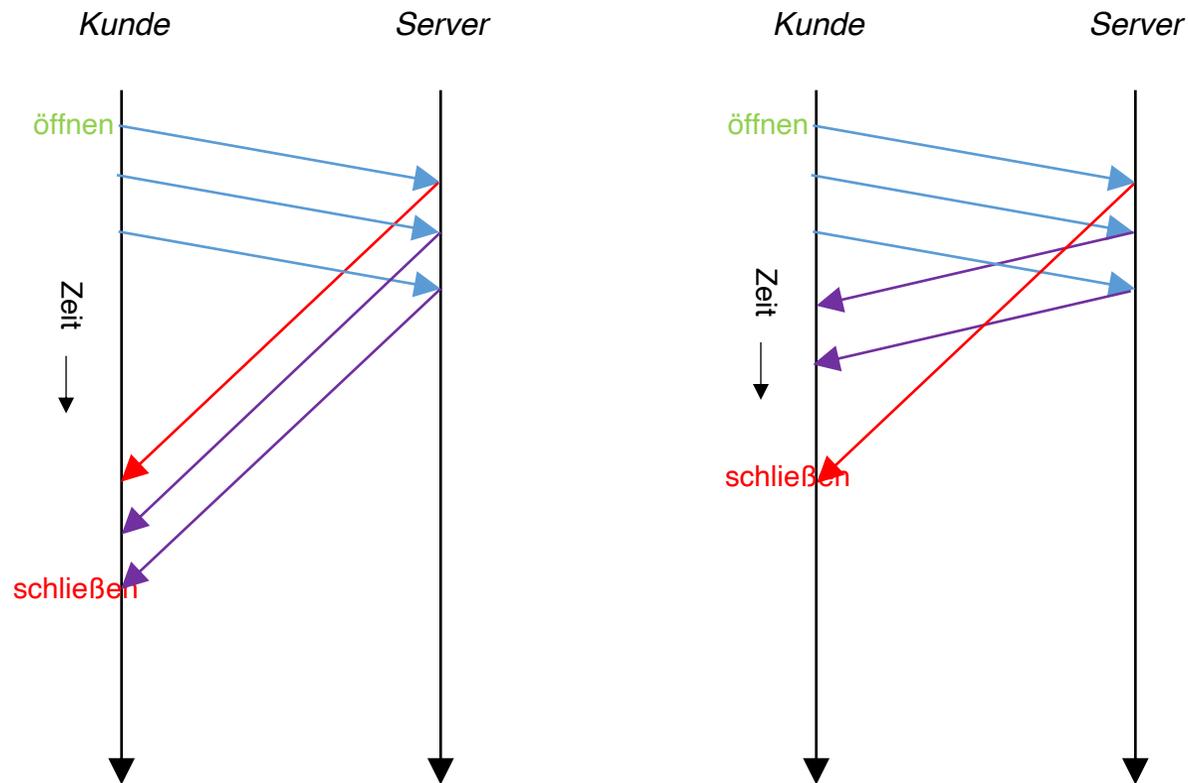
## http-Anfrage-Pipelining



# HOLB (Head of Line Blocking)

pipelining  
head of line blocking mit http/2

http-Multiplexing



Wenn die **erste** Antwort mehr Zeit benötigt,  
werden die folgenden Antworten entsprechend  
empfangen

## Multiplexing

Die HTTP-Anfragen können  
unabhängig von einer  
vorherigen Antwort gestellt  
werden, die Reihenfolge  
spielt keine Rolle.

# Öffnen wir mehrere TCP-Verbindungen!!!

New York Times: <https://www.nytimes.com/>

**10** verschiedene Hosts  
137 Anfragen  
3720 kB Daten

=

|                                                                                                                  |                       |      |           |
|------------------------------------------------------------------------------------------------------------------|-----------------------|------|-----------|
|  18UP-Religion-thumbStandard.jpg  | graphics8.nytimes.com | jpeg | 3,90 KB   |
| recommendationsModule.js                                                                                         | js.nytimes.com        | js   | 22,95 KB  |
| bidirectionalNumericStepper.css                                                                                  | graphics8.nytimes.com | css  | 1,31 KB   |
| mtr.js                                                                                                           | js.nytimes.com        | js   | 29,08 KB  |
| nytModal.css                                                                                                     | css.nytimes.com       | css  | 4,50 KB   |
| build.js                                                                                                         | js.nytimes.com        | js   | 35,79 KB  |
| bottom.js                                                                                                        | js.nytimes.com        | js   | 58,53 KB  |
| meter.js?callback=jQuery17106812980242824992_1400...                                                             | meter-svc.nytimes.com | js   | 0,25 KB   |
|  search_button40x19.gif           | graphics8.nytimes.com | gif  | 0,43 KB   |
|  nytlogo379x64.gif                | i1.nytimes.com        | gif  | 4,45 KB   |
| miq8eej-d.css?3bb2a6e53c9684ffdc9a9ef6195b2a6262...                                                              | typeface.nytimes.com  | css  | 324,54 KB |
| effects.js                                                                                                       | js.nytimes.com        | js   | 37,57 KB  |
| embeddedSlideshowStandard.js                                                                                     | js.nytimes.com        | js   | 36,60 KB  |
|  vertical_rule.gif               | css.nytimes.com       | gif  | 0,04 KB   |
|  hp-map-promo-v3.png            | graphics8.nytimes.com | png  | 45,82 KB  |
|  highway_icon_sm.png            | graphics8.nytimes.com | png  | 1,55 KB   |
|  sprite-nyt5-masthead-icons.png | css.nytimes.com       | png  | 11,43 KB  |
|  sprite-nyt5-icons-v2.png       | css.nytimes.com       | png  | 19,66 KB  |
| hierarchical.jsonp?callback=jsonTaxonomyCallback&...                                                             | json8.nytimes.com     | js   | 79,00 KB  |
| meter.js?callback=jQuery17106812980242824992_1400...                                                             | meter-svc.nytimes.com | js   | 0,25 KB   |

**2018**

**http/1.1**

**3,4 Sekunden**

# The New York Times

Inspector Konsole Debugger Netzwerkanalyse Stilbearbeitung Laufzeitanalyse Speicher Web-Speicher

Adressen durchsuchen | Alles HTML CSS JS XHR Schriften Grafiken Medien WebSockets Sonstiges  Cache deaktivieren Keine Drosselung

| Status | Meth... | Host                 | Datei               | Initiator              | Typ  | Übertragen | Größe   |
|--------|---------|----------------------|---------------------|------------------------|------|------------|---------|
| 200    | GET     | static01.nyt.com     | icon-search.svg     | img                    | svg  | 1,17 KB    | 590 B   |
| 200    | GET     | static01.nyt.com     | icon-search.svg     | keepassxc-browse...    | svg  | 1,17 KB    | 590 B   |
| 200    | GET     | static01.nyt.com     | vaccine_summary.js  | /:1684 (fetch)         | json | 1,24 KB    | 990 B   |
| 200    | GET     | static01.nyt.com     | summary.json        | dashboard.js:1 (fet... | json | 1,24 KB    | 990 B   |
| 200    | GET     | static01.nyt.com     | summary.json        | fetch                  | json | 1,24 KB    | 990 B   |
| 200    | GET     | static01.nyt.com     | index2.js           | script                 | js   | 1,34 KB    | 850 B   |
| 200    | GET     | static01.nyt.com     | index2.js           | script                 | js   | 1,34 KB    | 850 B   |
| 200    | GET     | content.api.nytim... | current-and-seven-c | main-31c22c0489...     | json | 1,38 KB    | 2,29 KB |
| 200    | GET     | www.nytimes.com      | geoip               | fetch                  | json | 1,44 KB    | 244 B   |
| 200    | GET     | www.nytimes.com      | geoip               | godzown.js:1 (fetch)   | json | 1,44 KB    | 244 B   |
| 200    | GET     | static01.nyt.com     | godzown.js          | script                 | js   | 1,45 KB    | 1,43 KB |
| 200    | GET     | static01.nyt.com     | godzown.js          | script                 | js   | 1,45 KB    | 1,43 KB |
| 200    | GET     | meter-svc.nytim...   | meter.js?sourceApp: | main-31c22c0489...     | json | 1,53 KB    | 511 B   |
| 200    | GET     | static01.nyt.com     | i18n.js             | script                 | js   | 1,59 KB    | 1,35 KB |
| 200    | GET     | static01.nyt.com     | i18n.js             | script                 | js   | 1,59 KB    | 1,35 KB |

148 Anfragen 14,69 MB / 9,41 MB übertragen Beendet: 4,45 min DOMContentLoaded: 693 ms load: 1,96 s

Kopfzeilen Cookies Anfrage Antwort

Kopfzeilen durchsuchen Blockieren Erneut send...

GET https://static01.nyt.com/newsgraphics/2020/03/16/coronavirus-maps/13c2c00f93718a543fd9012160d620d69ecbfcae/images/icon-search.svg

Status 200 OK  
Version HTTP/2  
Übertragen 1,17 KB (590 B Größe)  
Referrer Policy strict-origin-when-cross-origin

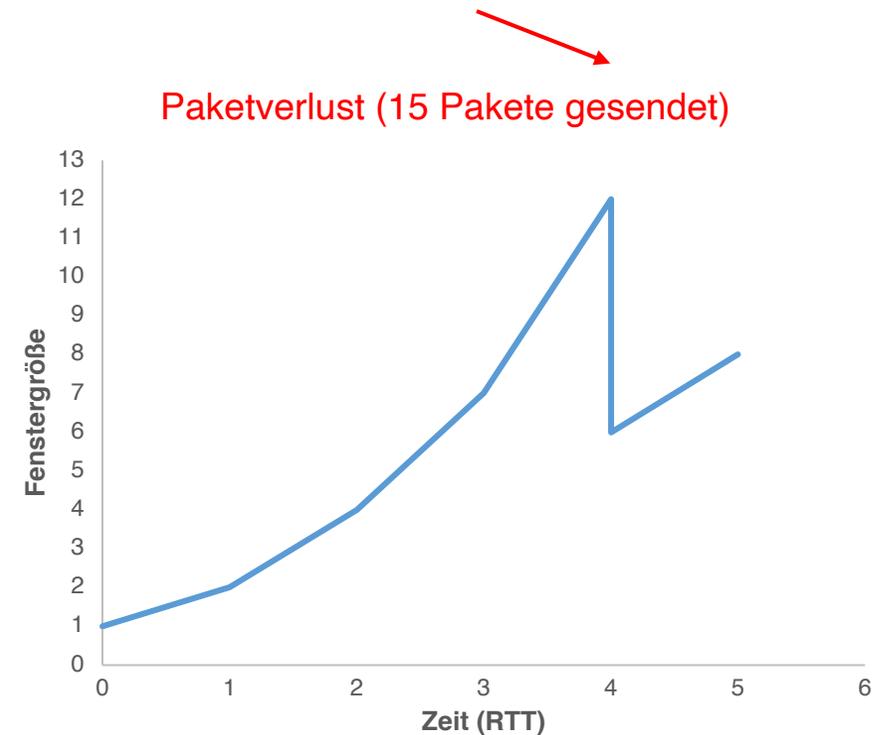
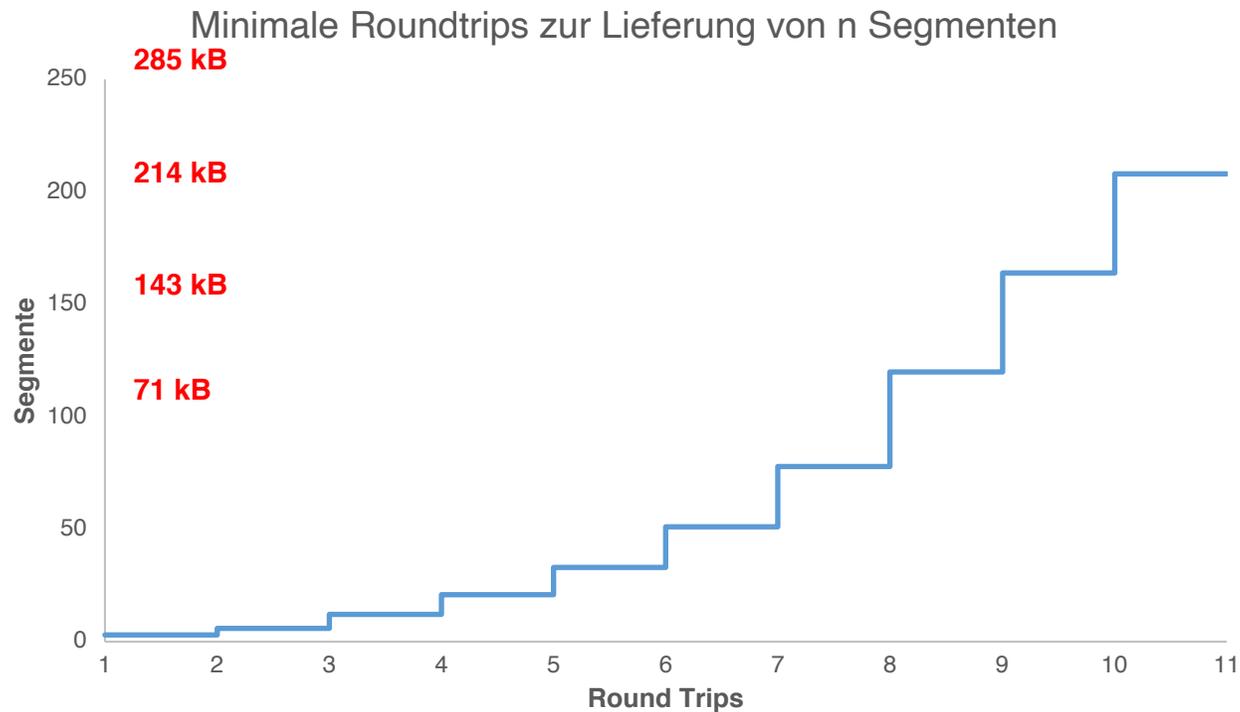
Antwortkopfzeilen (856 B) Unformatiert

- accept-ranges: bytes
- access-control-allow-origin: \*
- age: 3
- cache-control: max-age=5
- content-encoding: gzip
- content-length: 339
- content-type: image/svg+xml
- date: Tue, 30 Mar 2021 12:30:13 GMT
- etag: "1e8befd045e58abe2c32de3b6f5d2c98"

# TCP Congestion Control & Avoidance...

TCP ist dafür ausgelegt, das Netzwerk zu sondieren, um die verfügbare Kapazität herauszufinden

## TCP Slow Start - Funktion, kein Fehler



# Anfragen und Transfergröße

| Resource Type | Requests | Transfer Size |
|---------------|----------|---------------|
| Total         | 86       | 1,171.5 KiB   |
| Script        | 35       | 333.9 KiB     |
| Image         | 20       | 290.7 KiB     |
| Font          | 9        | 253.5 KiB     |
| Stylesheet    | 16       | 227.2 KiB     |
| Document      | 3        | 65.4 KiB      |
| Other         | 3        | 0.9 KiB       |
| Media         | 0        | 0 KiB         |
| Third-party   | 20       | 266.4 KiB     |

# HTTP-Archiv sagt....

Auswertungen für TOP 100 Websites (April 2017)

## Total Transfer Size & Total Requests



# HTTP/2 Requests

The percent of all requests in the crawl using HTTP/2.

DESKTOP **66.9%** ▲106.5%  
MOBILE **67.3%** ▲103.3%

## Timeseries of HTTP/2 Requests

Source: httparchive.org

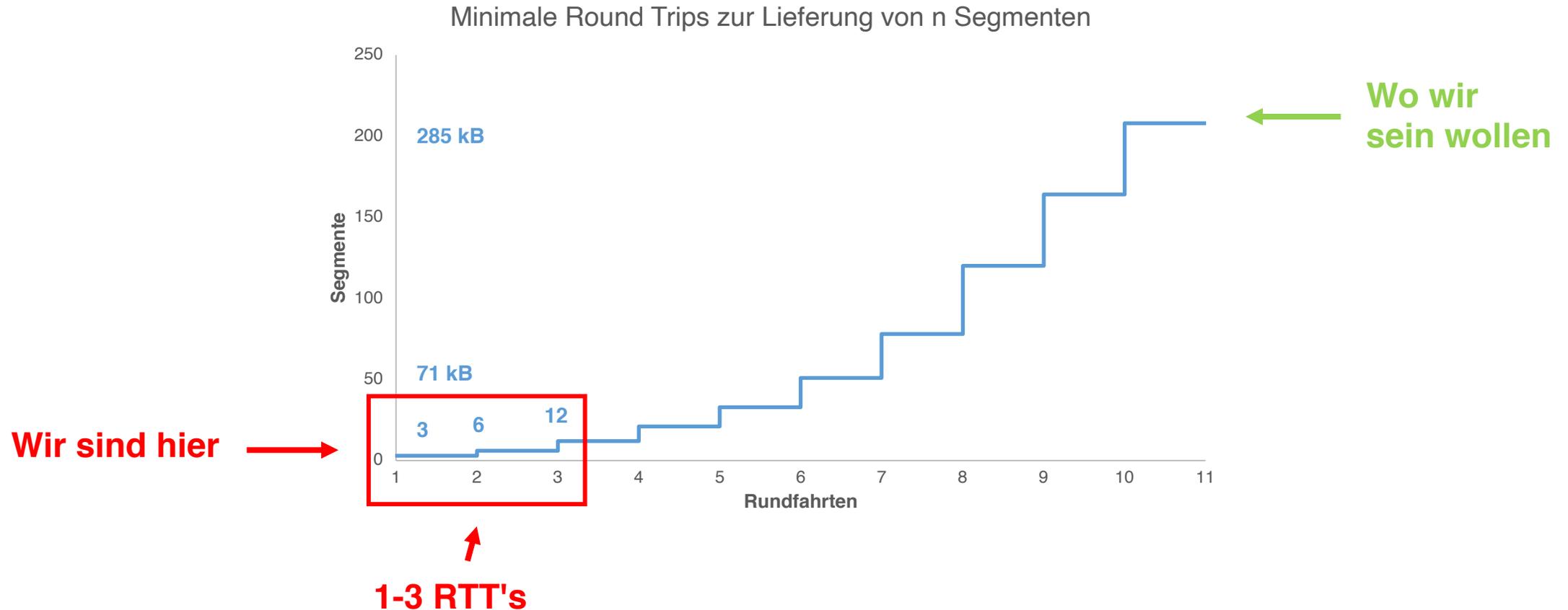


<https://httparchive.org/reports/state-of-the-web#h2>

# HTTP-Archiv sagt....

1,171 kb, 86 Anfragen, ~14kb **pro Anfrage!**

DER MEISTE HTTP-VERKEHR BESTEHT AUS KLEINEN, STOSSARTIGEN TCP-STRÖMEN



# Beispiel:

facebook

428 ms zum Herunterladen von 40 kB Daten!

Das sind 0,8 Mbit und es wird nicht schneller, egal wie breitbandig die Leitung ist (100Mbit)

| ✓    | Methode | Datei           | Host                    | Typ   | Größe    | Laufzeit | Netzwerkanalyse                            |
|------|---------|-----------------|-------------------------|-------|----------|----------|--------------------------------------------|
| 200  | GET     | /               | www.facebook.com        | html  | 42,71 KB | → 428 ms | Kopfzeilen   Cookies   Parameter   Antwort |
| POST | POST    | bz              | www.facebook.com        | plain | 0 KB     | → 0 ms   | DNS-Auflösung: → 3 ms                      |
| 200  | GET     | WhbVBdzLYIG.css | fbstatic-a.akamaihd.net | css   | 20,10 KB | → 63 ms  | Verbindungsaufbau: → 16 ms                 |
| 200  | GET     | bOZ26aPaCfu.css | fbstatic-a.akamaihd.net | css   | 96,65 KB | → 63 ms  | Senden: → 22 ms                            |
| 200  | GET     | x3bsMJyVkPp.css | fbstatic-a.akamaihd.net | css   | 39,72 KB | → 58 ms  | Warten: → 223 ms                           |
| 200  | GET     |                 |                         |       | 70,00 KB |          | Empfangen:                                 |

# **Also, was soll ein Entwickler tun?**

*Reparieren wir HTTP 1.1! Verwenden wir in der Zwischenzeit HTTP/2...*

# Header-Komprimierung ...

Benutzer-Agent: 109 Bytes  
Sprache: 35 Byte  
Akzeptieren: 74 Bytes

**typische Header-Größen von  
700-800 Byte Overhead sind  
üblich**

Minimum IPv4 + TCP-Header: 40 Bytes  
Minimum IPv6 + TCP-Header: 60 Bytes

MTU = 1500 (Ethernet)

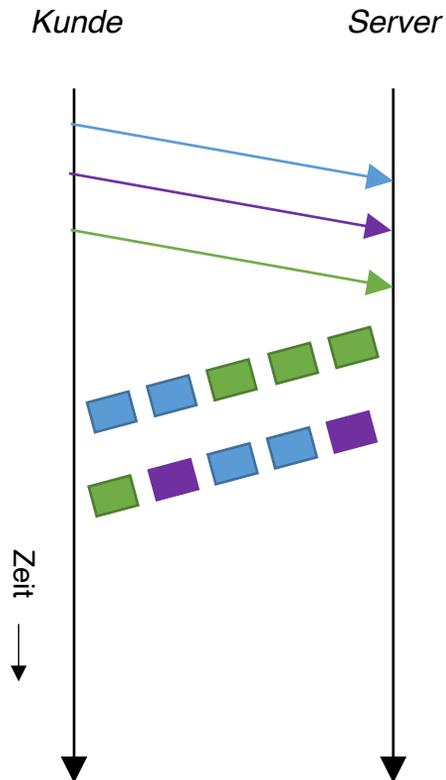
MSS IPv4 = 1500 - 40 = 1460 Bytes

MSS - Header = **700** Bytes

## ▼ Request Headers

```
:host: www.google.at
:method: GET
:path: /
:scheme: https
:version: HTTP/1.1
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
accept-encoding: gzip,deflate,sdch
accept-language: de-DE,de;q=0.8,en-US;q=0.6,en;q=0.4
cache-control: max-age=0
cookie: PREF=ID=f5a1baf9d7d88d7c:U=1ef9536cd1e0298f:FF=0:TM=1398599679:LM=1398599679:S=xxBkOAKSYI6fdobE; NID=67=HYokDmhN3LTUA
YSwhgaxyGehLE8bF9ij3bM0_8uAjMLQtHYK
dnt: 1
user-agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/34.0.1847.137 Safari/537.36
x-client-data: COy1yQEII17bJAQiktskBCKm2yQEIXLbJAQiehsoBCKSIygEIuYjKAQih1MoBCKmUygE=
```

# HTTP/2 in Aktion



- Vollständiges Request & Response-Multiplexing
- Mechanismus zur Priorisierung von Anfragen
- Viele kleine Dateien? Kein Problem
- Höhere TCP-Fenstergröße
- Effizientere Nutzung der Server-Ressourcen
- TCP Fast-retransmit für schnellere Wiederherstellung

## Anti-Patterns

- Domänenaufteilung  
*Jetzt müssen wir entschärfen - puh!*



# HTTP/2-Server-Push

**Prämisse:** Server kann Ressourcen zum Client schieben

- ***Besorgnis: aber ich will die Daten nicht! Aufhören!***
  - Client kann SYN\_STREAM abbrechen, wenn er die Ressource nicht hat
- Ressource geht in den Cache des Browsers (keine Client-API)

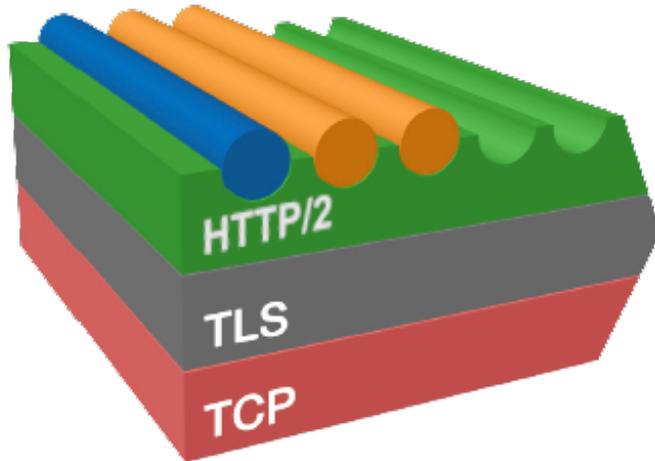
**Kurzmeldung:** Google verwendet bereits "Server Push"

- Heute nennt man das "Inlining"
- Inlining funktioniert für eindeutige Ressourcen, bläht ansonsten Seiten auf

**Erweiterter Anwendungsfall:** Forward Proxy (ala Amazons Silk)

- Proxy hat volle Kenntnis über Ihren Cache, kann Daten intelligent an den Client pushen

# Verschlüsseln Sie alle Dinge!!!



## HTTP/2 läuft über TLS

- Philosophische Gründe
- Politische Gründe
- **Pragmatische + Einsatzgründe - Bing!**

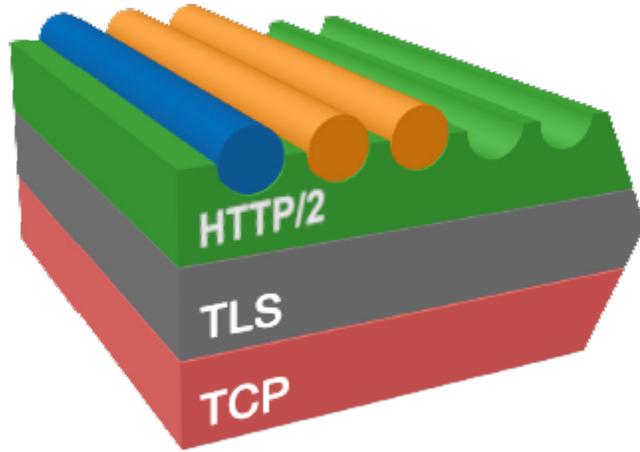
## Beobachtung: zwischengeschaltete Proxies sind im Weg

- Manche tun es absichtlich, viele unabsichtlich
- *Bsp: Antivirus / Packet Inspection / QoS / ...*

## SDHC / WebSocket: Kein TLS funktioniert... in 80-90% der Fälle

- 10 % der Zeit scheitern Dinge ohne erkennbaren Grund
- In der Praxis laufen alle großen Webservice (WS)-Einsätze als WSS

# Aber ist TLS nicht *langsam*?



## CPU

"Auf unseren Produktions-Frontend-Maschinen **macht SSL/TLS für weniger als 1% der CPU-Last**, weniger als 10KB Speicher pro Verbindung und weniger als 2 % des Netzwerk-Overheads."

- Adam Langley (Google)

## Latenzzeit

- TLS Next Protocol Aushandlung
  - Protokoll-Aushandlung als Teil des TLS-Handshakes
- TLS-Fehlstart
  - reduzieren Sie die Anzahl der RTTS für den Full Handshake von zwei auf einen
- Sitzung fortsetzen, ...

# Wer unterstützt HTTP/2?

- **Chrome**, schon immer...
  - Chrome auf Android + iOS
- **Firefox 13+, Safari, Edge, ....**

## Server

- Apache
- nginx
- tomcat
- Nodejs
- ...

## Third party

- Twitter
- Wordpress
- Facebook\*
- Akamai
- ...

## Alle Google-Services

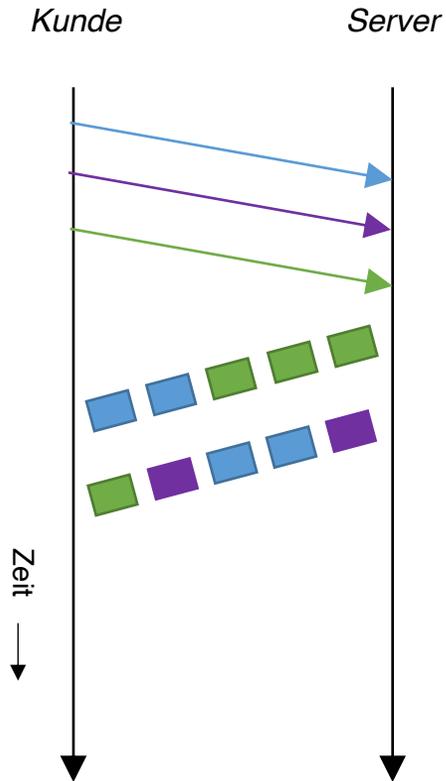
- Suche, GMail, Docs
- GAE + SSL-user
- ...

**Aber halt, es gibt einen Haken!**

*es gibt immer ein "gotcha"...*

# ~~HTTP Head of Line Blocking~~

# -> TCP Line Blocking



TCP: ordnungsgemäße, zuverlässige Zustellung...  
***Was ist, wenn ein Paket verloren geht?***

## **~1~2% Paketverlustrate**

- CWND's werden zerhackt
- Fast-Retransmit hilft, aber...
- HTTP/2-Blockaden

Auch hohe RTT-Links sind ein Problem

- Traffic Shaping
- ISP's entfernen das dynamische window scaling!

[linux: tracebox]

*Etwas zum Nachdenken...*

HTTP  HTTPS

**0.960 s**

79% faster than HTTP

Aufrufen -> <http://www.httpvshttps.com/>

# HTTP/3?

|           |                 |                 |
|-----------|-----------------|-----------------|
| Anwendung | HTTP            | HTTP/3          |
|           | TLS             | QUIC            |
| Transport | TCP             | UDP             |
| Internet  | IP              | IP              |
| Netzwerk  | Ethernet, Wi-Fi | Ethernet, Wi-Fi |

## HTTP/2 vs. HTTP/3: Gemeinsamkeiten und Unterschiede

### Unterschiede:

- HTTP/3 baut im Gegensatz zu HTTP/2 auf **UDP statt auf TCP** auf
- Durch eine **integrierte TLS-1.3-Verschlüsselung** verzichtet HTTP/3 auf eine zusätzliche Verschlüsselungsanfrage (Handshakes) auf der TLS-Ebene und umgeht so unnötige Sicherheitsabfragen.
- **Fehlerkorrektur**, die bereits auf QUIC-Ebene erfolgt (Quick UDP Internet Connections)

### Gemeinsamkeiten:

- Beide Protokolle nutzen Header-Kompression, HTTP/3 löst jedoch die an eine Paketreihenfolge gebundene HTTP/2 HPACK-Kompression durch QPACK ab.
- HTTP/3 unterstützt wie HTTP/2-Server-Pushs, also das beschleunigte Senden von CSS- und JavaScript-Daten, die der Browser zum Darstellen einer Seite sowieso benötigt.
- Beide Protokolle nutzen Anfrage-/Antwort-Multiplexing, also das parallele Streamen von Daten verschiedener Ressourcen.
- Stream-Priorisierung sorgt bei beiden Protokollen dafür, dass Seiteninhalte priorisiert geladen werden, ohne auf die Fertigstellung von weiteren Anfragen zu warten.