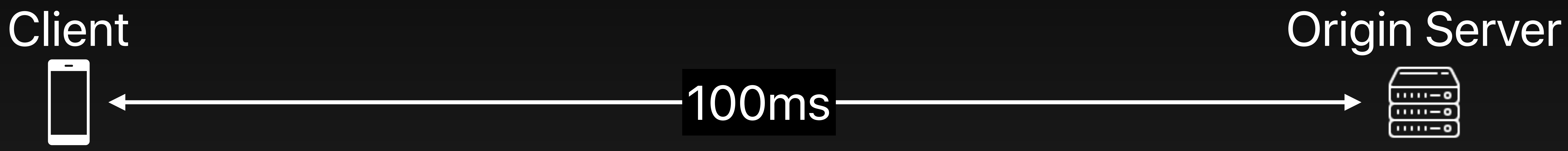
Pre-warming TLS Tunnel

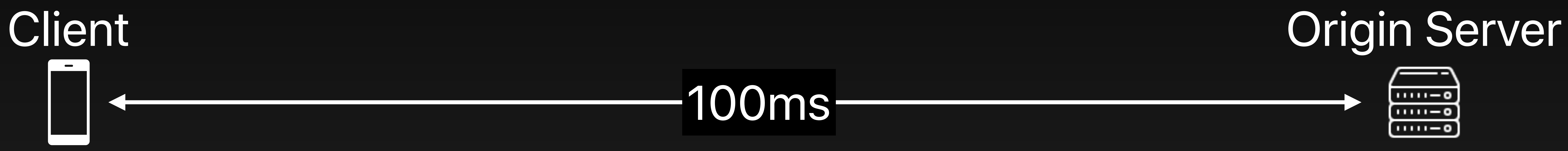
Apache Traffic Server Fall 2020 Summit

Masaori Koshiba (masaori@apache.org)

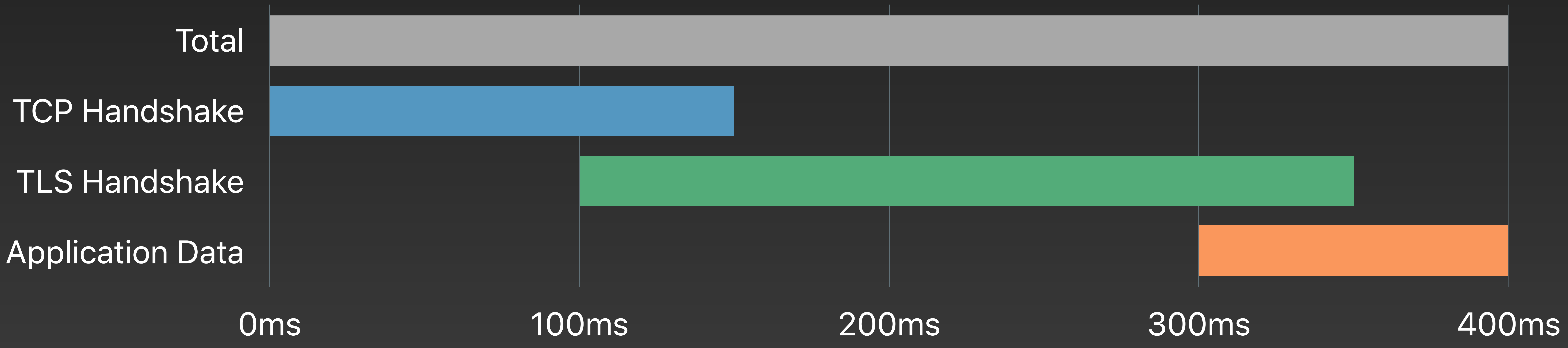
Reduce Latency of TLS Connections


Client
100ms
Origin Server

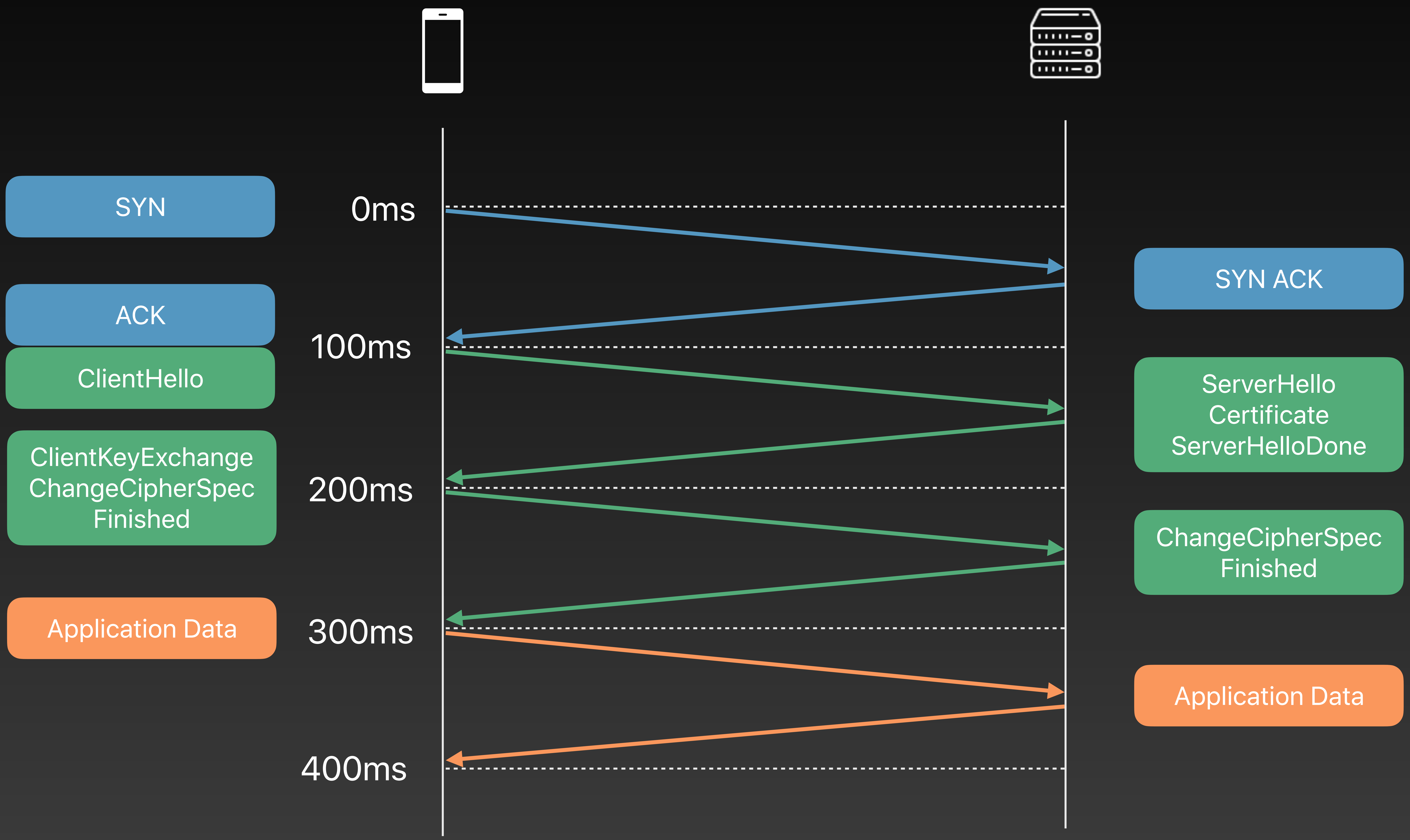
Time to First Byte



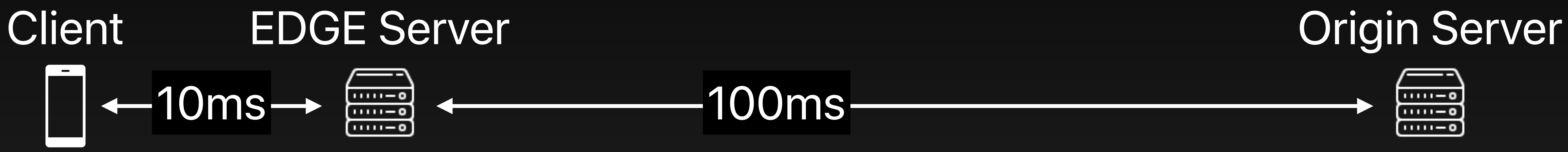
Time to First Byte > **400**ms



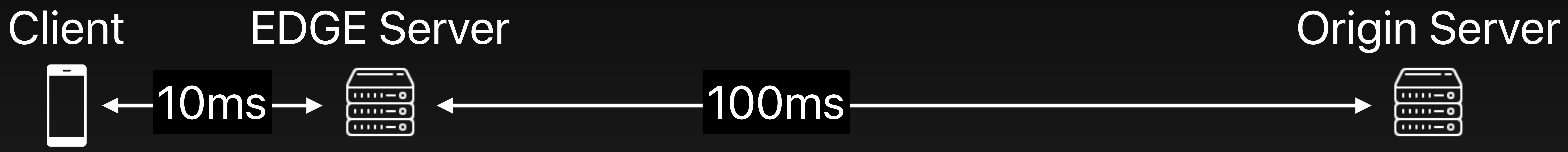
TLS 1.2 Full Handshake



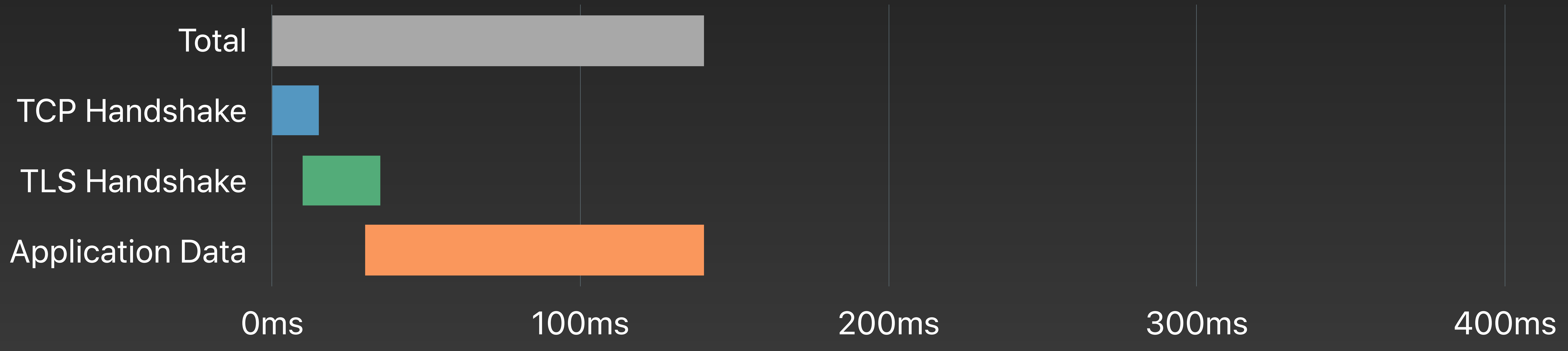
Early Termination & Connection Pool



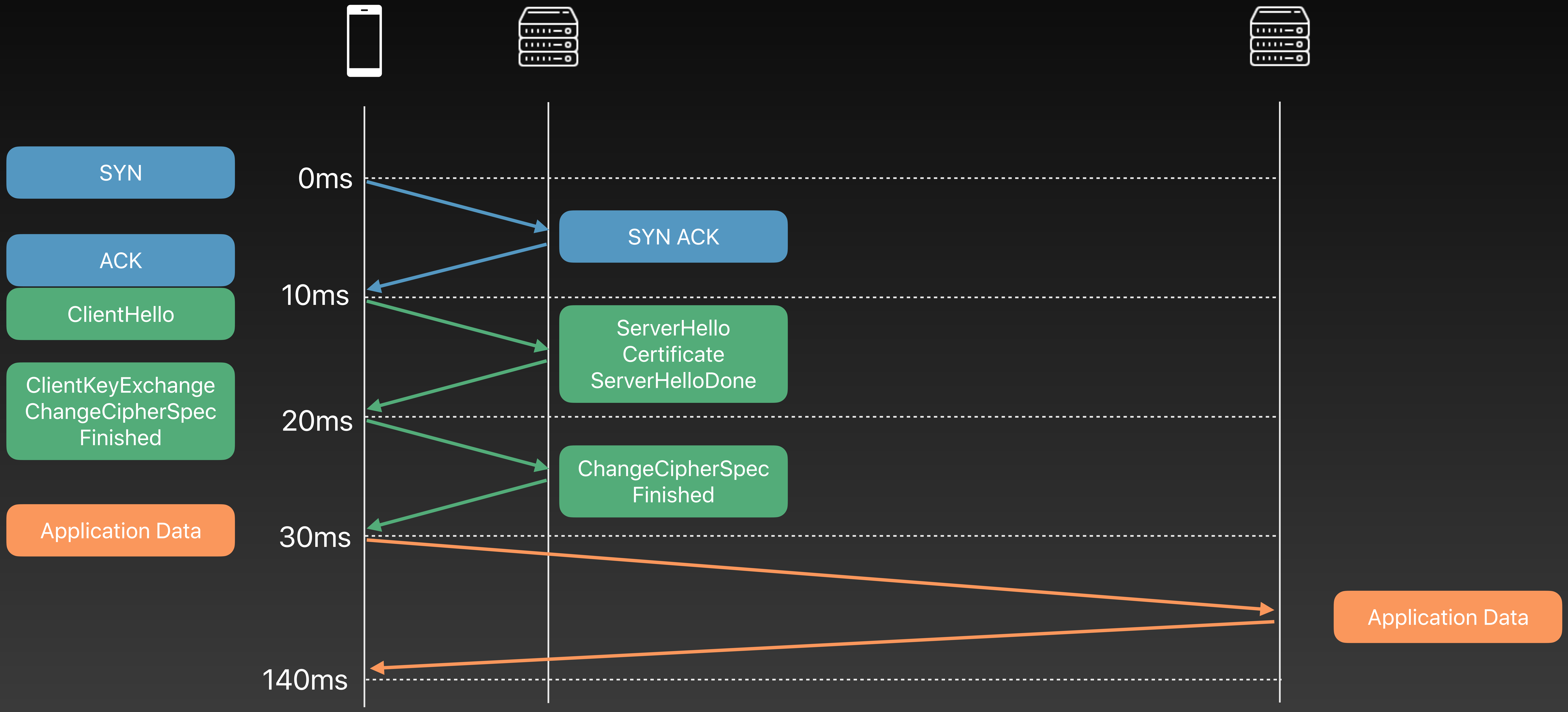
Early Termination & Connection Pool



Time to First Byte > **140**ms



Early Termination & Connection Pool

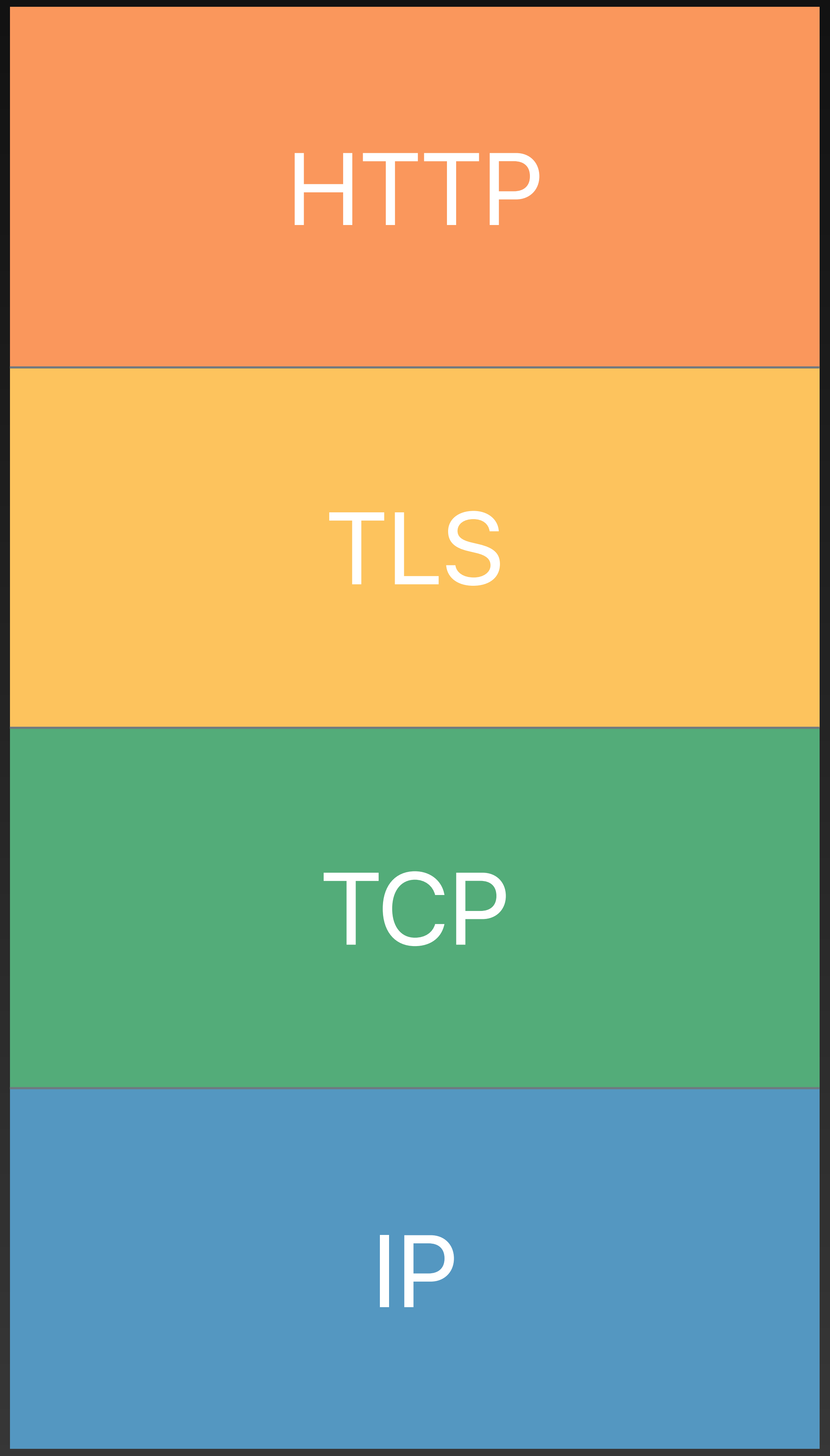


Implementation

Requirements

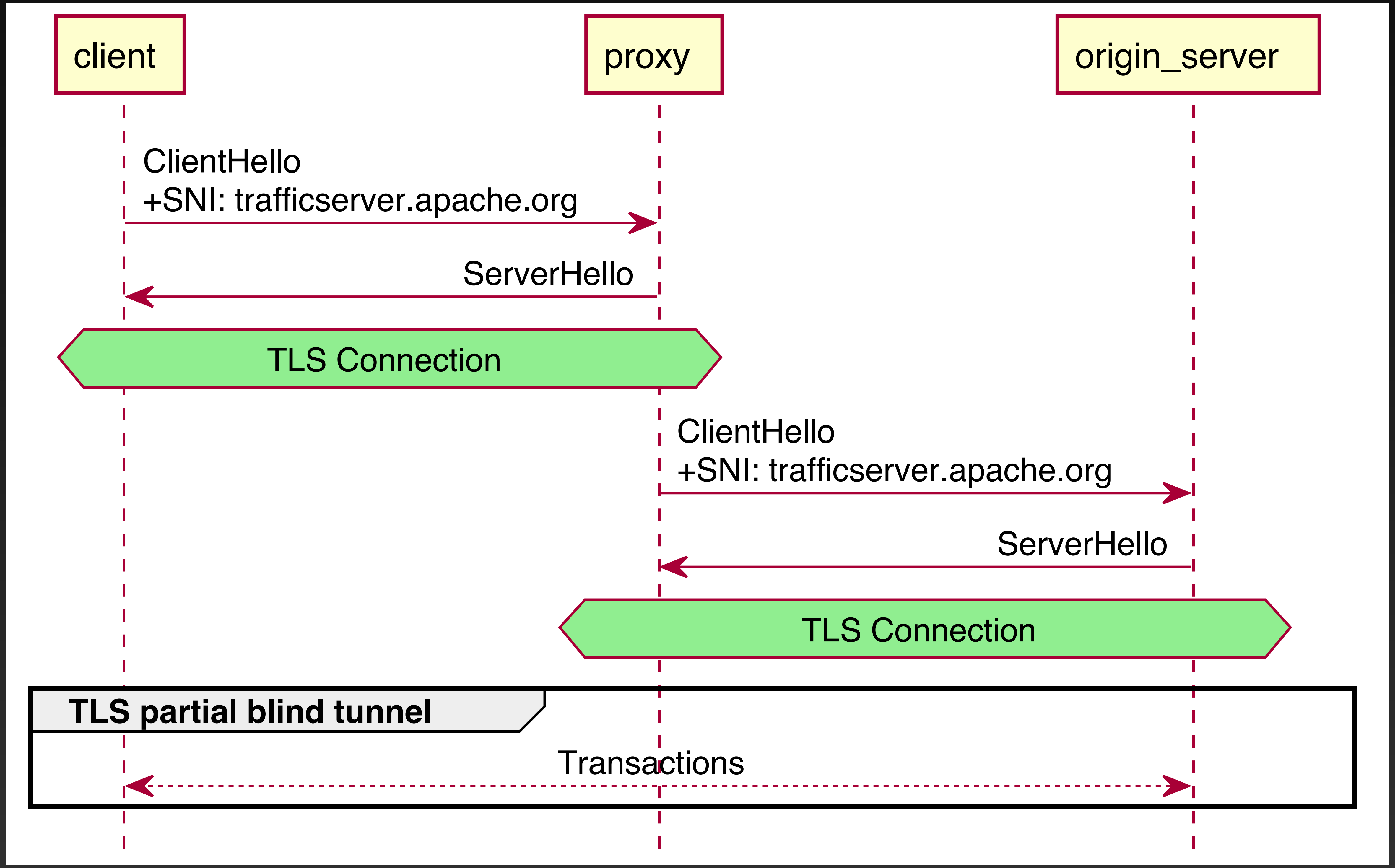
- TLS Proxy
  - Multiple Application Protocol Support
- TLS Connection Pool
  - No Keep-Alive on TLS
  - Keep the connection pool “hot”

→**Pre-warming**

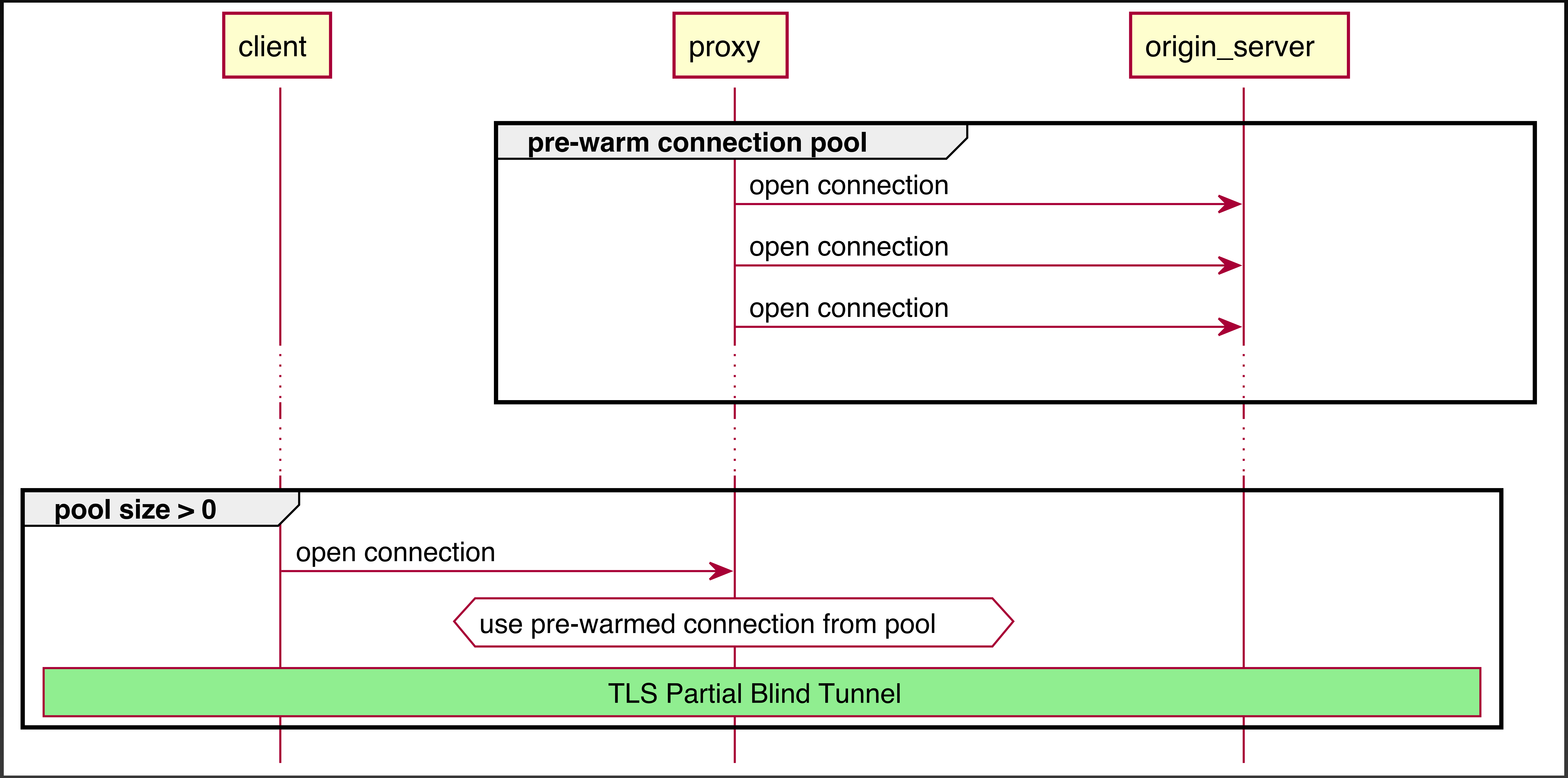


TLS Partial Blind Tunnel

- SNI Routing #2754
  - @persiaAziz, Nov 2017
  - ATS v8.0.0
- Adds partial_blind_routes sni action #6538
  - @randall, Mar 2020
  - ATS v9.0.0

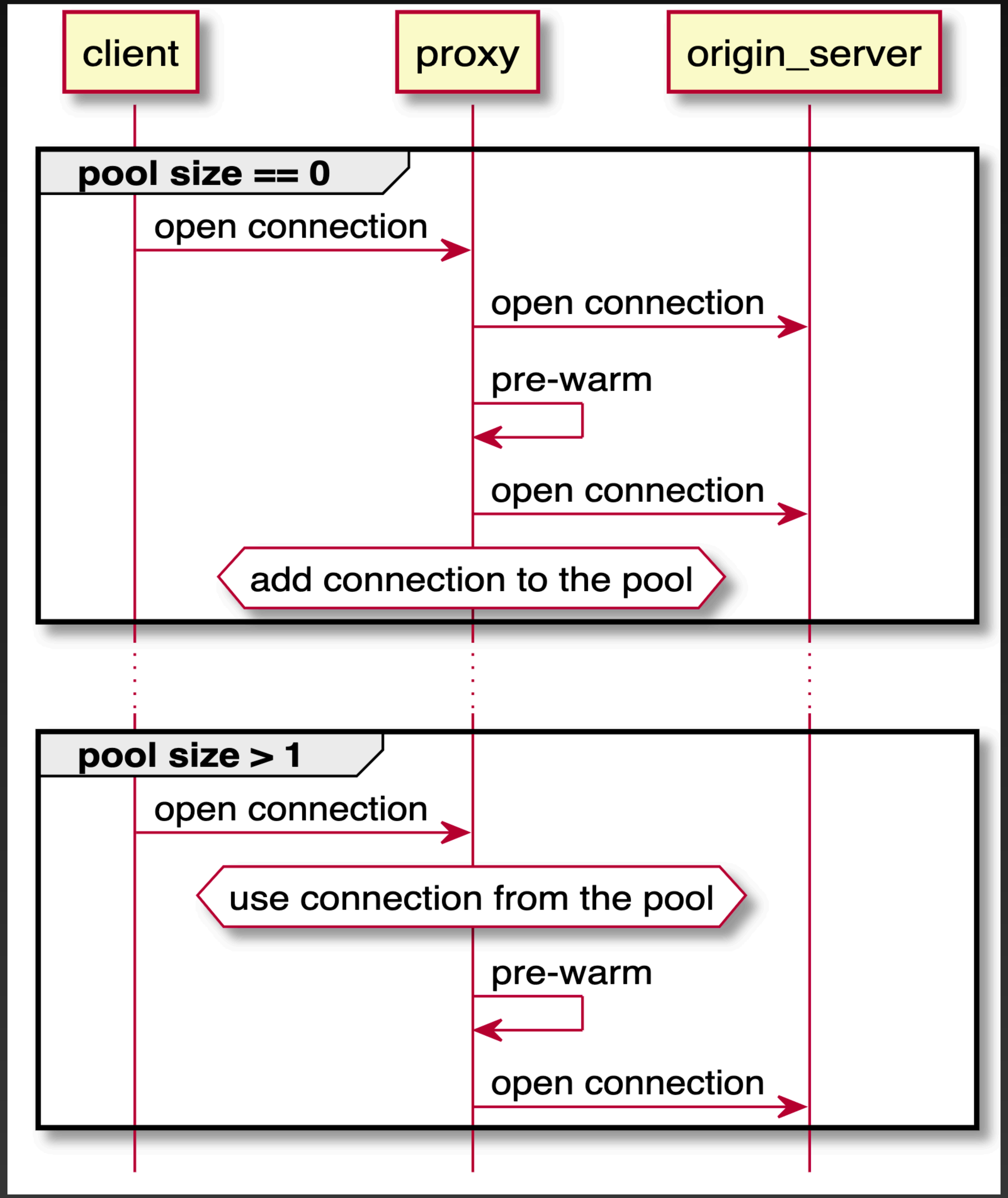


Pre-warm TLS Tunnel #7241

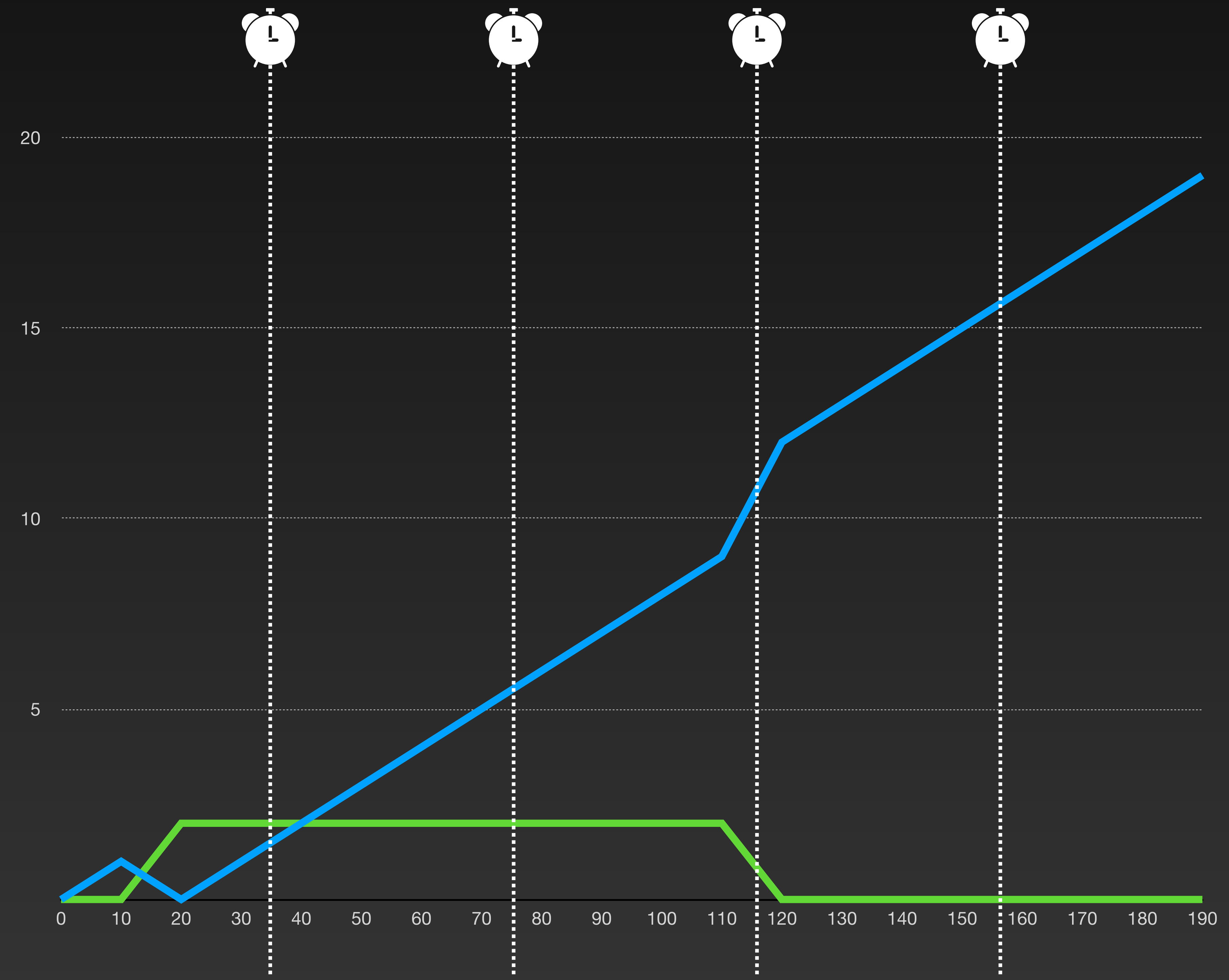


Pre-warming Algorithm

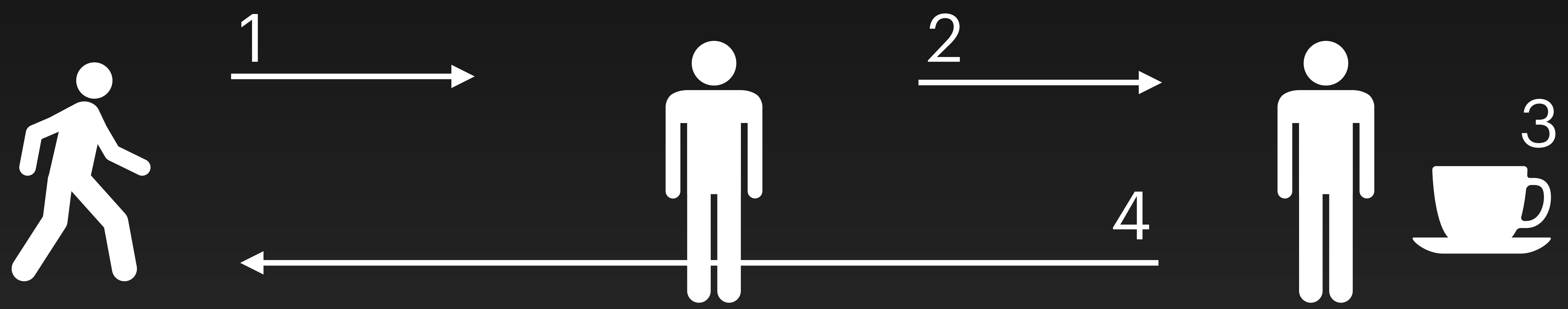
Event based pre-warming



Periodical pre-warming



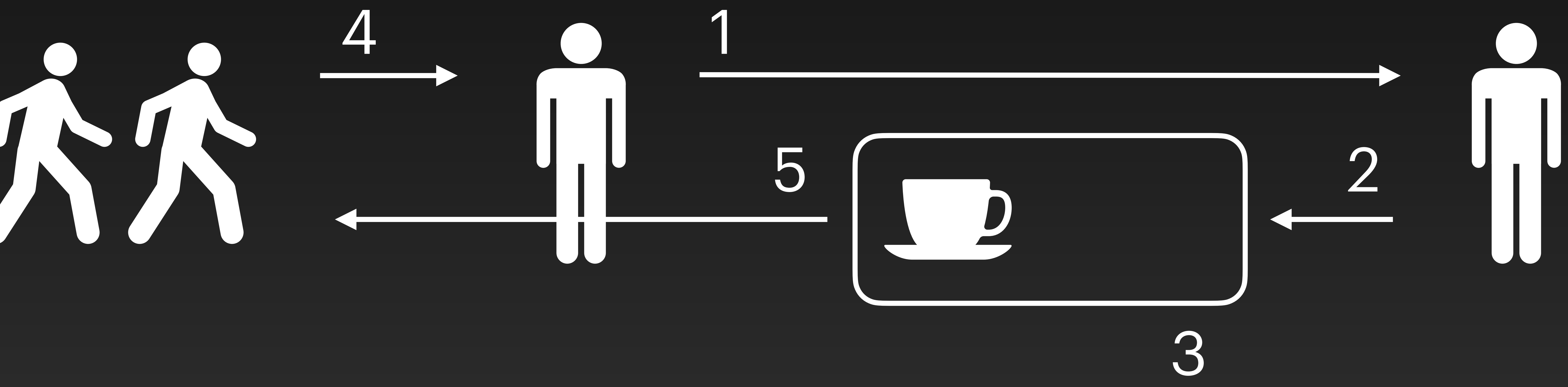
Coffee Shop



1. take an order
2. ask barista
3. brew coffee
4. serve coffee
5. back to step 1

Brew coffee BEFORE taking an order

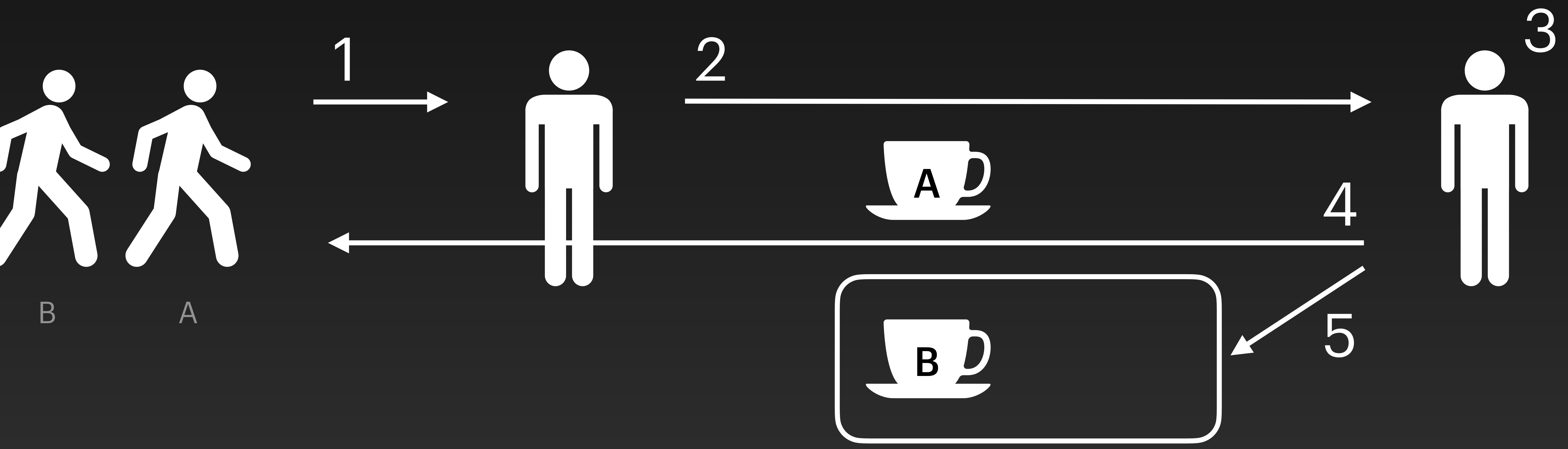
Event based pre-warming



1. ask barista
2. brew coffee
3. keep it in the pool
4. take an order
5. serve coffee
6. back to step 1

One more cup of coffee, please!

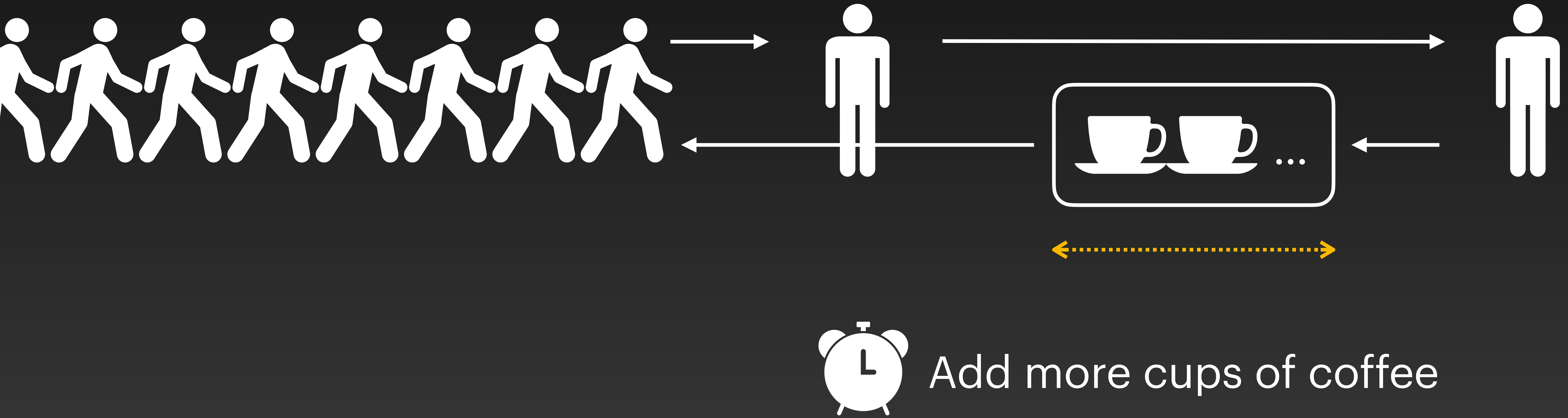
Event based pre-warming



1. take an order
2. ask barista
   2 cups of coffee
3. brew coffee
4. serve coffee
5. keep 1 in the pool

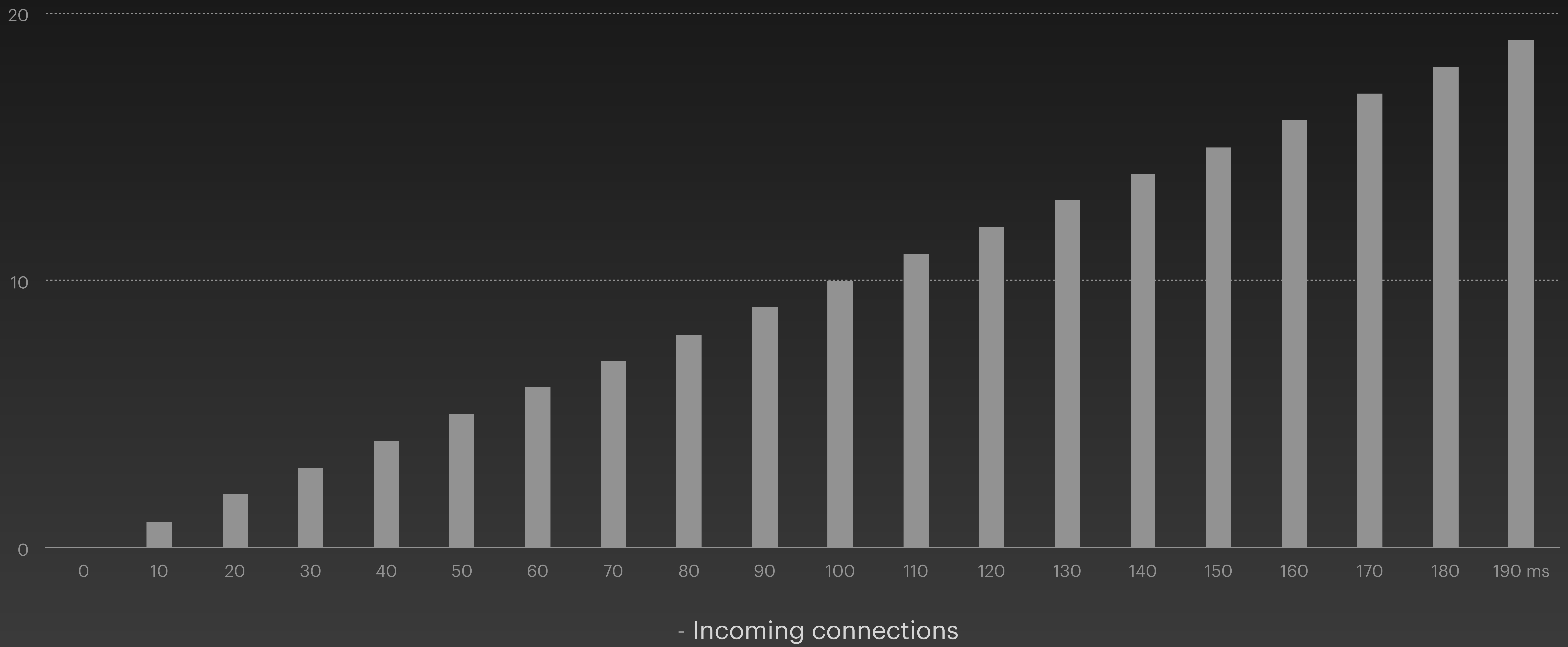
How many cups of coffee do you need?

Periodical pre-warming

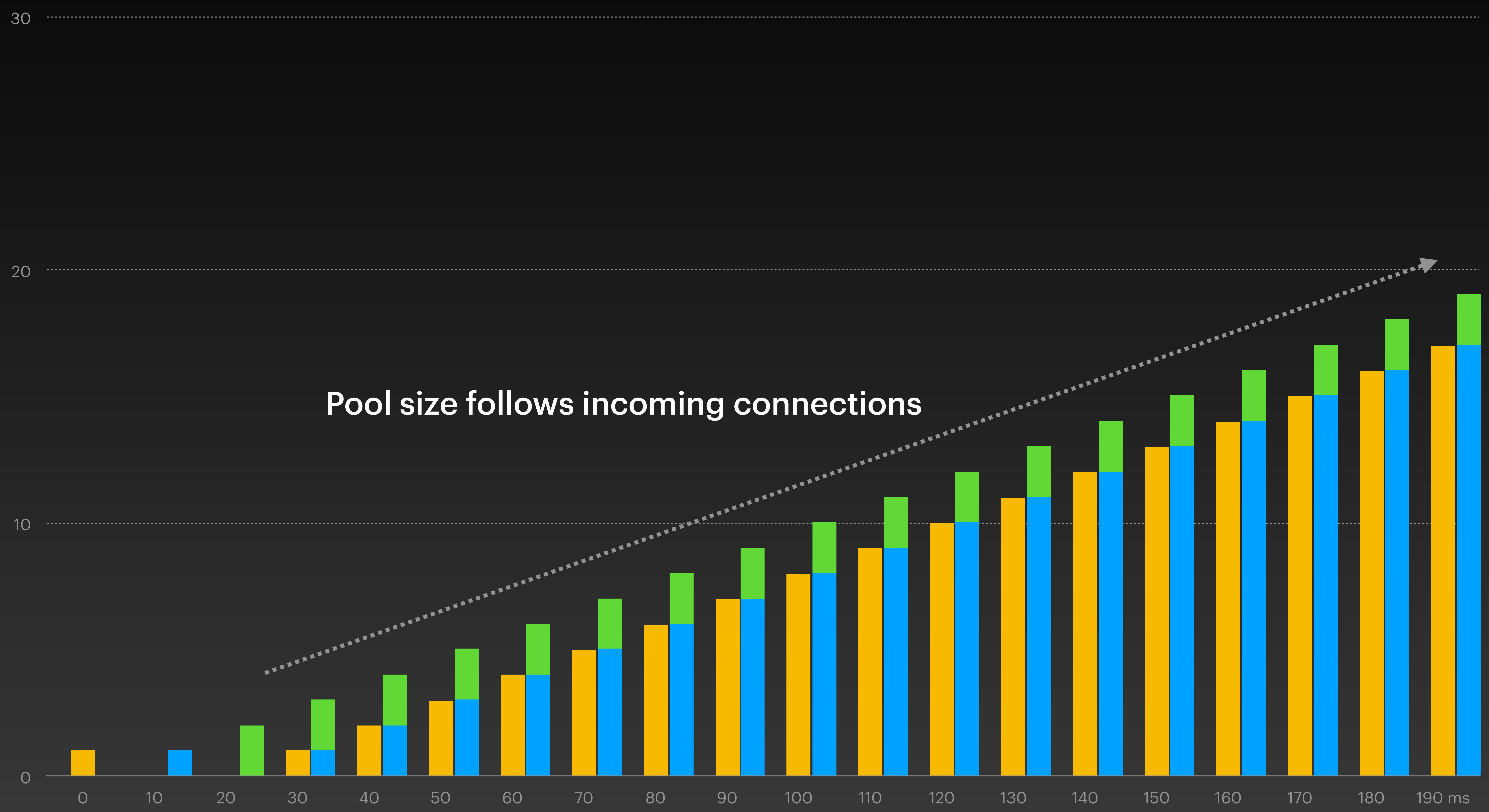


Example of Algorithm

- Incoming requests keep increasing (+1 per 10ms)
- Time to pre-warm = 20ms

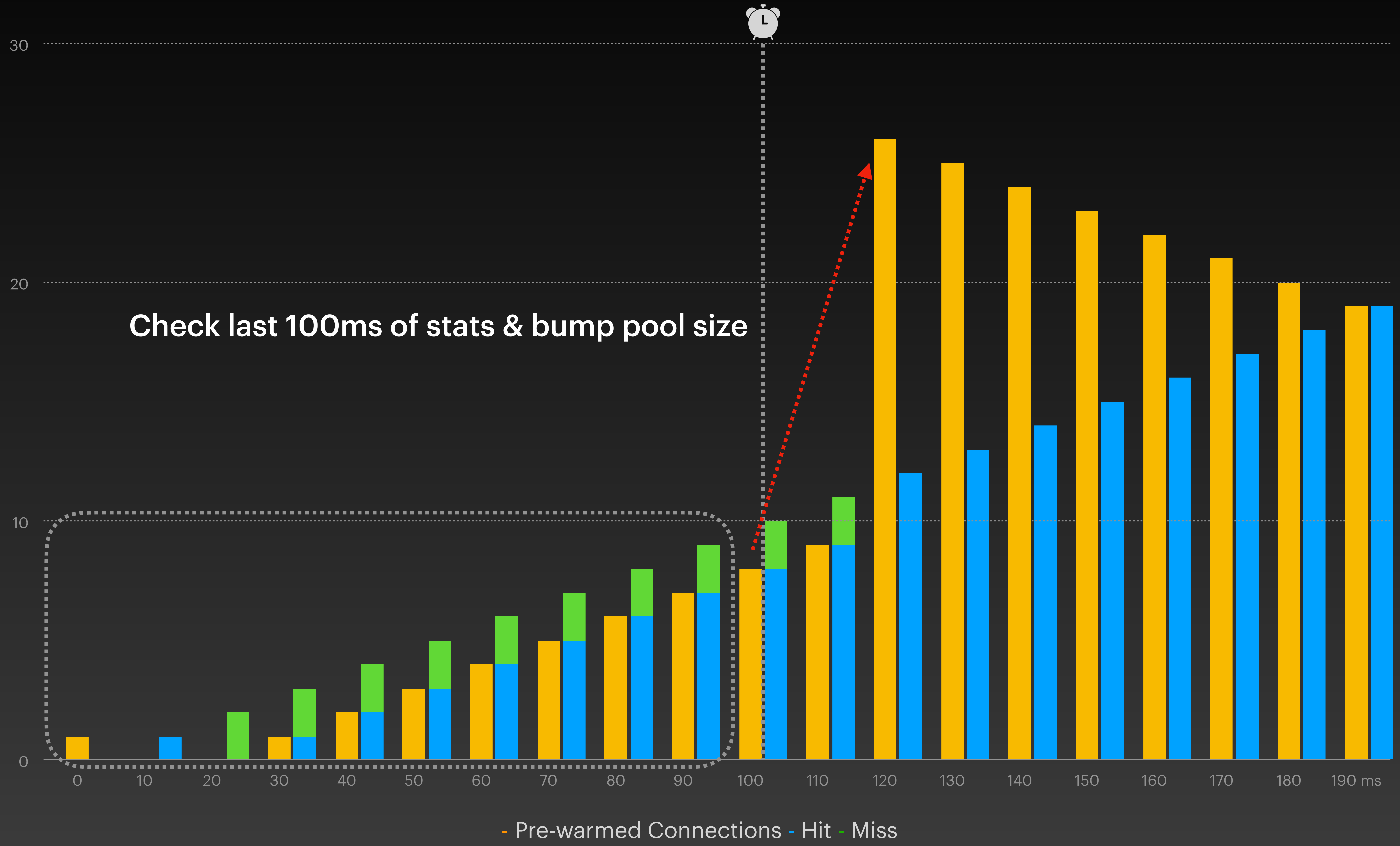


Event based pre-warming



- Pre-warmed Connections - Hit - Miss

Event based pre-warming + Periodical pre-warming



Other Hacks

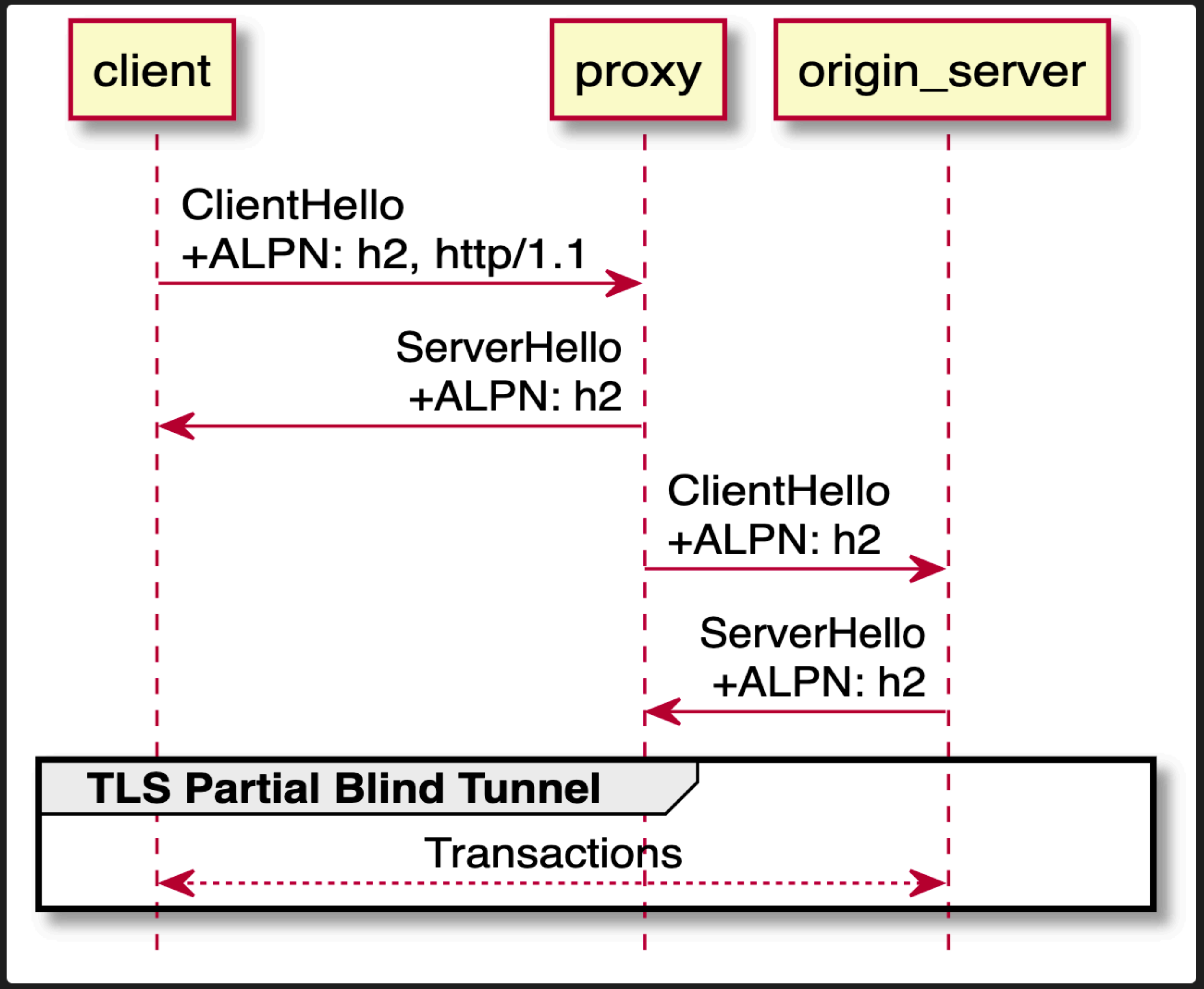
- Timeouts
- Exponential Retry
- Logging
- Metrics (per pool)
- Configs (records.config, sni.yaml)

ALPN Support on TLS Partial Blind Tunnel

- Config on proxy
  - e.g.

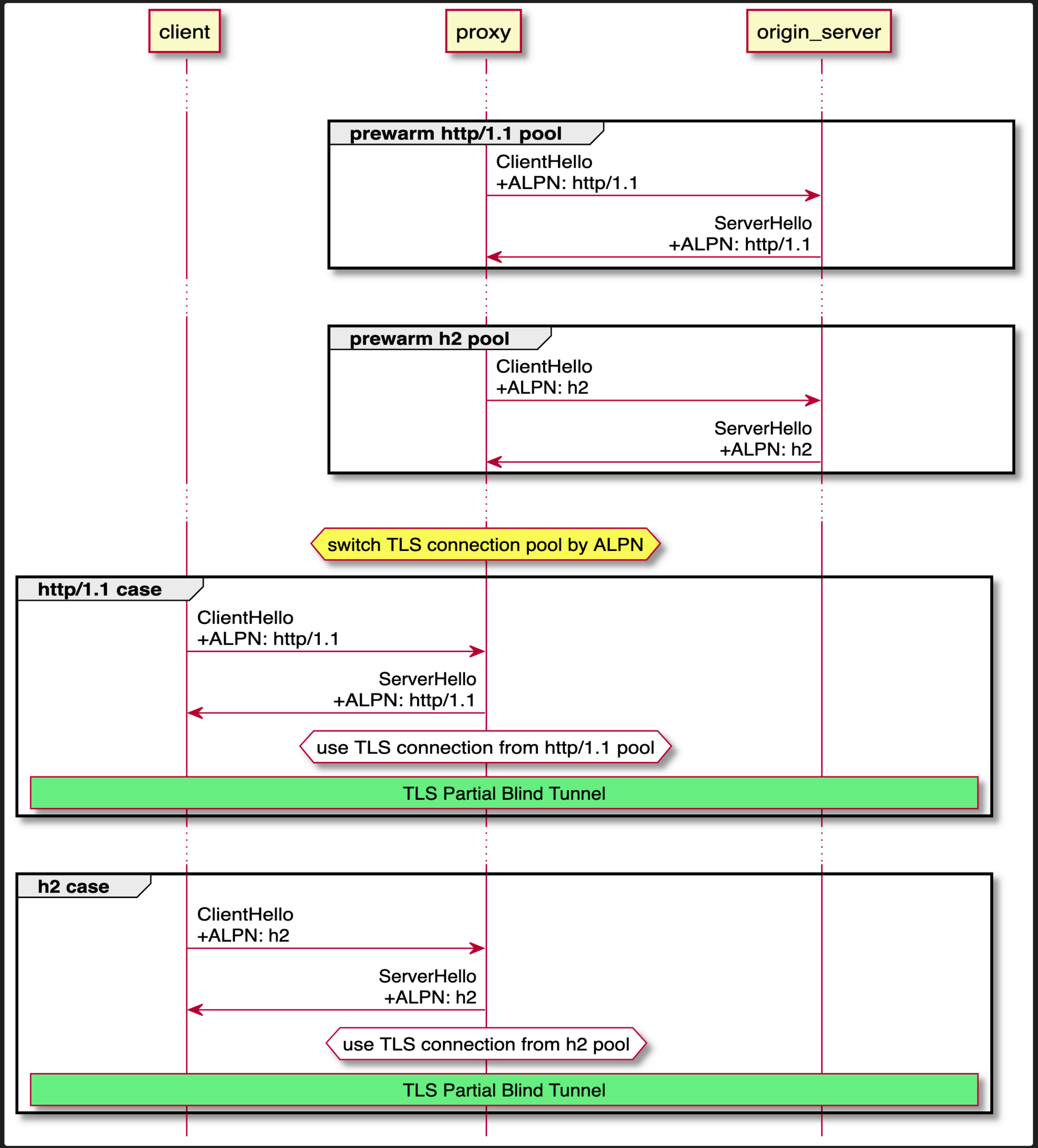
```
tunnel_alpn:
  - h2
  - http/1.1
```

- Froward negotiated ALPN Protocol ID



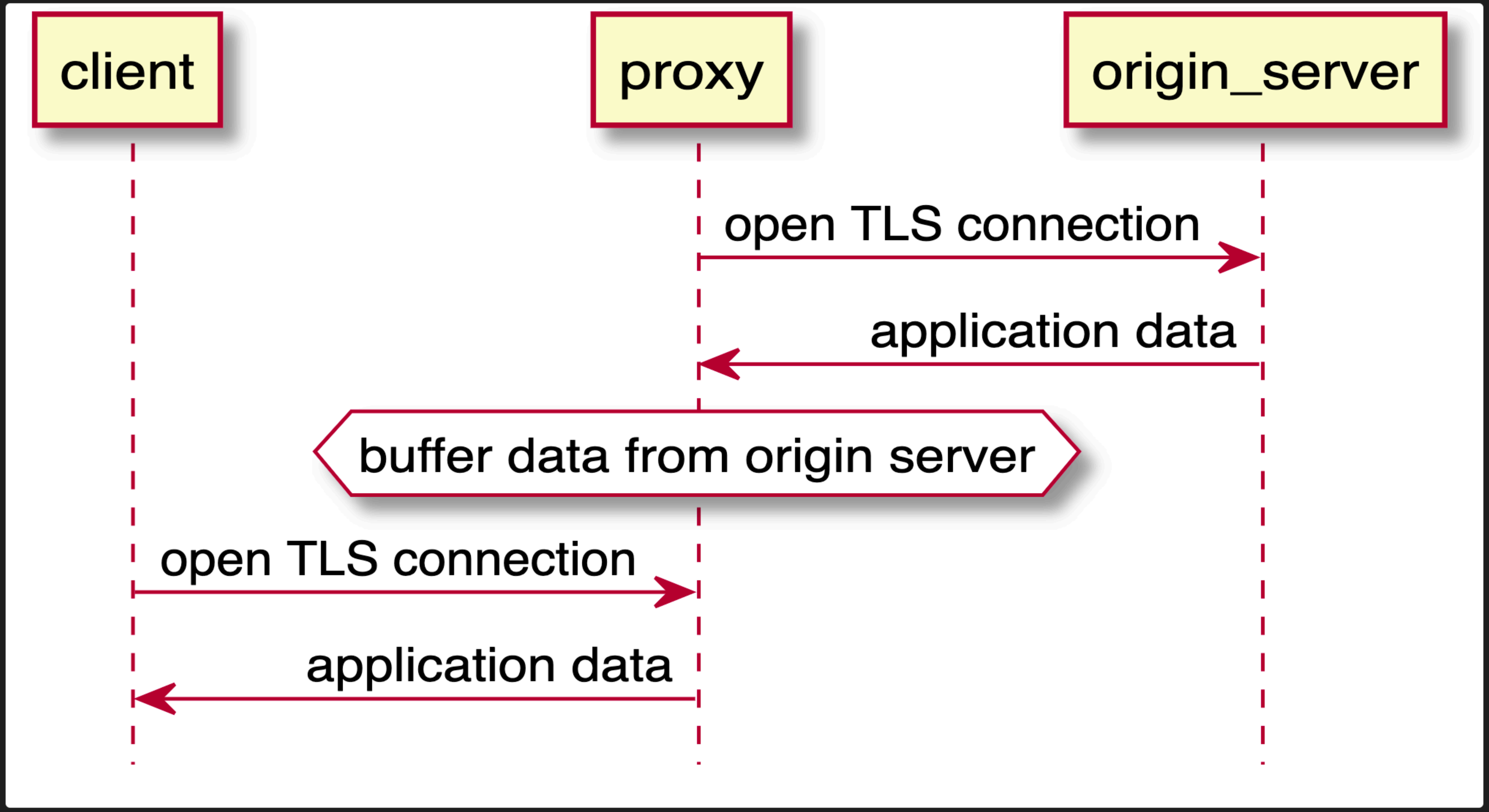
ALPN Based Connection Pool

- Pre-warm pool per ALPN Protocol ID
- Switch pool by ALPN Protocol ID



Buffering Data from Origin Server

- e.g. HTTP/2 Settings Frame

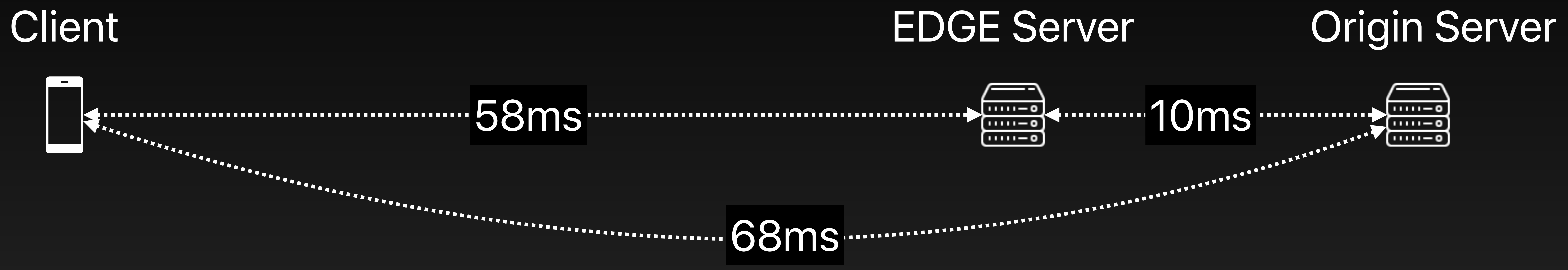


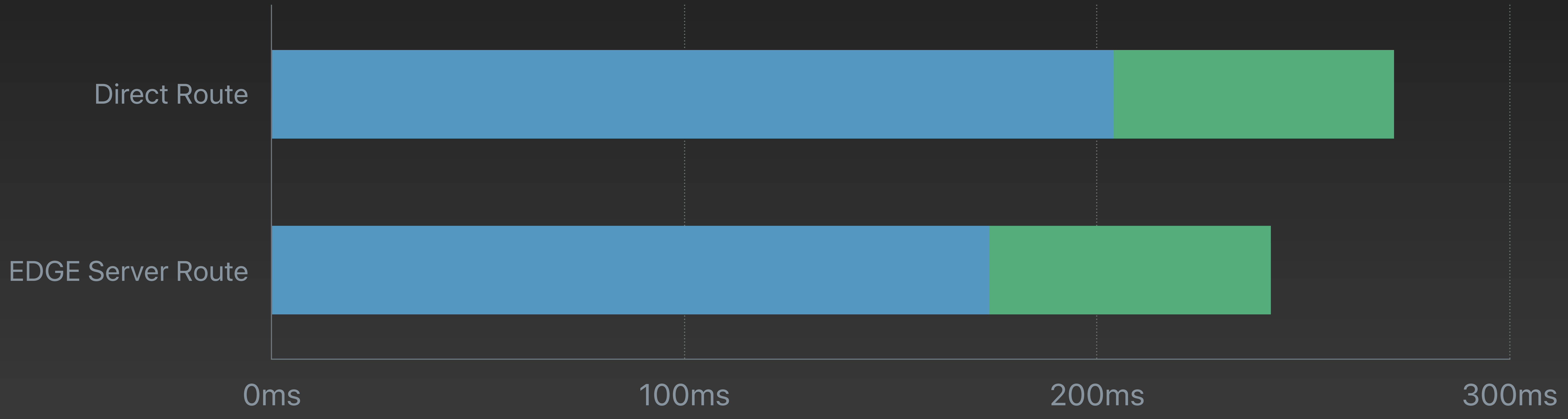
Example of sni.yaml

```
sni:
- fqdn: trafficserver.apache.org
  partial_blind_route: origin.trafficserver.apache.org:443
  http2: false
  tunnel_alpn:
    - h2
    - http/1.1
  tunnel_prewarm: true
  tunnel_prewarm_connect_timeout: 10
  tunnel_prewarm_inactive_timeout: 150
  tunnel_prewarm_min: 1
  tunnel_prewarm_max: 100
```

Trial of PoC

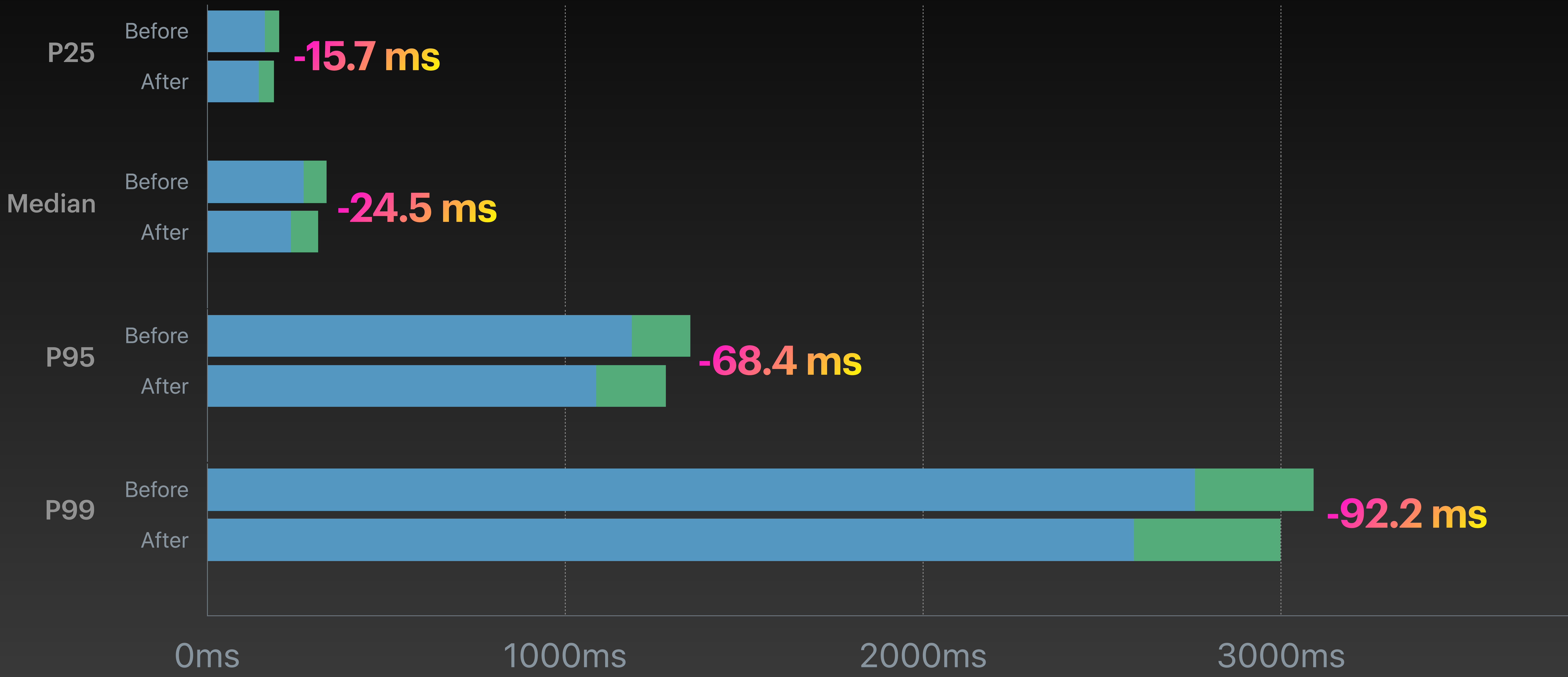
Conditions / Expectations





-30ms

Result - Time to First Byte

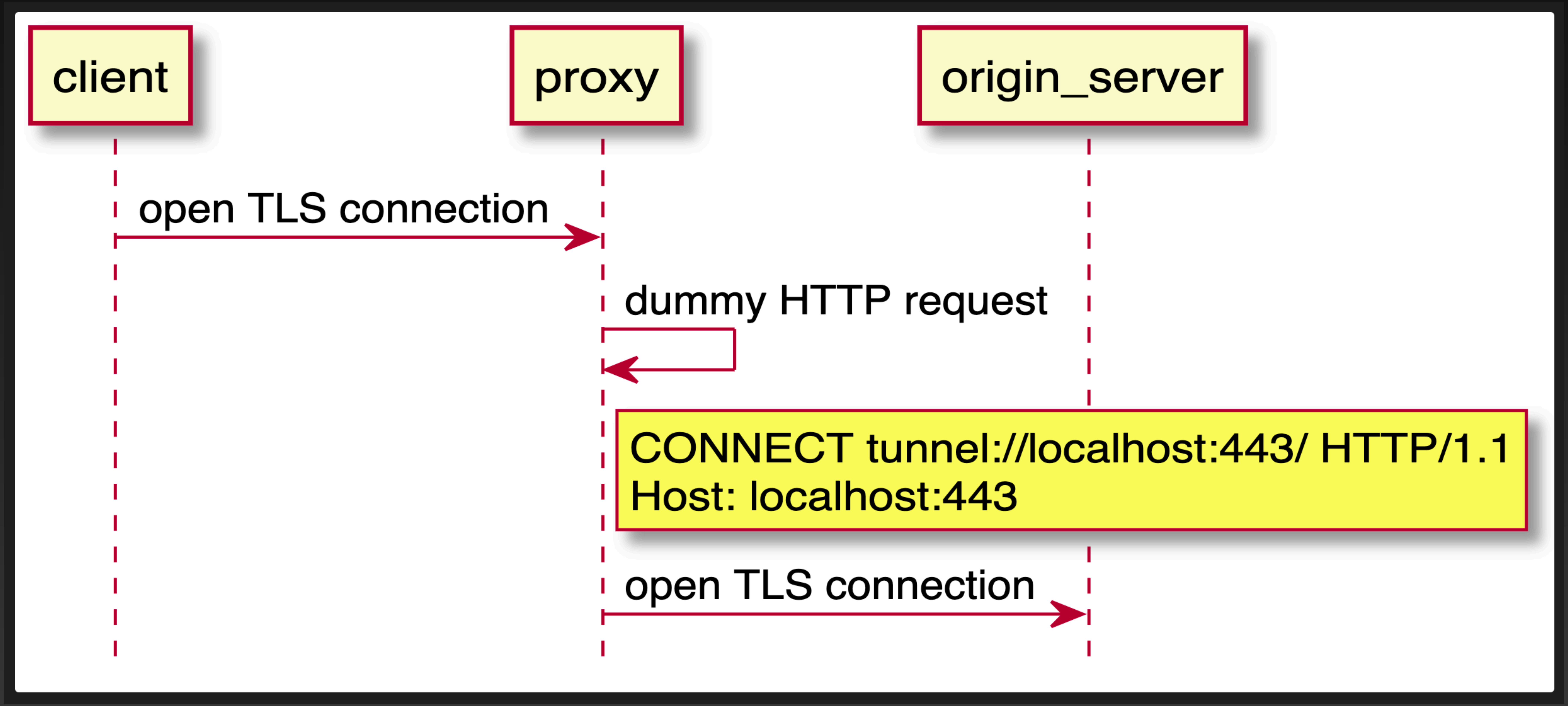


Smooth Pool Expansion in the beginning

current pool size
total_hit / total_miss (delta)
pre-warmed connection hit rate
100.0%
80.0%
60.0%
40.0%
20.0%
0%
99.99%

Issues

SNI Routing Implementation



CONNECT tunnel://localhost:443/ HTTP/1.1
Host: localhost:443

Connection Metrics

Incoming or Outgoing?

✗

```
proxy.process.net.connections_currently_open
proxy.process.ssl.total_handshake_time
proxy.process.ssl.ssl_error_syscall
```

✓

```
proxy.process.ssl.total_attempts_handshake_count_in
proxy.process.ssl.total_attempts_handshake_count_out
```

Summary

- Pre-warming TLS Tunnel reduces Time to First Byte (TTFB)
- Trial of PoC
  - TTFB is improved 7% (-24.5ms)
  - Pre-warmed Connection Hit Rate is 99%

Future Work

Short Term

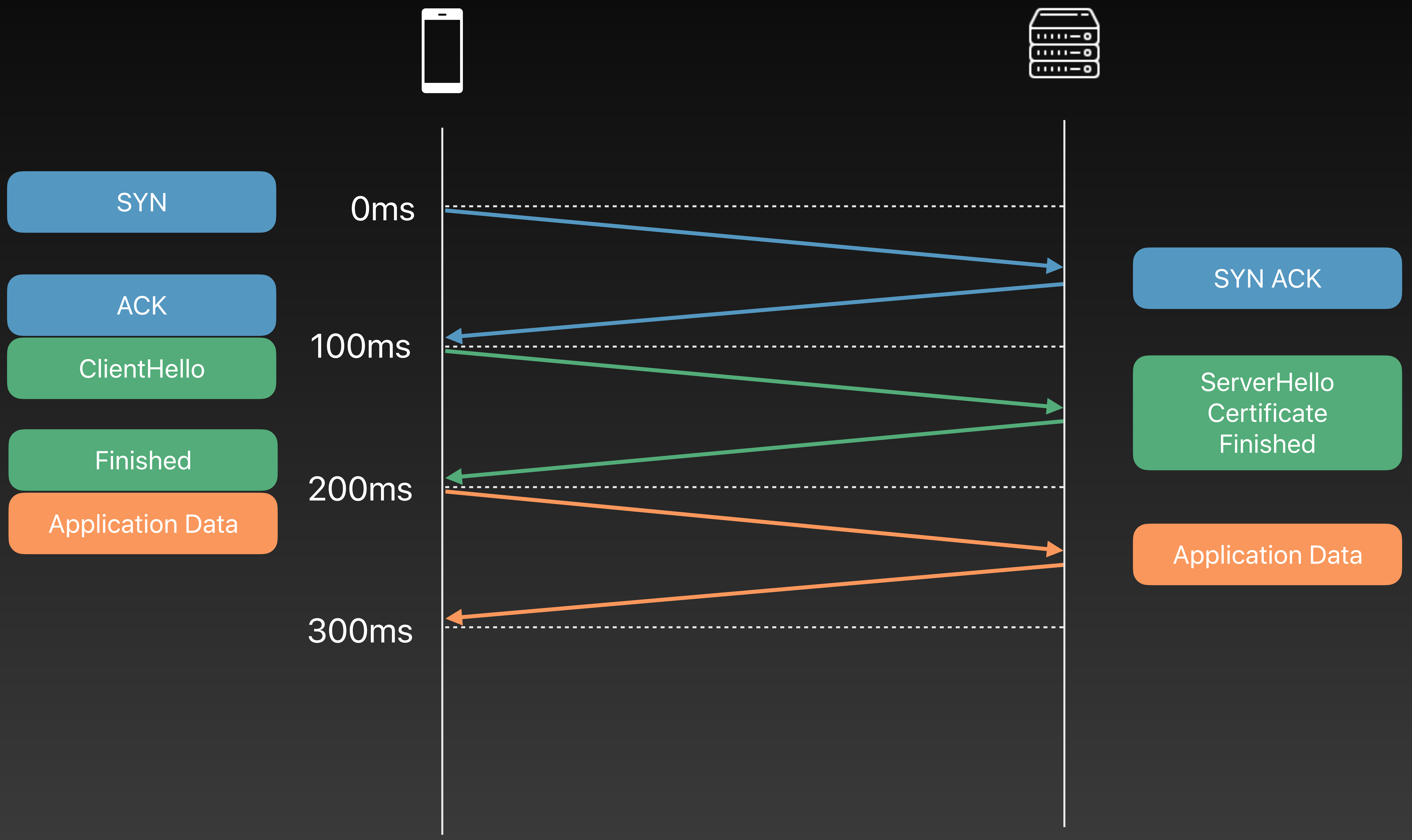
- Open PRs
- Tune Pre-warming Algorithm

Long Term

- Huge refactoring of ATS as L4 Proxy
- Pre-warm HTTP Session

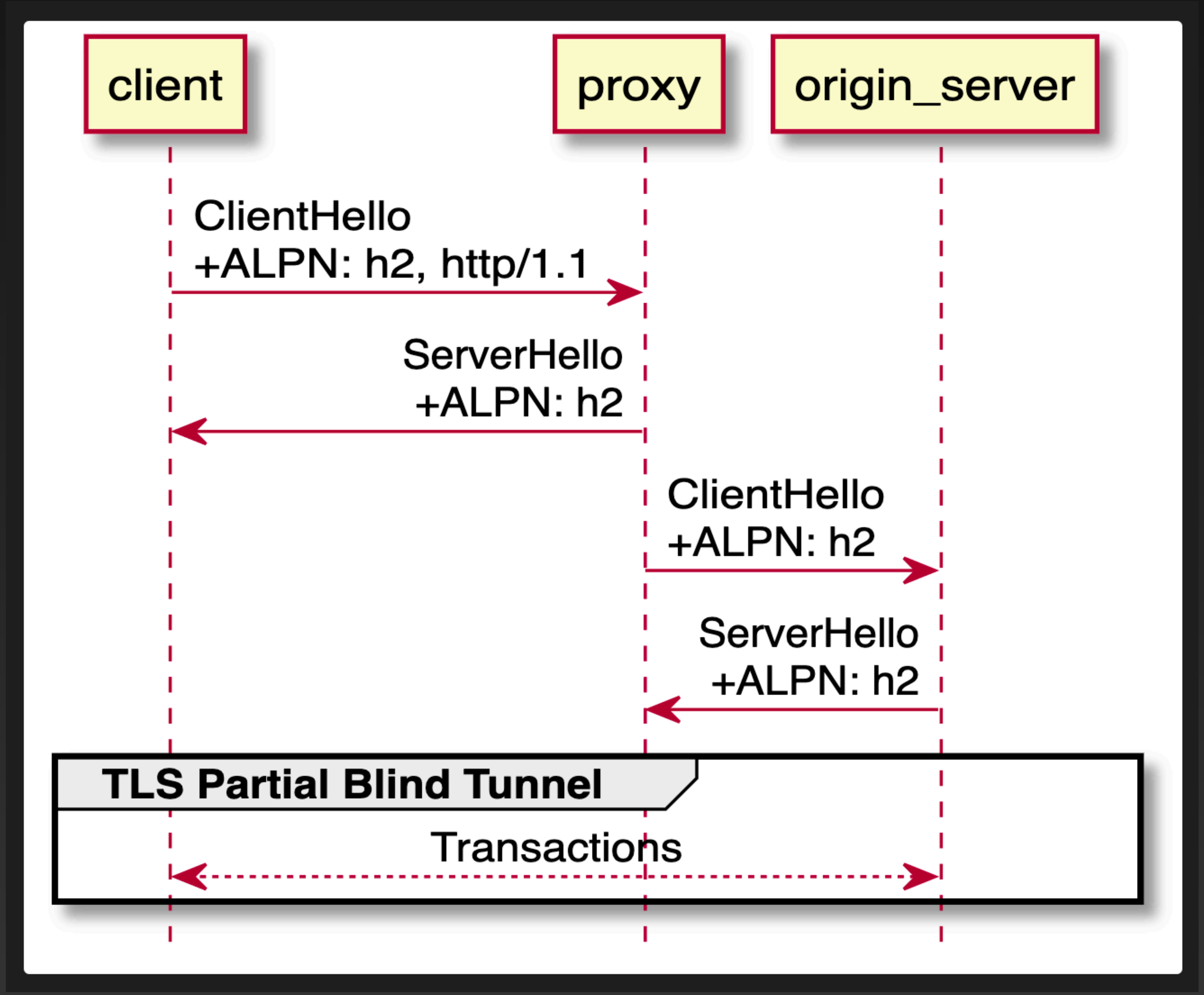
traffic server

TLS 1.3 1-RTT Handshake

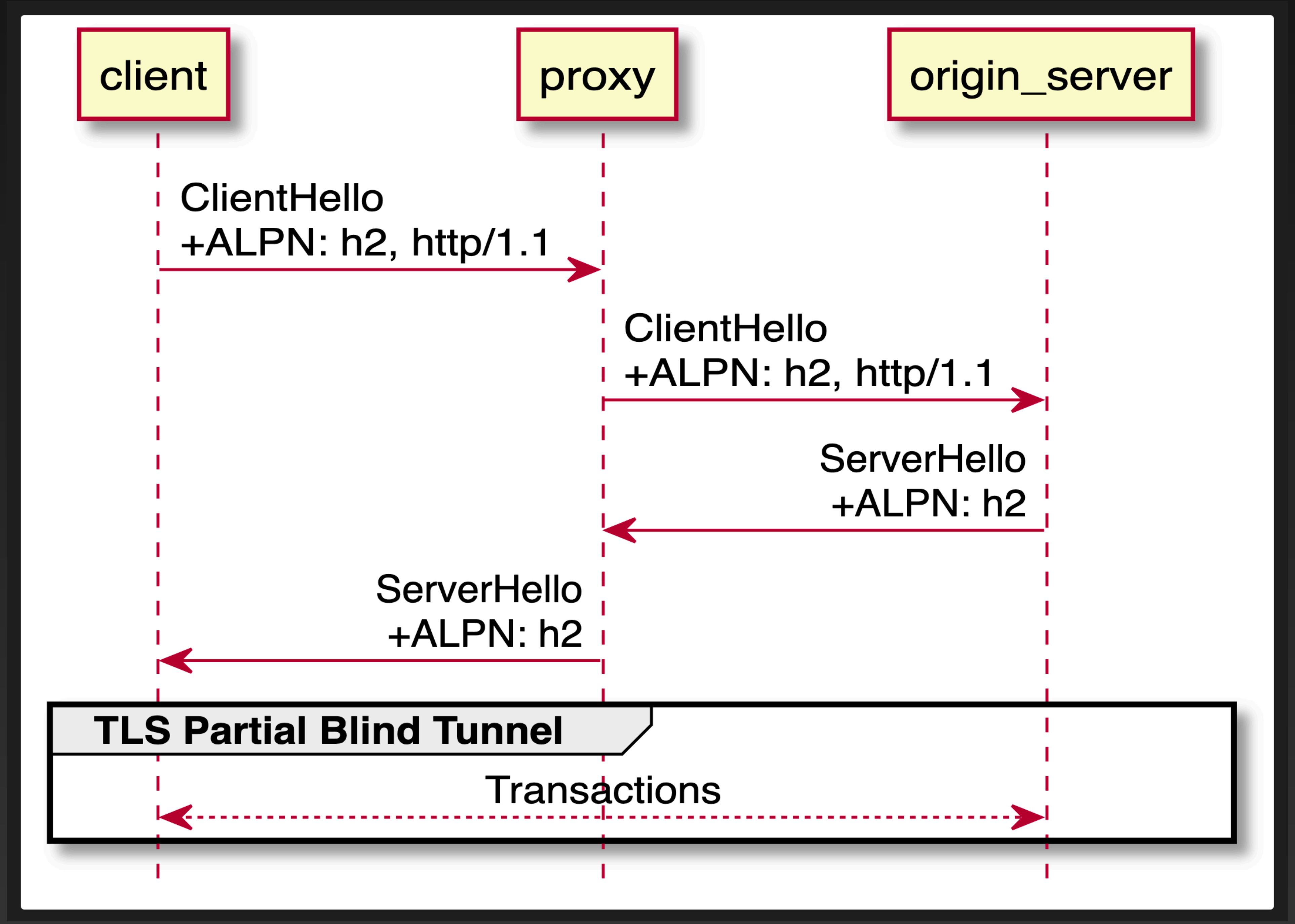


ALPN Support on TLS Partial Blind Tunnel

Current Implementation



Ideal Implementation


client
proxy
origin_server
ClientHello
+ALPN: h2, http/1.1
ClientHello
+ALPN: h2, http/1.1
ServerHello
+ALPN: h2
ServerHello
+ALPN: h2
TLS Partial Blind Tunnel
Transactions