

Von den Bedürfnissen eines Webserver

Amadeus Alfa, chemmedia AG



WiFi

DSL

Vodafone

DE
CI
X

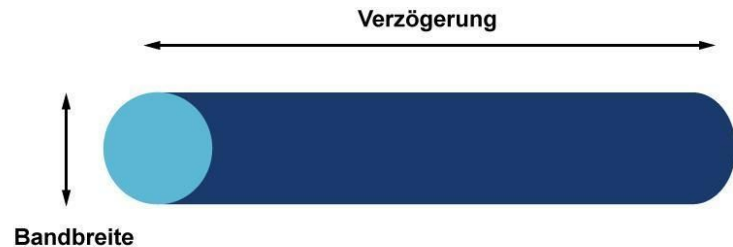
OVH



```
Shell
deus-apfel:Home alfa$ ping -c3 www.google.de
PING www.google.de (216.58.213.195): 56 data bytes
64 bytes from 216.58.213.195: icmp_seq=0 ttl=55 time=44.274 ms
64 bytes from 216.58.213.195: icmp_seq=1 ttl=55 time=69.347 ms
64 bytes from 216.58.213.195: icmp_seq=2 ttl=54 time=48.505 ms

--- www.google.de ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 44.274/54.042/69.347/10.959 ms
deus-apfel:Home alfa$
```

```
Shell
Last login: Tue Aug 15 20:06:12 on ttys000
deus-apfel:Home alfa$ ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1): 56 data bytes
64 bytes from 192.168.2.1: icmp_seq=0 ttl=64 time=28.464 ms
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=18.092 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=145.080 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=63.260 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=14.117 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=64 time=61.506 ms
64 bytes from 192.168.2.1: icmp_seq=6 ttl=64 time=24.847 ms
64 bytes from 192.168.2.1: icmp_seq=7 ttl=64 time=9.351 ms
64 bytes from 192.168.2.1: icmp_seq=8 ttl=64 time=89.550 ms
64 bytes from 192.168.2.1: icmp_seq=9 ttl=64 time=26.102 ms
64 bytes from 192.168.2.1: icmp_seq=10 ttl=64 time=18.309 ms
64 bytes from 192.168.2.1: icmp_seq=11 ttl=64 time=49.395 ms
64 bytes from 192.168.2.1: icmp_seq=12 ttl=64 time=7.278 ms
64 bytes from 192.168.2.1: icmp_seq=13 ttl=64 time=80.070 ms
^C
--- 192.168.2.1 ping statistics ---
14 packets transmitted, 14 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 7.278/45.387/145.080/37.772 ms
deus-apfel:Home alfa$
```



```
Shell
deus-apfel:Home alfa$ traceroute -n www.google.de
traceroute to www.google.de (216.58.213.195), 64 hops max, 52 byte packets
 1 192.168.2.1 78.534 ms 21.134 ms 12.641 ms
 2 192.168.0.1 14.609 ms 67.196 ms 19.805 ms
 3 83.169.183.129 17.988 ms 83.840 ms 26.757 ms
 4 88.134.237.223 21.076 ms 77.029 ms 21.830 ms
 5 88.134.235.66 94.889 ms 26.005 ms 29.083 ms
 6 88.134.202.69 63.534 ms
   88.134.235.23 27.800 ms 24.768 ms
 7 72.14.212.140 69.270 ms 38.888 ms 78.894 ms
 8 108.170.241.236 45.909 ms
   108.170.241.204 71.950 ms
   108.170.241.140 36.672 ms
 9 216.239.41.209 84.516 ms
   216.239.42.213 34.296 ms
   216.239.42.170 93.343 ms
10 72.14.237.98 62.839 ms
   72.14.236.18 52.158 ms
   72.14.237.98 37.041 ms
11 108.170.232.195 82.218 ms
   216.239.50.185 38.939 ms
   209.85.244.59 83.321 ms
12 72.14.233.217 45.563 ms 71.976 ms
   209.85.249.26 40.643 ms
13 209.85.247.15 75.109 ms 38.091 ms 82.871 ms
14 216.58.213.195 38.026 ms 86.842 ms 124.970 ms
deus-apfel:Home alfa$
```

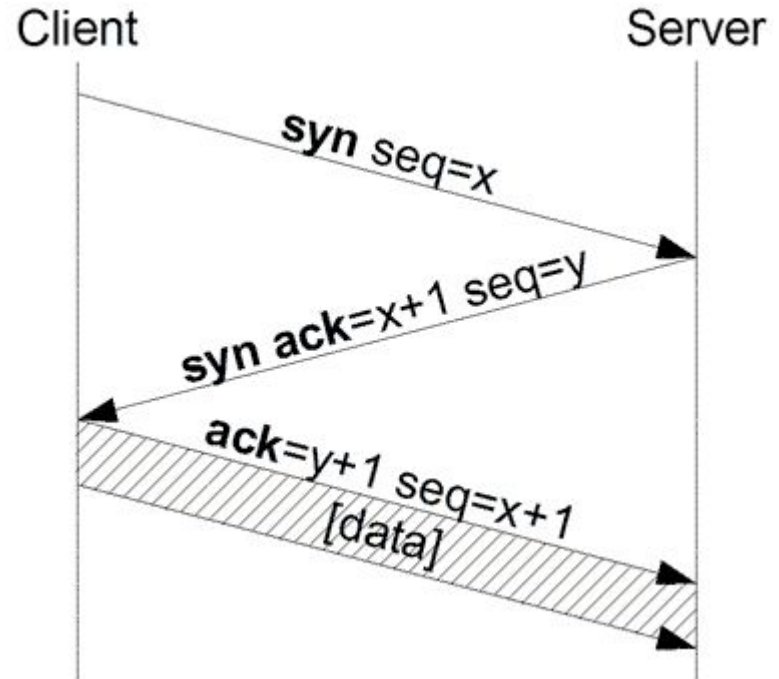


Round-Trip-Time minimieren

- Auswahl des ISPs/Backbones
- CDN-Infrastruktur nutzen
- Webapplikation

Achtung:

- TCP ist paketerorientiert
- RTT-Verhalten ist oft asymmetrisch

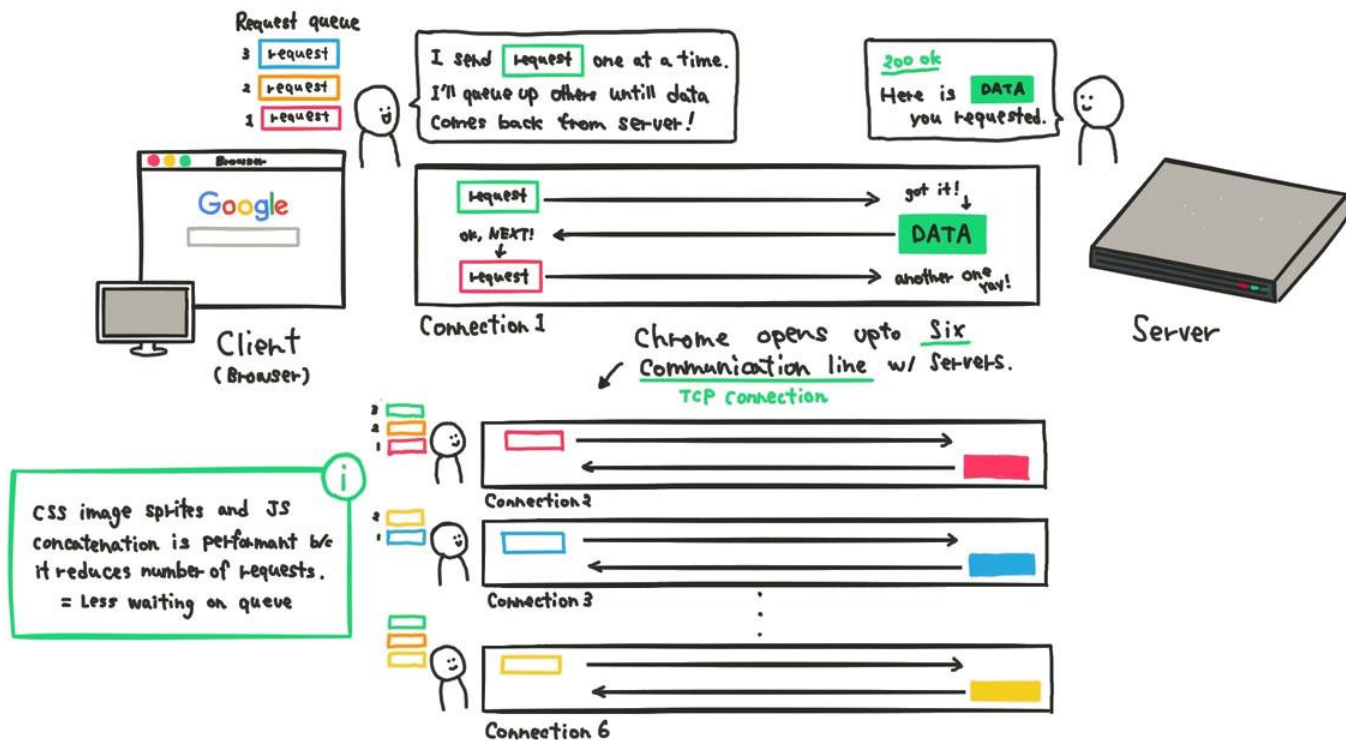


TCP (Transmission Control Protocol)

- Neue Verbindungen sind teuer (DNS, 3-Handshake, SSL, Retransmission, usw.) → Verbindung wiederverwenden
- Kapazität optimal ausnutzen (siehe “BDP”)
- Double-Buffer (parallele Requests) nutzen
- benötigte Ressourcen zeitig anfordern

Protocol (Pin HTTP) is a Communication rule between Client (browser) & Server.

HTTP 1.x



HTTP 1.1 Pipelining Support

- Internet Explorer: **Nein**
- Mozilla: **Ja, standardmäßig deaktiviert**
- Chrome: **Nein (wurde ausgebaut)**
- Safari: **Nein**
- Opera: **Ja**

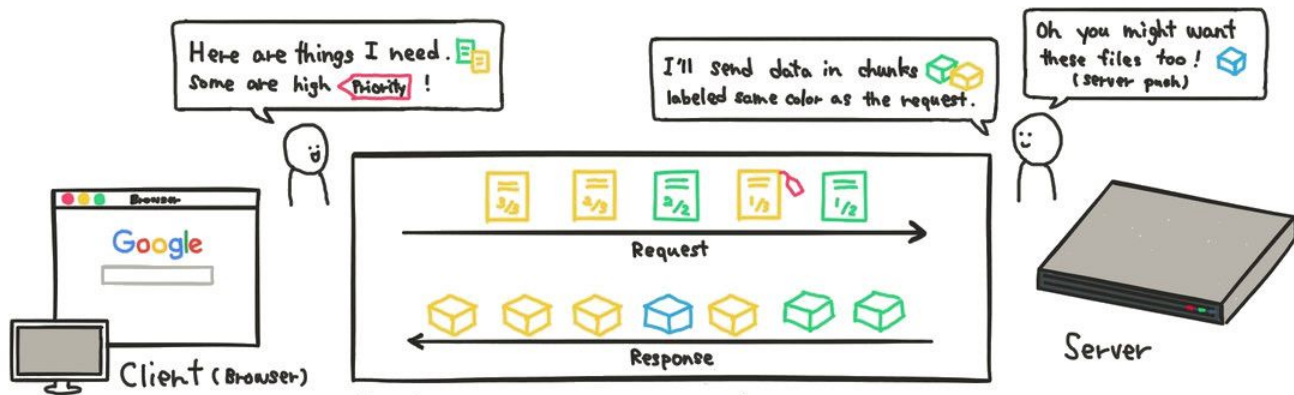
HTTP 1.1 Pipelining .. doch nicht

- (einige) Webserver können kein Pipelining → Heuristik-Magie
- (transparente) Proxy-Server stören
- Head-of-line-Blocking (erster Request dauert und blockt alle weiteren)
- Größe der angefragten Ressource

=> Einfaches Bündeln von Anfragen verschlechtert die Performance eher

HTTP2 is a different (NEW!) Communication rule that use more efficient data transport method.

HTTP 2 NEW!



HTTP2 use one communication line per origin
TCP connection domain
It use Stream ⚡ to transfer data.

TIP 1
CSS image sprites and JS concatenation does not matter.
It'll be better files are separated & small.

TIP 2
Open one TCP connection and take advantage of Stream!!
No domain sharding on HTTP2 ☺

TIP 3
Browser supports HTTP2 ONLY
When communication is encrypted
Setup HTTPS if you want HTTP2

HTTP 2.0 (seit 2015)

- basiert in einigen Teilen auf SPDY (Google, 2012)
- binäres Protokoll → effizienteres Parsing, kompakter, keine Anfälligkeit gegenüber Groß-/Kleinschreibung/Zeilenenden etc.
- Multiplexing (mit Priorisierung), Domain-Sharing wird überflüssig
- im Idealfall nur eine Connection
- HPACK (Header-Compression)
- fast alle Browser unterstützen HTTP/2 nur mit TLS \geq 1.2

⚡ HTTP/2 Test Verify HTTP/2.0 Support

Online **HTTP/2 test** - Verify if your server or CDN supports HTTP/2.

URL

e.g. www.google.com

☒ Public

⚡ Test

HTTP/2 Test History

URL	HTTP/2 Support
https://twitter.com	supported
https://ebay.de	not supported
https://youtube.com	supported
https://amazon.com	not supported
https://facebook.com	supported
https://www.google.de	supported

<https://tools.keycdn.com/http2-test>

HTTP 2.0 Support

- Internet Explorer: **Ja (ab IE11)**
- Mozilla: **Ja**
- Chrome: **Ja (ab Version 43)**
- Safari: **Ja (ab Version 9)**
- Opera: **Ja**

-
- Apache / Nginx: **Ja**

Ressourcen schonen

- Cache richtig nutzen
- Daten in passendem Umfang holen (z.B. Bildgrößen je Endgerät)
- nur tatsächlich benötigte Daten holen (Mobile Daten + Akku schonen)
- schnell viel auf einmal holen (High-Power-Mode nutzen)
- Kompression wenn sinnvoll (keine Binaries komprimieren)
- mit HTTP/2 wird Vulcanizing überflüssig

Cache richtig nutzen

- HTTP/2 unterstützt PUSH (Vorsicht!) und Cache-Digests
- kein Vulcanizing (kleine Änderung → Bundle muss neu holt werden)
- HPACK (vgl. Body-GZIP)
- Headers
 - Cache-Control (max-age, public/private, must-revalidate, no-cache, no-store)
 - Expires
 - Last-Modified
 - ETag

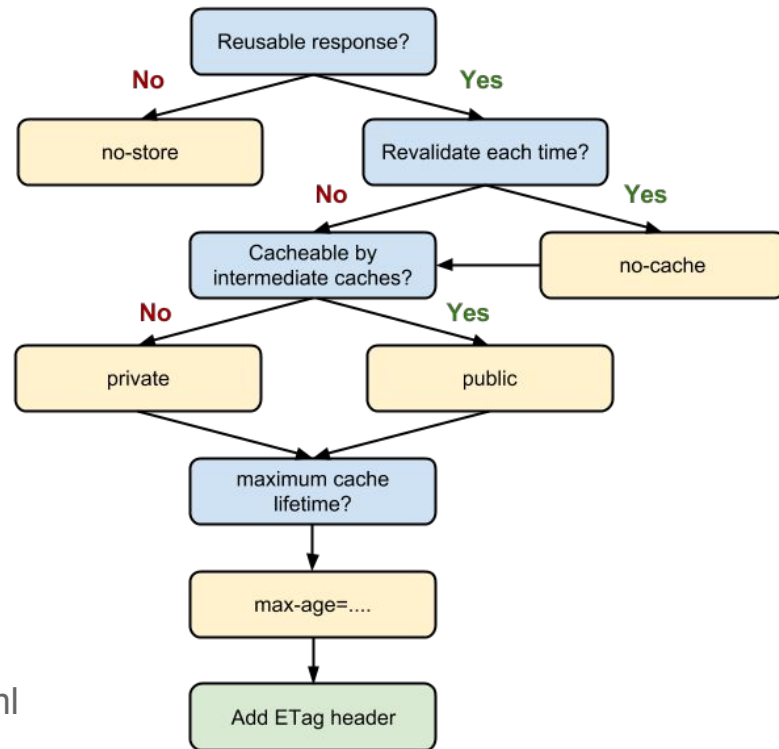
Cache richtig nutzen

- Chance zum Cachen zulassen
- auch mit HTTP/2 ist kein “Cache Busting” möglich

```
ExpiresByType text/html "access plus 1 month 15  
days 2 hours"
```

```
ExpiresByType image/gif "modification plus 5 hours  
3 minutes"
```

http://httpd.apache.org/docs/2.4/mod/mod_expires.html



Webserver ansprechen

- DNS-Auflösung
- SSL
- Rewrites
- Proxy-Server
- Filesystem (SYS-Calls)
- Prozessmanagement (Prefork, Worker, Event) → **Demo :-)**

<https://httpd.apache.org/docs/2.4/programs/ab.html>



Noch Fragen?

Amadeus Alfa, chemmedia AG