MARCUS BOINTON @ CONFOO MONTREAL 2024

HTTP/3 & QUIC

The next step in web performance

HOW DID WE GET HERE?

- HTTP/0.9: 1991, RFC
- HTTP/1.0: 1996, RFC1945
- HTTP/1.1: 1997, RFC2068,2616
- HTTP/2: 2015, RFC7540
- HTTP/3: 2022, RFC9114

WHAT DID HTTP/2 CHANGE?

- Binary protocol
  - More compact, header compression
- Multiplexing
  - Multiple resources in a single connection, with prioritisation
- Server push
- TLS only

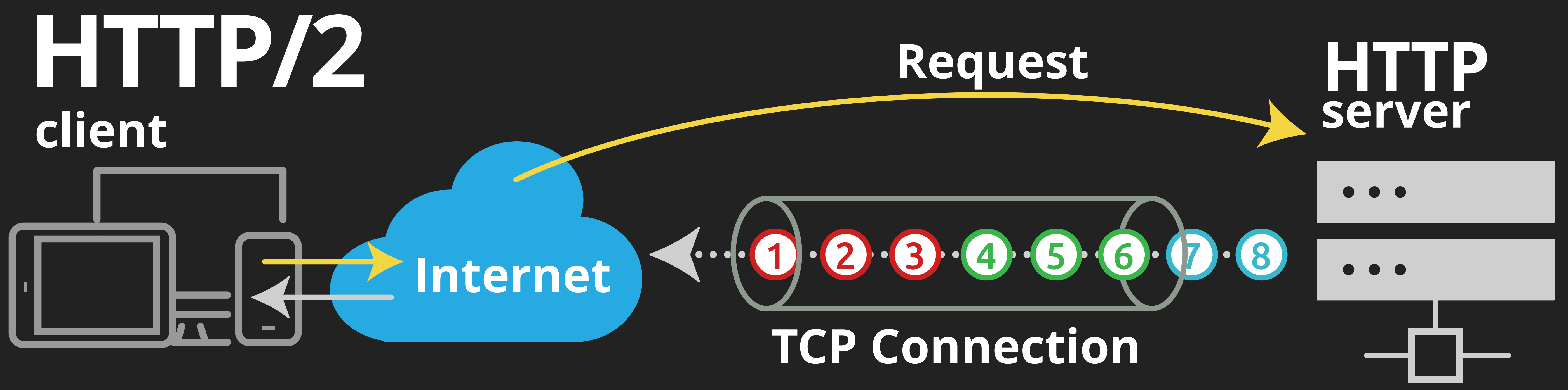
WHAT PROBLEMS DOES HTTP/2 HAVE?

- Head of line blocking
- Network switching
  - Connection re-establishment latency
- Difficult to upgrade, TCP part of host OS networking stack
- Congestion control in TCP

WHAT ARE QUIC AND HTTP/3?

- We can’t change TCP without replacing every device in the world
- Google designed QUIC as a workaround
  - A reimagining of TCP implemented over UDP
- Combines TLS and HTTP/3 into a single protocol with reduced overhead
- Implemented in userland instead of OS
- You’re using it already

HEAD-OF-LINE BLOCKING



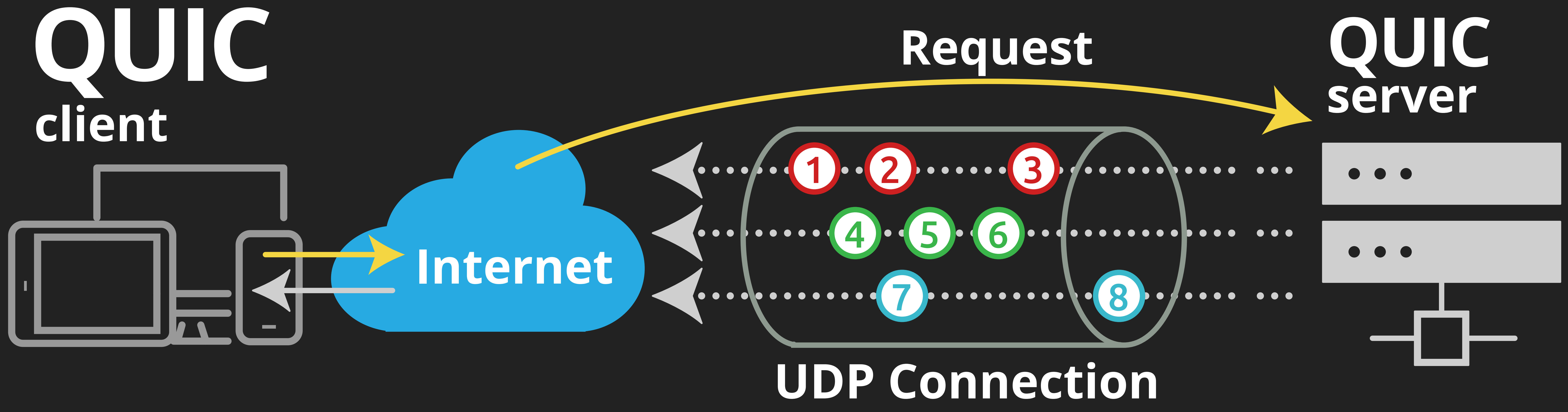
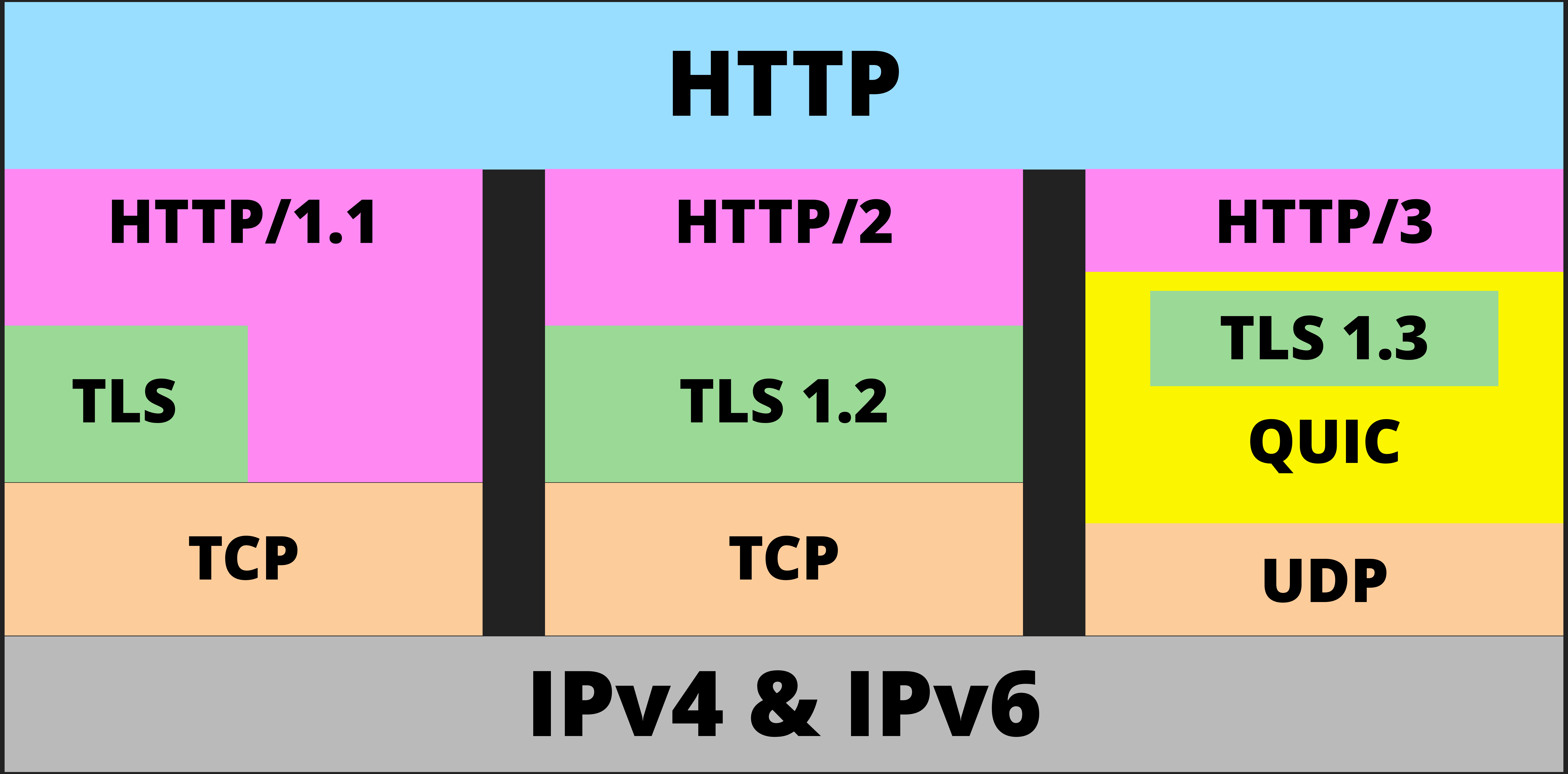
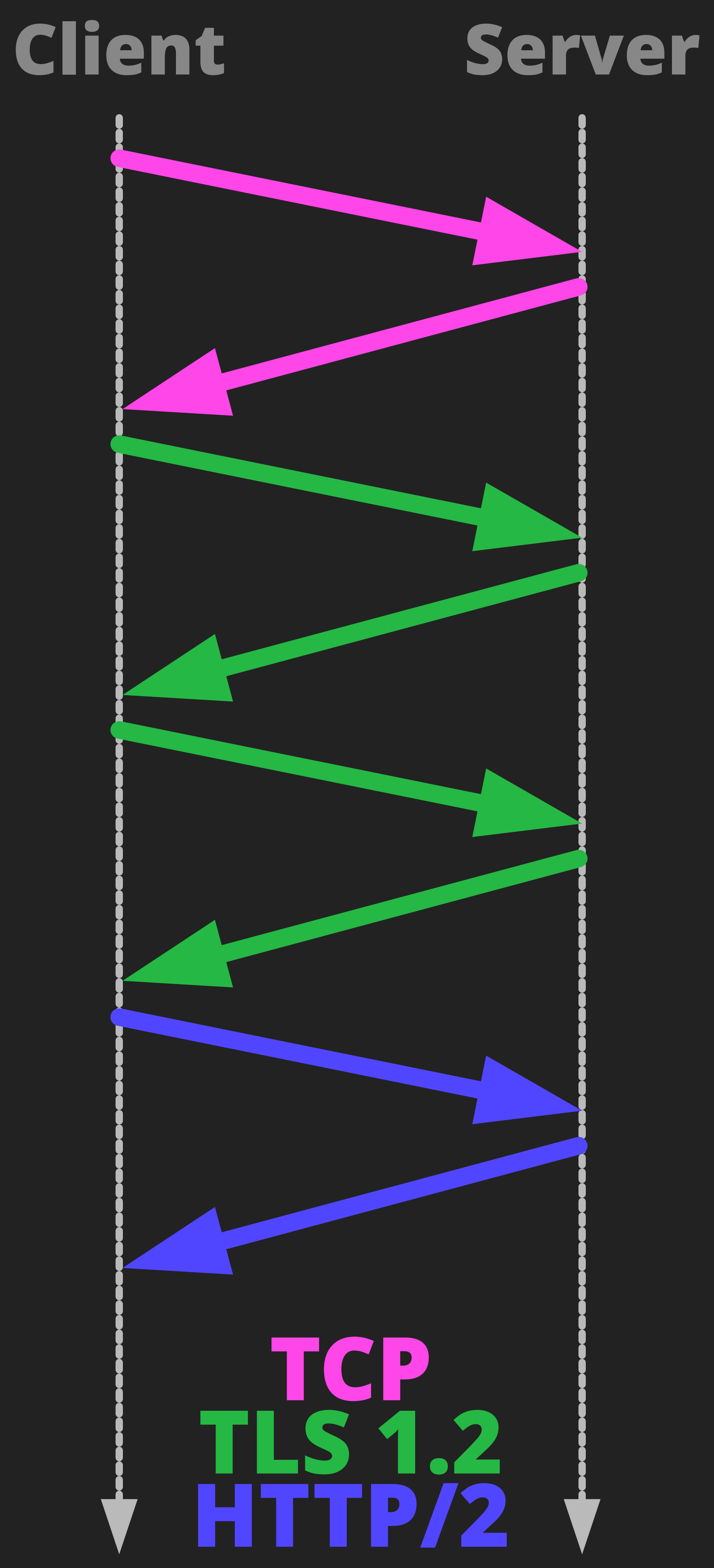


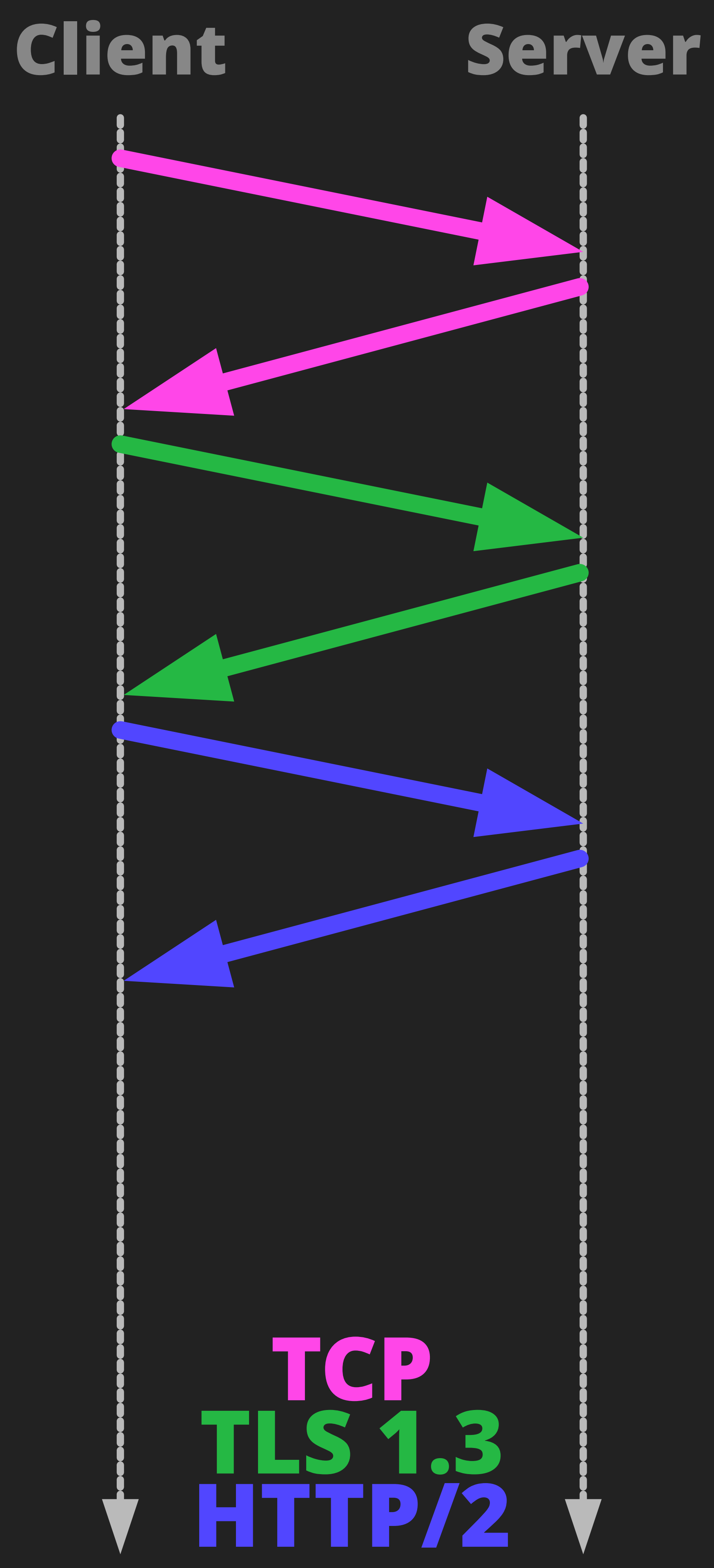
Image credit: http.dev

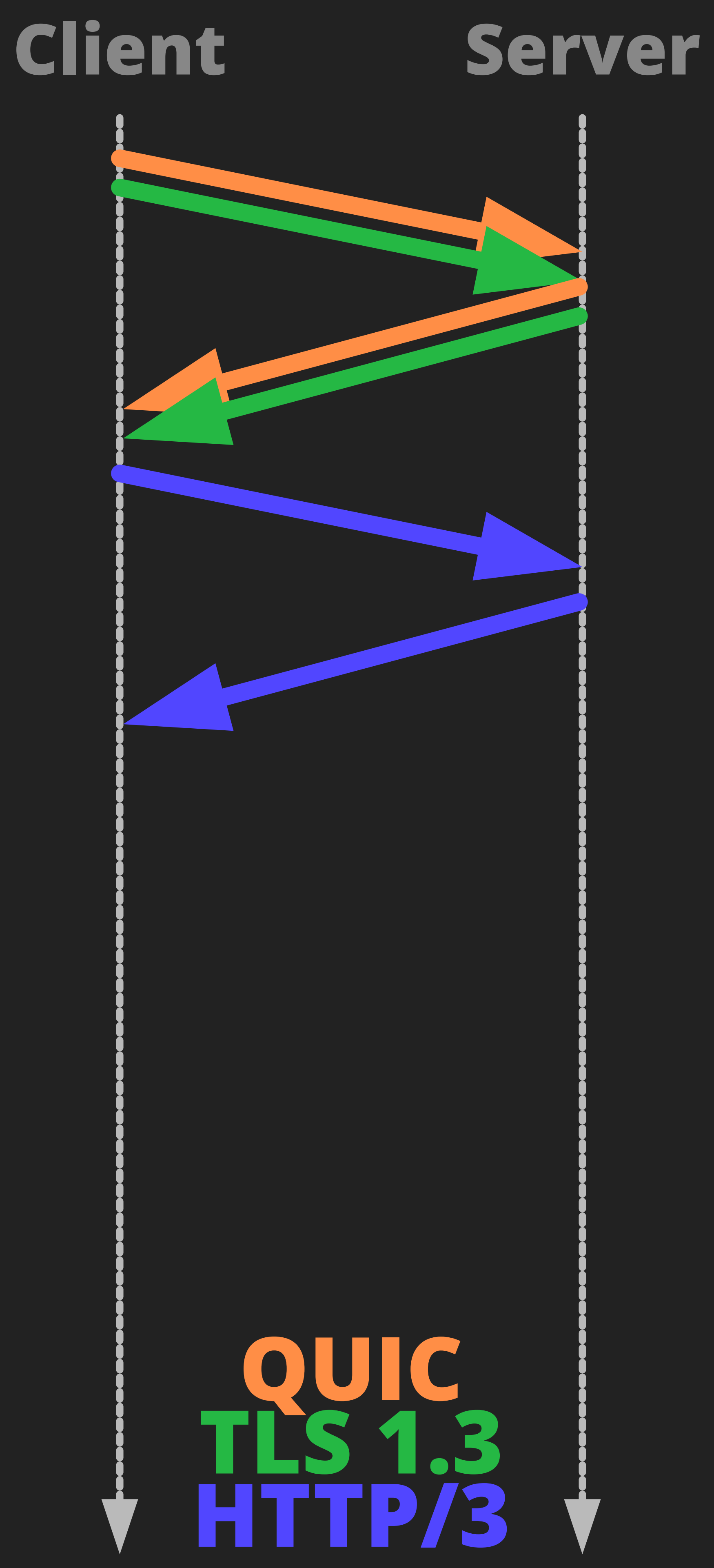
NETWORK LAYERS



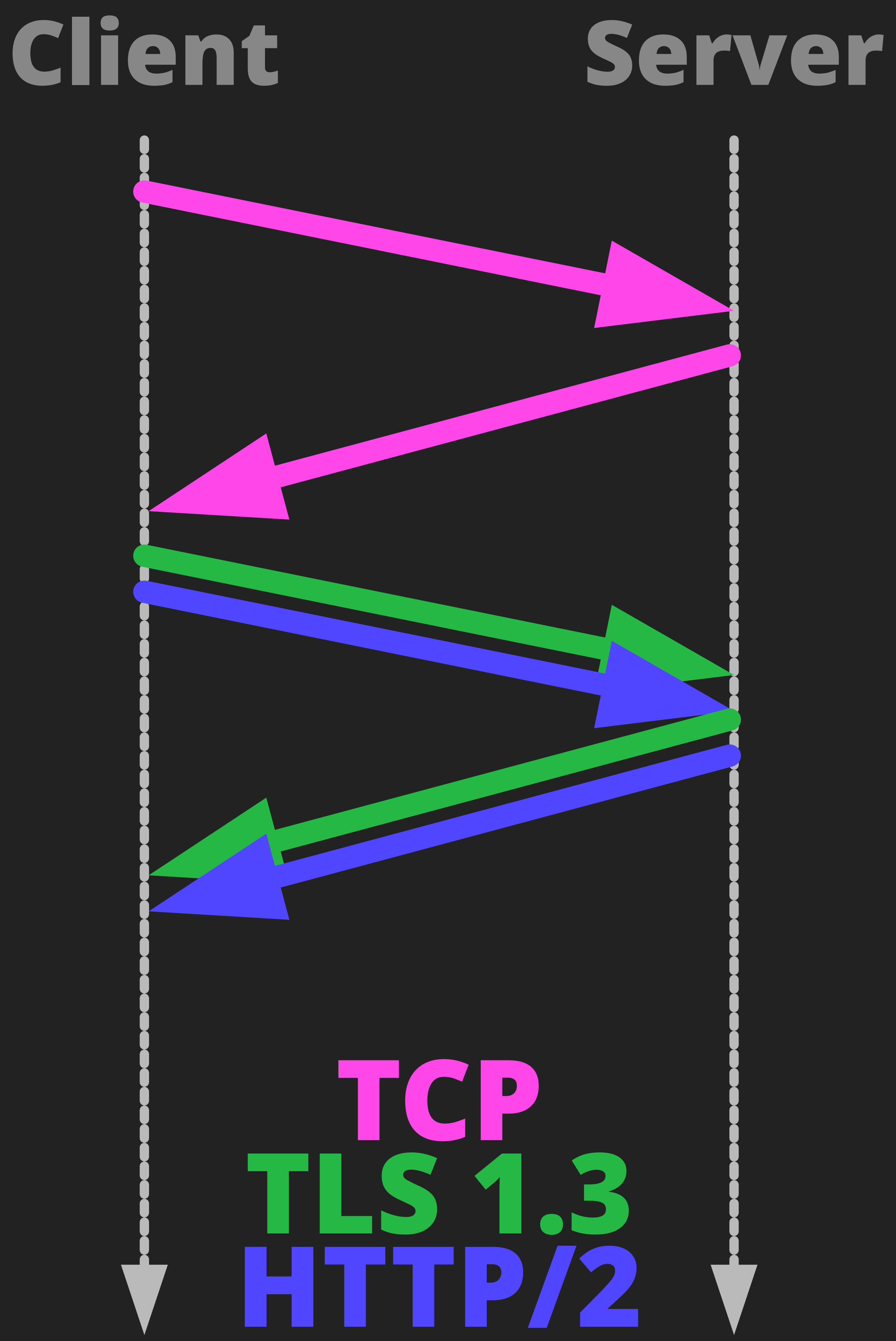
INITIAL CONNECTION

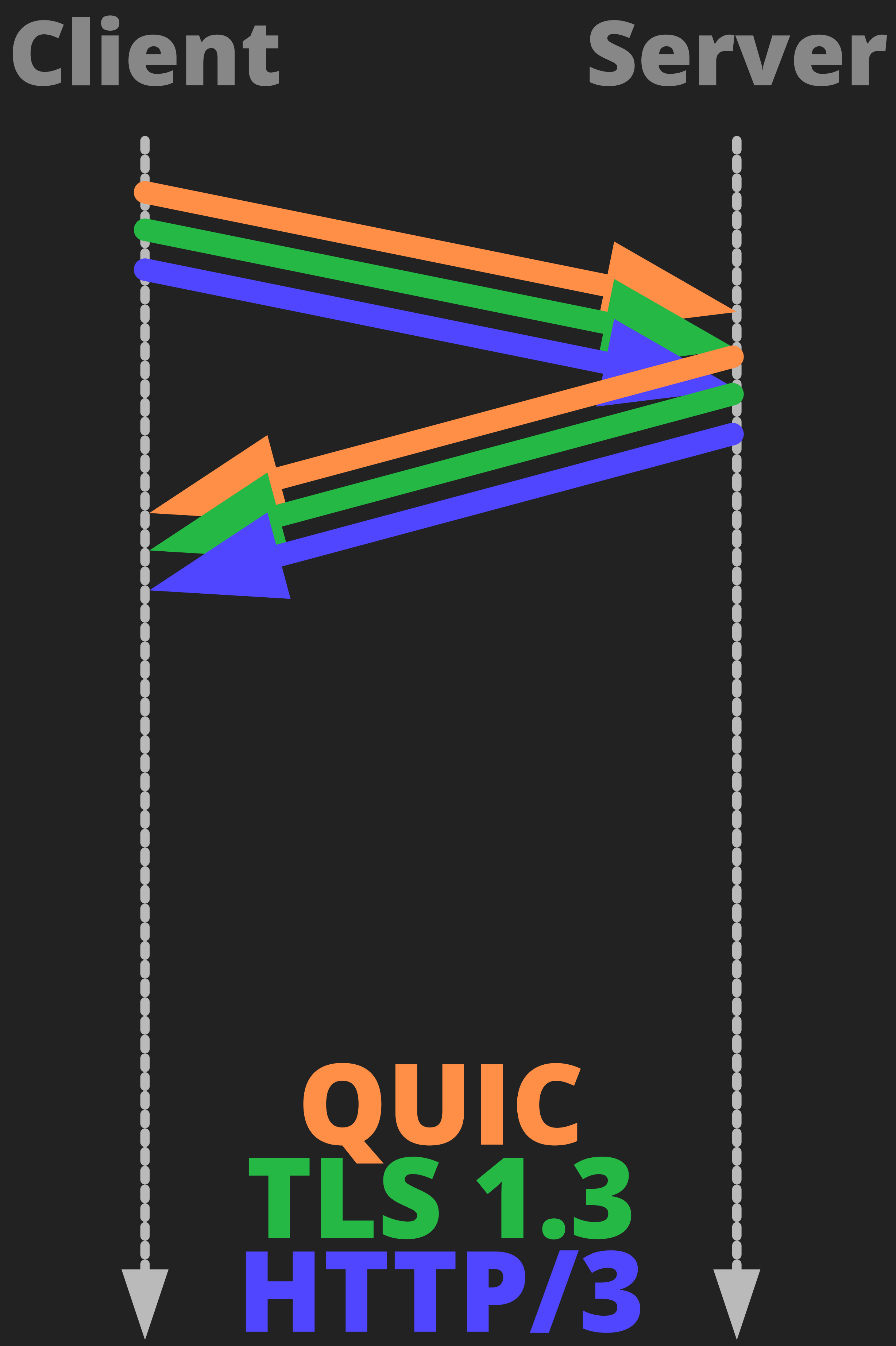






RESUMED CONNECTION





NETWORK SWITCHING

- With TCP, switching networks requires re-establishing connections
  - IP & port as identifier
  - Breaks session resumption each time
- QUIC uses a connection ID that moves between networks
  - More likely for session resumption to happen
  - Privacy? Cycles through a list of random IDs

HTTP/3 COMPRESSION

- HTTP/2 uses HPACK
  - Relies on packets arriving in order
    - Can cause HOLB
- HTTP/3 uses QPACK
  - Slightly lower compression ratios
  - Avoids HOLB

HTTP/3 IMPLEMENTATIONS

- Client, servers, libraries
- Clients: Chrome, Edge, Firefox, Safari (iOS 15)
- Servers: Litespeed, Caddy, Nginx, HAProxy
  - Not Apache!
- Libraries: h2o, nghttp3, libcurl, openssl 3.2.0
- Cloud services: CloudFlare
- All in userland, so not so subject to OS stagnation

HOW TO DEPLOY HTTP/3?

- How does a client know a server supports HTTP/3?
- Server can tell clients what protocols it can use
  - Alt-Svc header
  - DNS SVCB record

ALT-SVC HTTP HEADER

- RFC7838
- "Alternative service"
- Similar to HSTS for HTTPS
- `Alt-Svc: h3=":443"; ma=3600, h2=":443"; ma=3600`

SVCB DNS RECORDS

- “Service binding” records, RFC9460
- Saves an HTTP request, at the cost of a DNS lookup
- `example.com 3600 IN HTTPS 1 . alpn="h3,h2"`
- `example.com 3600 IN HTTPS 1 . alpn="h3,h2" ipv4hint="192.0.2.1" ipv6hint="2001:db8::1"`
- `example.com 3600 IN HTTPS 1 example.net alpn="h3,h2"`
- `example.com 3600 IN HTTPS 2 example.org alpn="h2"`

NGINX CONFIG EXAMPLE

```
server {
  listen 443 ssl;
  listen [::]:443 ssl;
  listen 443 quic;
  listen [::]:443 quic;
  http2 on;
  add_header Alt-Svc 'h3=":443"; ma=86400';
  ...
```

UFW APPLICATION CONFIG

```
[Nginx QUIC]
title=Web Server (Nginx, HTTP + HTTPS + QUIC)
description=Small, but very powerful and efficient web server
ports=80,443/tcp|443/udp
```

Enable with:

```
ufw allow from any to any app "Nginx QUIC"
```

SECURITY UPGRADE

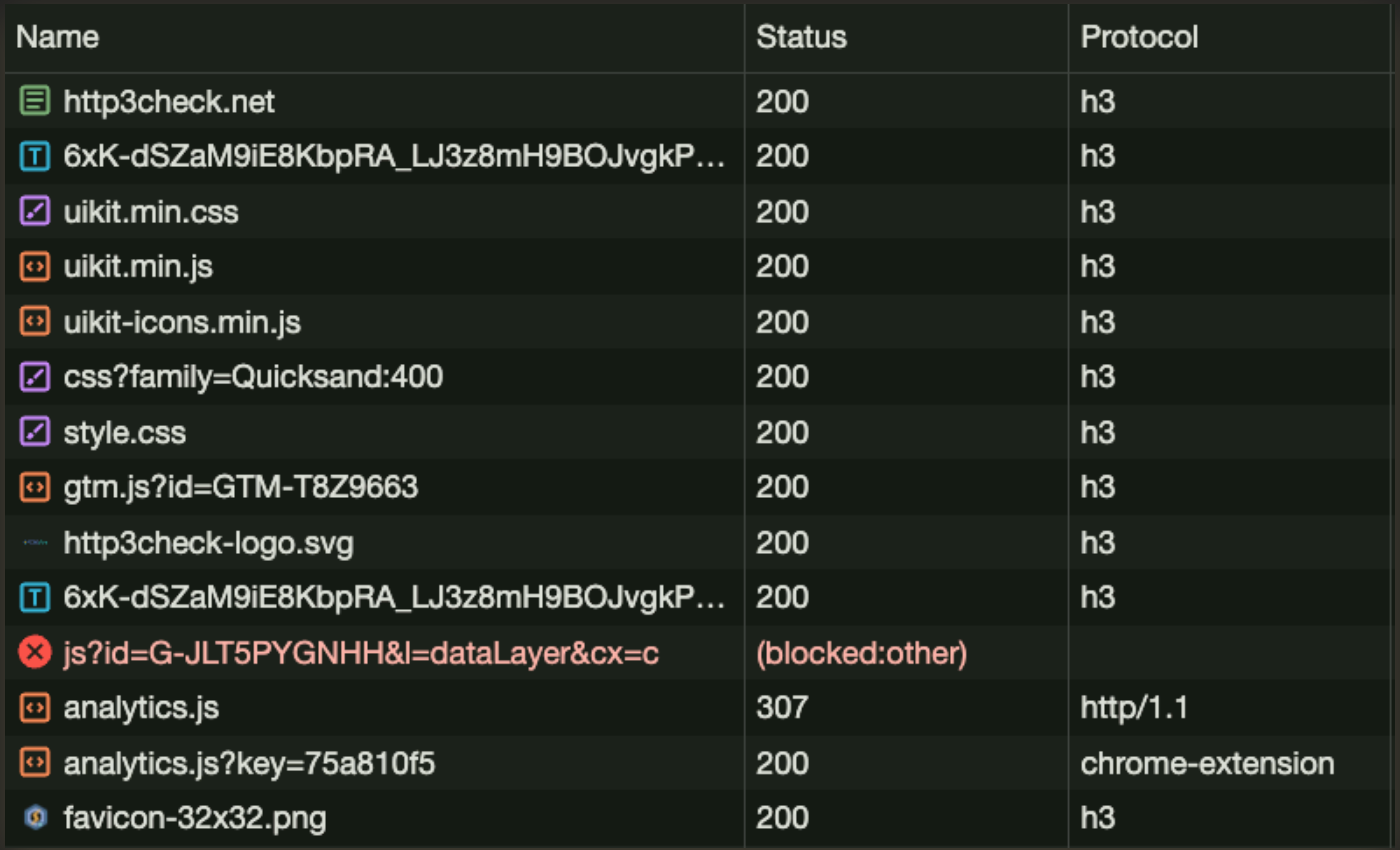
- QUIC requires TLS 1.3
  - Lower overhead
  - No weak cipher suites, KX, or hashes
  - Forward secrecy
  - Downgrade detection
- More is encrypted

OPTIMISING FOR HTTP/3

- The same as HTTP/2
- Only use a few domains
- Don’t worry about bundling
  - Request count doesn’t really matter
- Use defer / preload / async
- Use lazy loading

TESTING HTTP/3

- https://http3check.net/
- "HTTP Indicator" Chrome extension
- Dev tools will show "h3" as the protocol; right-click table header to enable
- Remember browser will connect via HTTP/2 first



Name	Status	Protocol
http3check.net	200	h3
6xK-dSZaM9iE8KbpRA_LJ3z8mH9BOJvgkP...	200	h3
uikit.min.css	200	h3
uikit.min.js	200	h3
uikit-icons.min.js	200	h3
css?family=Quicksand:400	200	h3
style.css	200	h3
gtm.js?id=GTM-T8Z9663	200	h3
http3check-logo.svg	200	h3
6xK-dSZaM9iE8KbpRA_LJ3z8mH9BOJvgkP...	200	h3
js?id=G-JLT5PYGNHH&l=dataLayer&cx=c	(blocked:other)	
analytics.js	307	http/1.1
analytics.js?key=75a810f5	200	chrome-extension
favicon-32x32.png	200	h3

IS IT ACTUALLY FASTER?

- It depends
- It's difficult to measure
- Biggest payoff will be in situations where its features make a difference:
  - Low-bandwidth
  - High congestion
  - High latency
  - Network switching

HTTP/3 PROBLEMS

- Networks might block UDP
- Version discovery latency
- It’s new, so will have more bugs
- More is encrypted, makes it harder to diagnose network issues
  - Not so corporate friendly

THE FUTURE OF QUIC

- QUIC deliberately dynamic spec
  - Version 2 (RFC9369) essentially unchanged
  - Mainly to exercise ability to update
  - Prevent "ossification", like MIME 1.0
- Pluggable congestion control
- Other protocols over QUIC - DNS, SSH

FURTHER READING

- https://www.debugbear.com/blog/http3-quic-protocol-guide
- https://http.dev/3
- https://www.csoonline.com/article/569541/6-ways-http-3-benefits-security-and-7-serious-concerns.html
- Robin Marx at SmashingConf: https://vimeo.com/725331731

QUESTIONS?

THANK YOU

- `@Synchro@phpc.social`
- `@SynchroM`
- **Synchro** on GitHub and Stack Overflow
- Open to job offers!