Client-side performance

Practical Distributed Systems 2022: Lecture 2b

Jarosław Rzeszótko
RTB House

Performance from the users perspective

Example user interaction with a web service (HTTP/1.0, no keepalive etc.):

- Type URL in the browser
- Resolve domain name in DNS
- Establish HTTP connection to main host
  - Establish TCP connection to host
  - Establish TLS connection
- Fetch HTML over the established connection
- Parse HTML, build DOM, start rendering
- Fetch assets: JS/CSS/images/fonts/etc.
- …

Performance from the users perspective

Example user interaction with a web service (HTTP/1.0, no keepalive etc.):

- …
- Fetch assets: JS/CSS/images/fonts/etc.
  - For each asset:
    - Resolve domain name in DNS
    - Establish TCP connection to host
    - Possibly establish TLS connection
  - …
- …

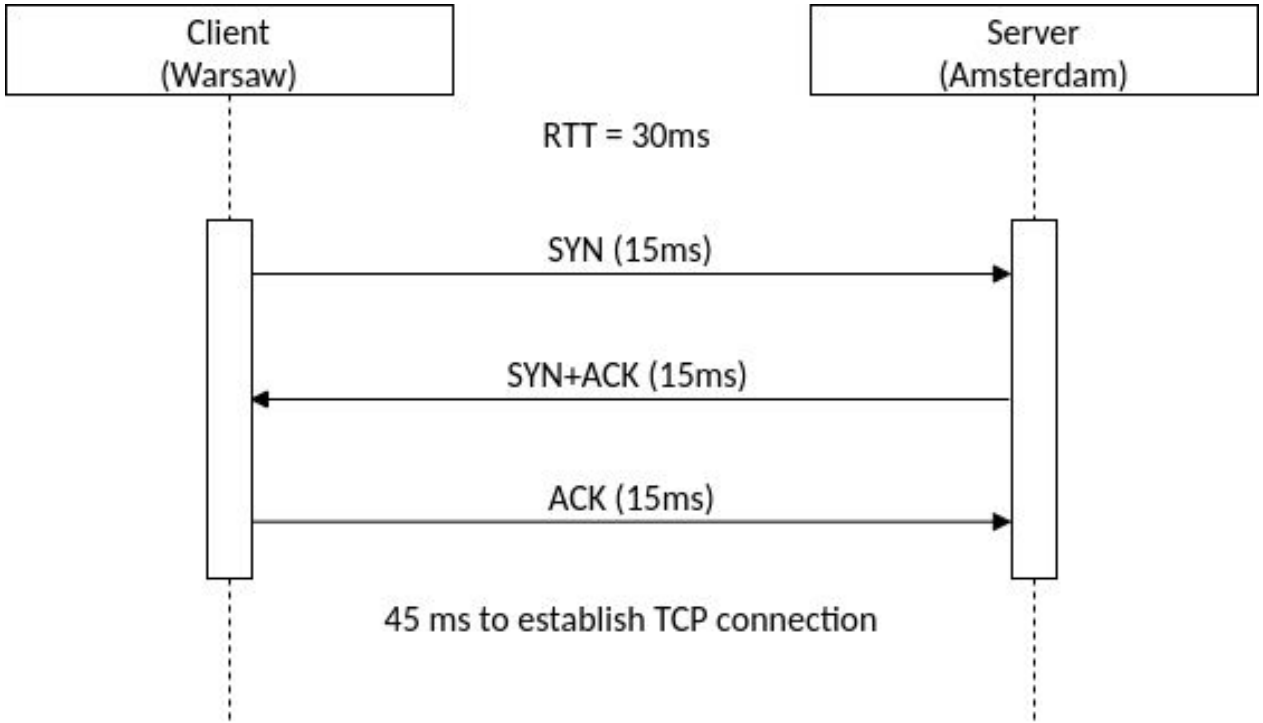
Problem: roundtrips

Roundtrip time / RTT / ping - time needed for any amount of data to flow from source to destination and back

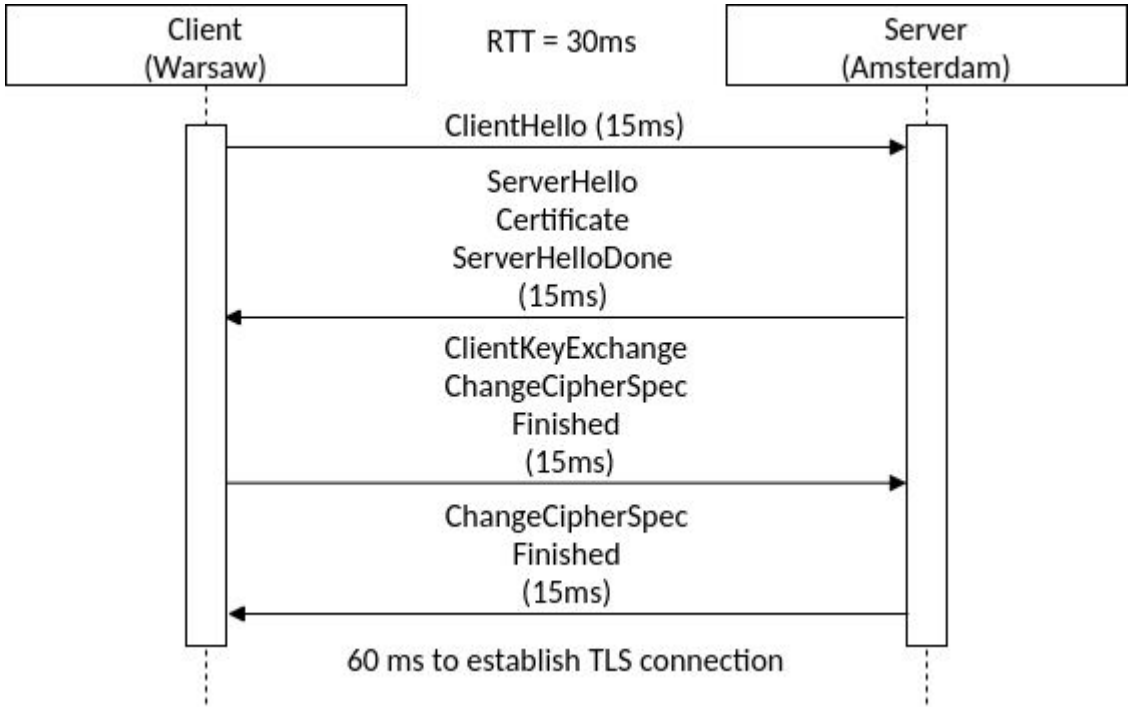
Can be measured with the ping command:

```
PING creativecdn.com (185.184.8.65) 56(84) bytes of data.
64 bytes from ip-185-184-8-65.rtbhouse.net (185.184.8.65): icmp_seq=1 ttl=56 time=32.5 ms
64 bytes from ip-185-184-8-65.rtbhouse.net (185.184.8.65): icmp_seq=2 ttl=56 time=32.5 ms
64 bytes from ip-185-184-8-65.rtbhouse.net (185.184.8.65): icmp_seq=3 ttl=56 time=35.0 ms
64 bytes from ip-185-184-8-65.rtbhouse.net (185.184.8.65): icmp_seq=4 ttl=56 time=35.7 ms
64 bytes from ip-185-184-8-65.rtbhouse.net (185.184.8.65): icmp_seq=5 ttl=56 time=33.1 ms
^C
--- creativecdn.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 32.467/33.740/35.652/1.335 ms
```

Problem: roundtrips



Problem: roundtrips



Problem: roundtrips

Problem:
For two given geographic locations of client and server,
roundtrip time has a physical lower limit given by speed of light

Problem: roundtrips

```
Speed of light in vacuum:           300 000 000 m/s
Speed of light in optical fiber:    200 000 000 m/s
Distance from Warsaw to Amsterdam:    1 100 000 m

1 100 000 m / 200 000 000 m/s = 0.0055 s = 5.5 ms

2 * 5.5 ms = 11 ms best-case round trip
```

https://tools.bunny.net/latency-test?query=creativecdn.com

Problem: roundtrips

TCP is not using full bandwidth immediately after handshake:

The OS sending data uses a **congestion control** algorithm that controls how much data is injected into the network

Problem: roundtrips

- **Congestion window (cwnd)** on the sending side OS controls how many bytes can be in-flight
- TCP starts in “slow start” (unfortunate name):

```
cwnd = 10*MSS =~ 15kB (on Linux)
cwnd += MSS after every ACK
```

- Exponential growth:

```
cwnd(t+RTT) = 2*cwnd(t)
```

- Continues until congestion is detected

Problem: roundtrips

With 20ms RTT and no loss:

```
segments_acked(0ms)   = 0     cwnd(0ms)   = 10
segments_acked(20ms)  = 10    cwnd(20ms)  = 20
segments_acked(40ms)  = 30    cwnd(40ms)  = 40
segments_acked(60ms)  = 70    cwnd(60ms)  = 80
segments_acked(80ms)  = 150   cwnd(80ms)  = 160
segments_acked(100ms) = 310   cwnd(100ms) = 320
```

310*1.5kB = 465kB sent and ACKed after 100ms

Problem: head-of-line blocking in HTTP up to 1.1

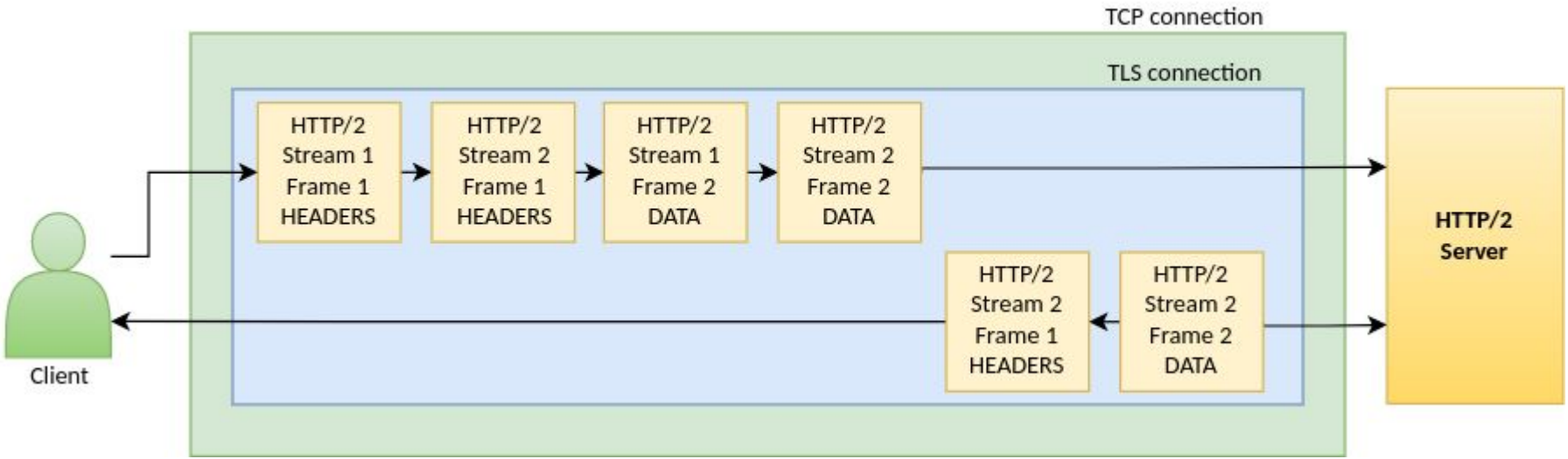
When an HTTP request is sent over a TCP connection (perhaps also TLS encrypted),
second request can not be sent until the server processes the first requests and responds

Head-of-line blocking: open more connections?

- Problem: opening connections costs memory and CPU (with TLS)
  - In the client
  - In the server
  - In the infrastructure between the client and the server

HTTP/2

- Multiple interleaved streams



Problem: head-of-line blocking in TCP

HTTP/2 still uses TCP, if the TCP stream is missing a byte somewhere in the stream, all HTTP/2 streams will be paused until that byte is retransmitted and received

HTTP/3

- UDP and own congestion control

Measuring end-to-end latency in the browser

Browsers expose APIs helpful for measuring latency:

- **Navigation Timing API**
  - For measuring root document load times
- **Resource Timing API**
  - For measuring load times of assets

Can collect data in client-side JS and send to some internal reporting API

Resources

Client-side latency:

- **High Performance Browser Networking**
  O'Reilly book also available online for free
  https://hpbn.co/