Deploying and Debugging HTTP/3

and other cautionary tales

Robin Marx
@programmingart

Akamai


Dopey
Sneezy
Happy
Grumpy
Doc
Bashful
Sleepy


HUNGRY

Shared state between Client and Server

Common occurrence:

- Client establishes a **first** connection to the server
- They exchange ***some*** state
- That state can be re-used to improve **the next connection**

Several use cases:

- Session resumption (e.g., session tickets)
- Authentication/SSO (e.g., JWT)
- Address validation
- Connection parameters
- …

You don’t ACTUALLY want to keep state at server

Shared state explosion with distributed backends:

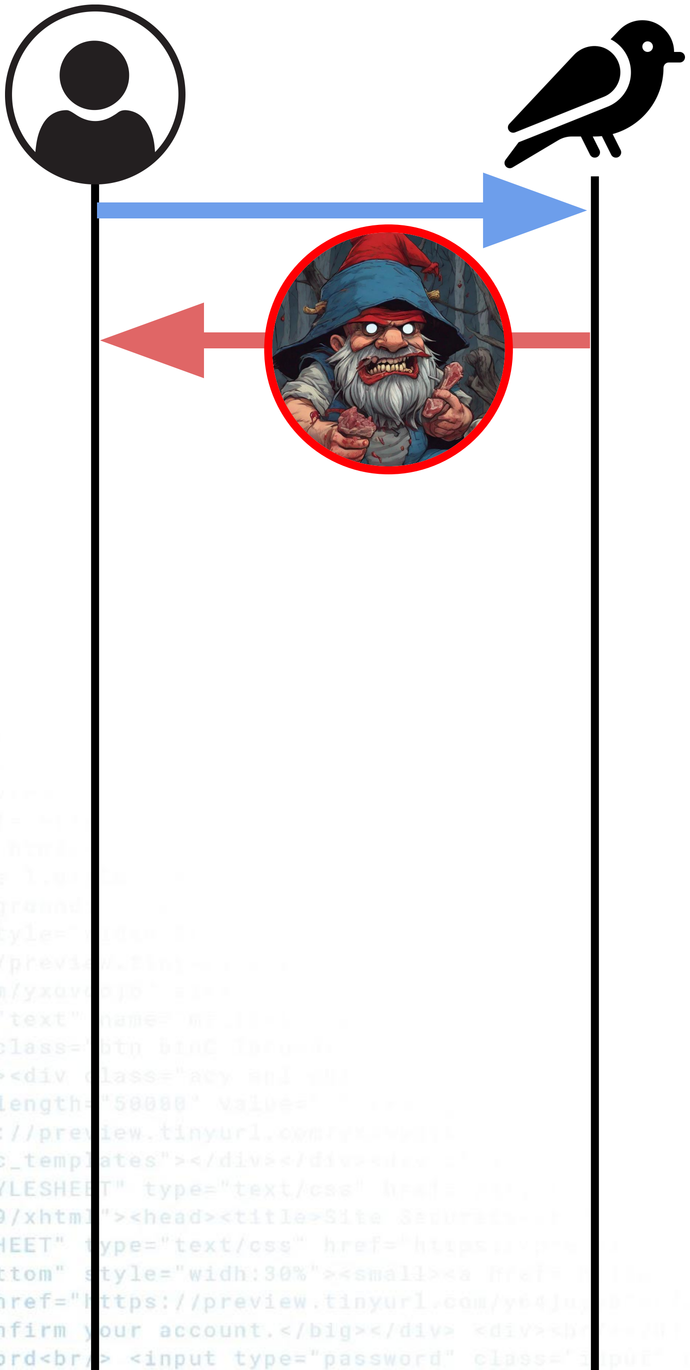
- All either need a copy
- OR need to query shared database = slow

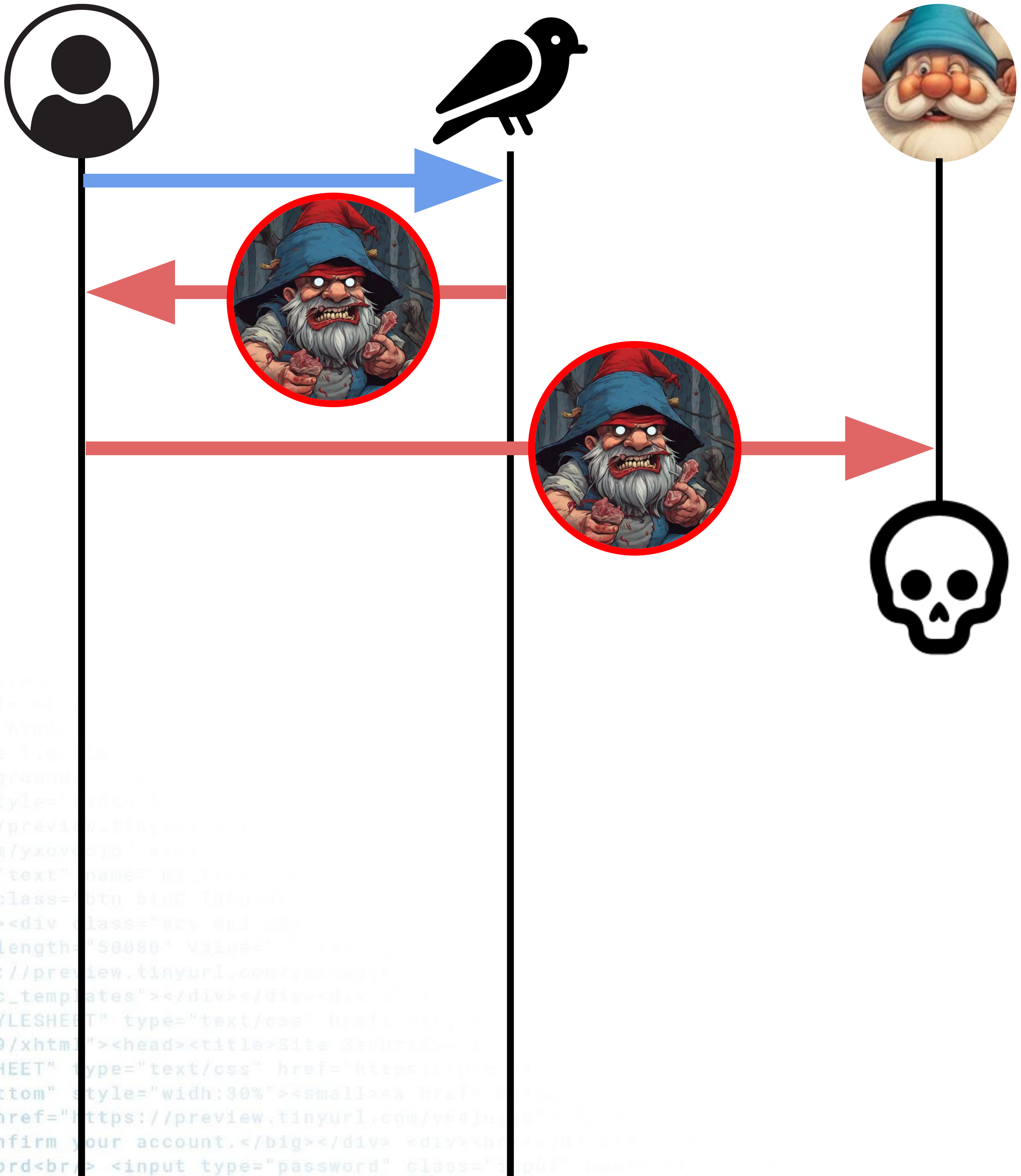
Actual solution:

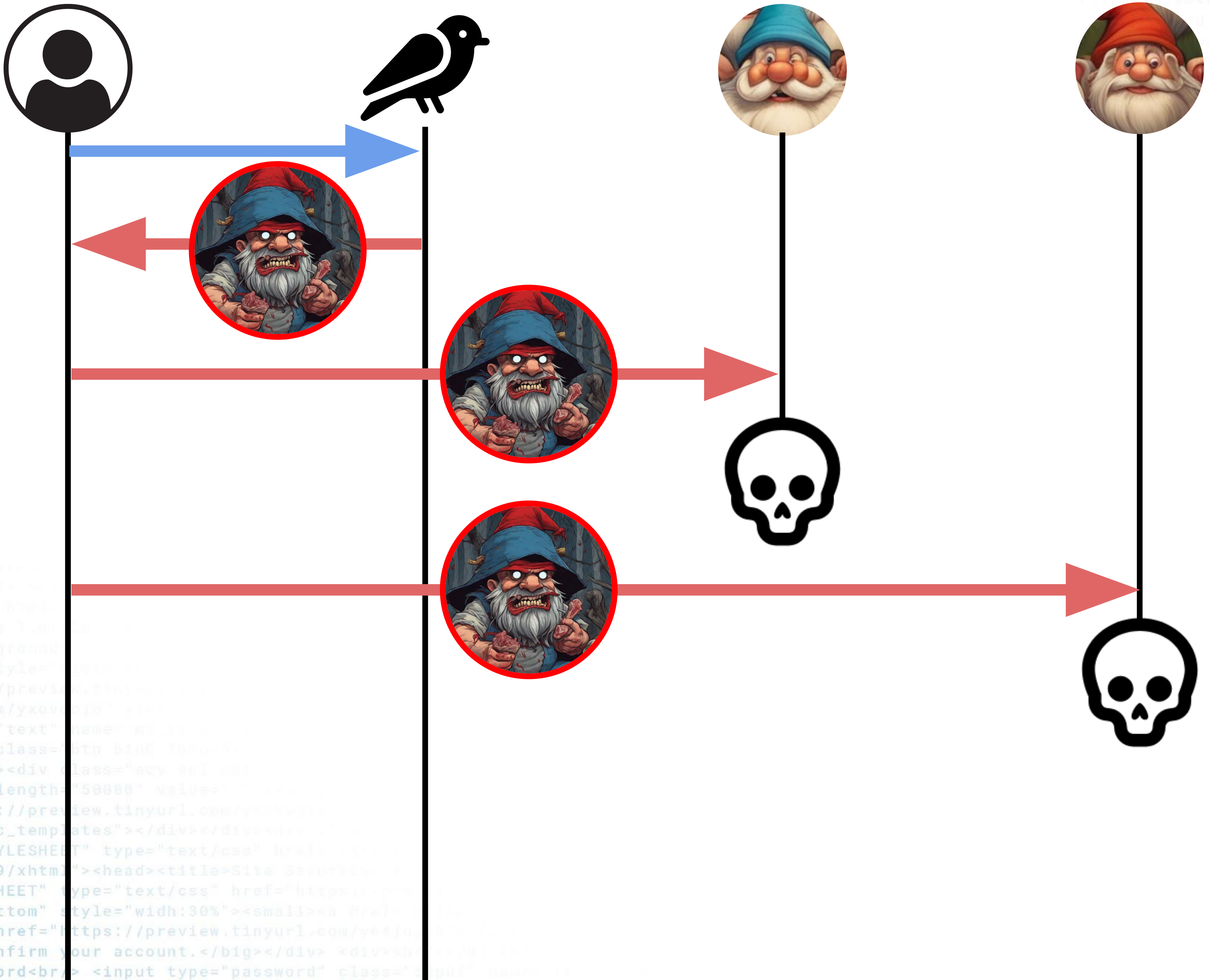
- Encrypt state on server with **private key**
- Send blob to **client to store**
- Client sends blob with next connection
- ⇒ Servers “only” need **shared private key**!

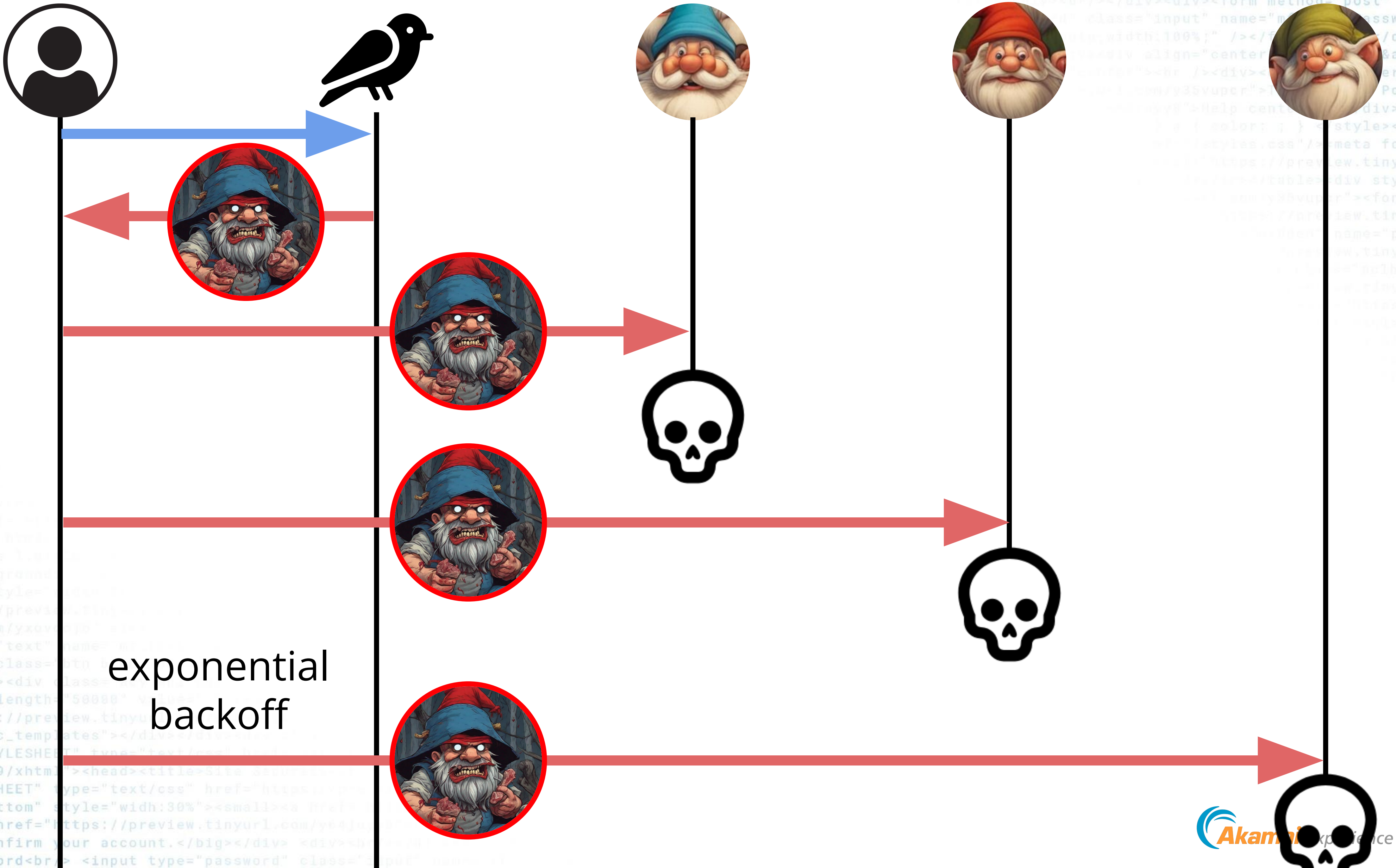
What happens…

if the blob generated by new server code, causes old server code to CRASH?








exponential
backoff

How would you handle this?

Rollback doesn't "fix" it

- Stops handing out new Hungries, but existing ones will still be used

Just deploy new version to 100%?

- I see you like to live dangerously

How would you handle this?

Rollback doesn't "fix" it

- Stops handing out new Hungries, but existing ones will still be used

Just deploy new version to 100%?

- I see you like to live dangerously

Actual solution they used: BIG RED BUTTON!

- Disable QUIC on the fleet for x **days**

"New type of bug" : **Contagion Bug**

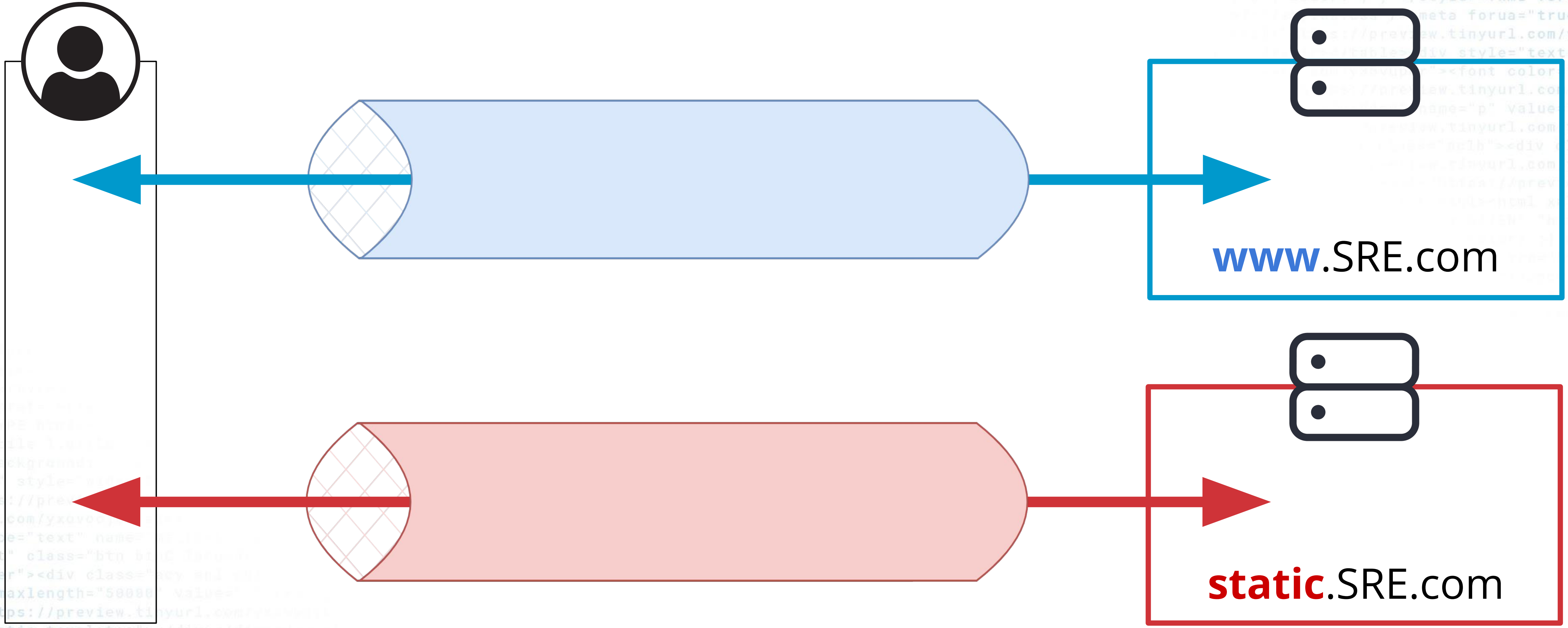
- Contaminated client state, impossible to purge

Firewall returns 403 Forbidden for HTTP/3,

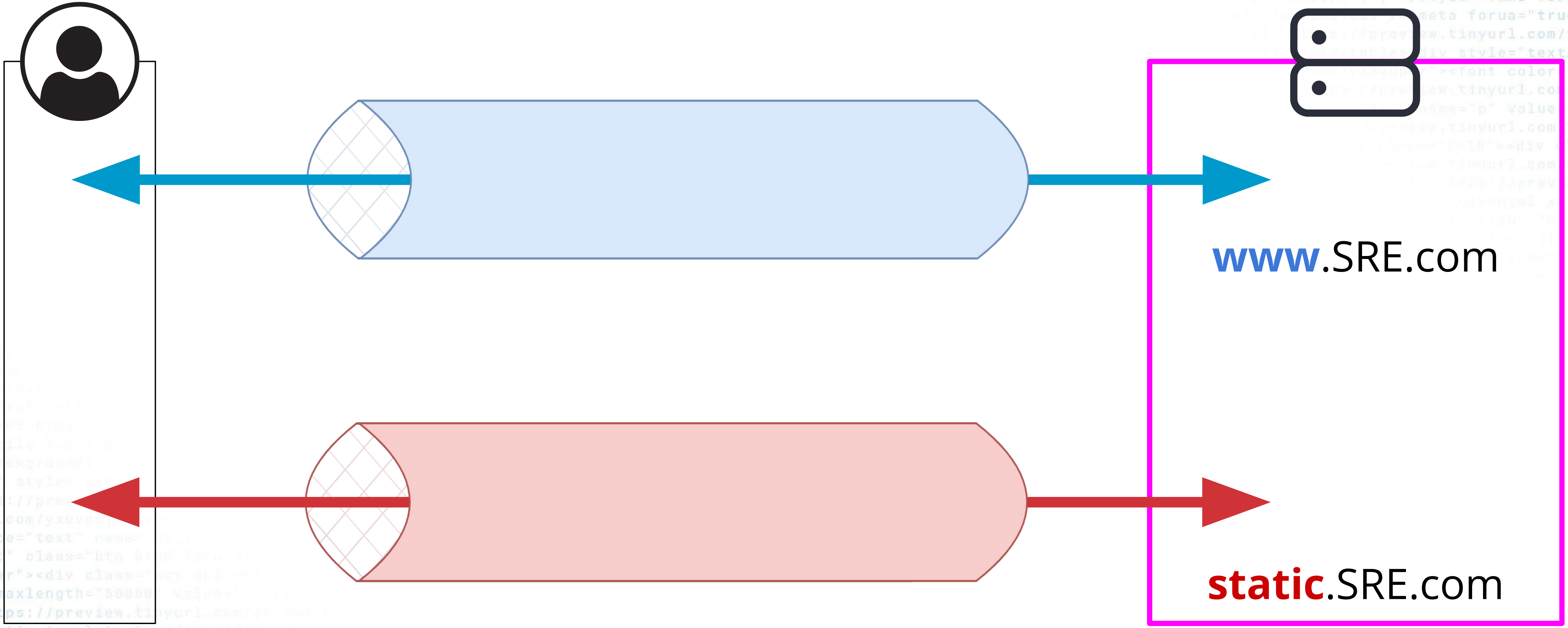
on a domain that has H3 disabled…

Site Site, Different (sub)Domains

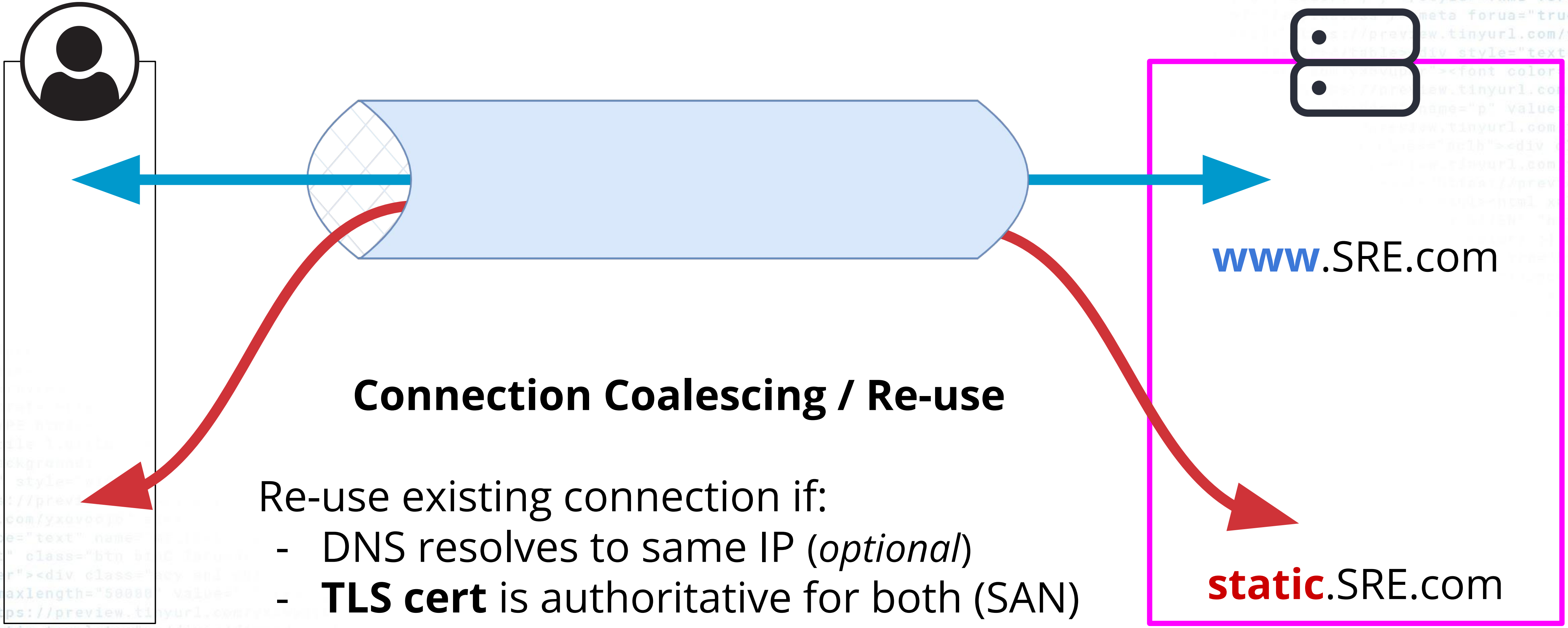
Different Domains, Different Connections



Different Domains, Same “Server”



Different Domains, Common Connection

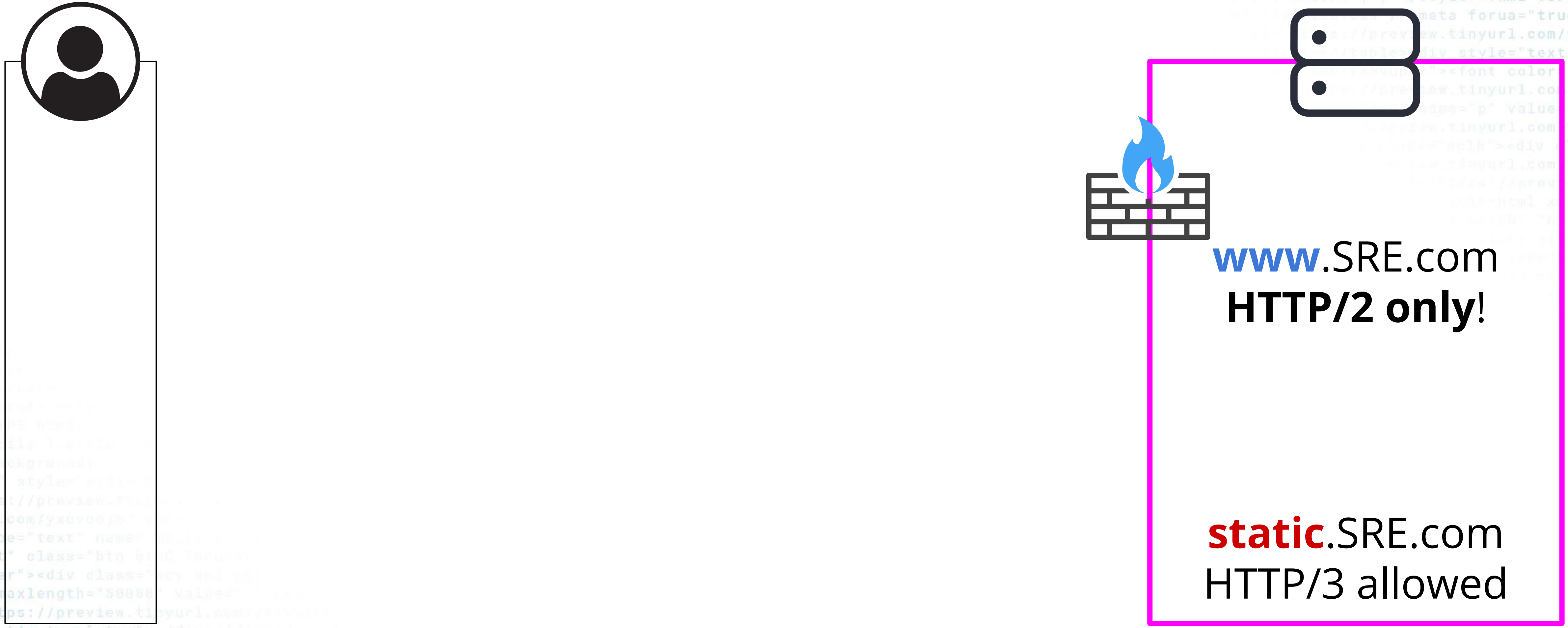


Connection Coalescing / Re-use

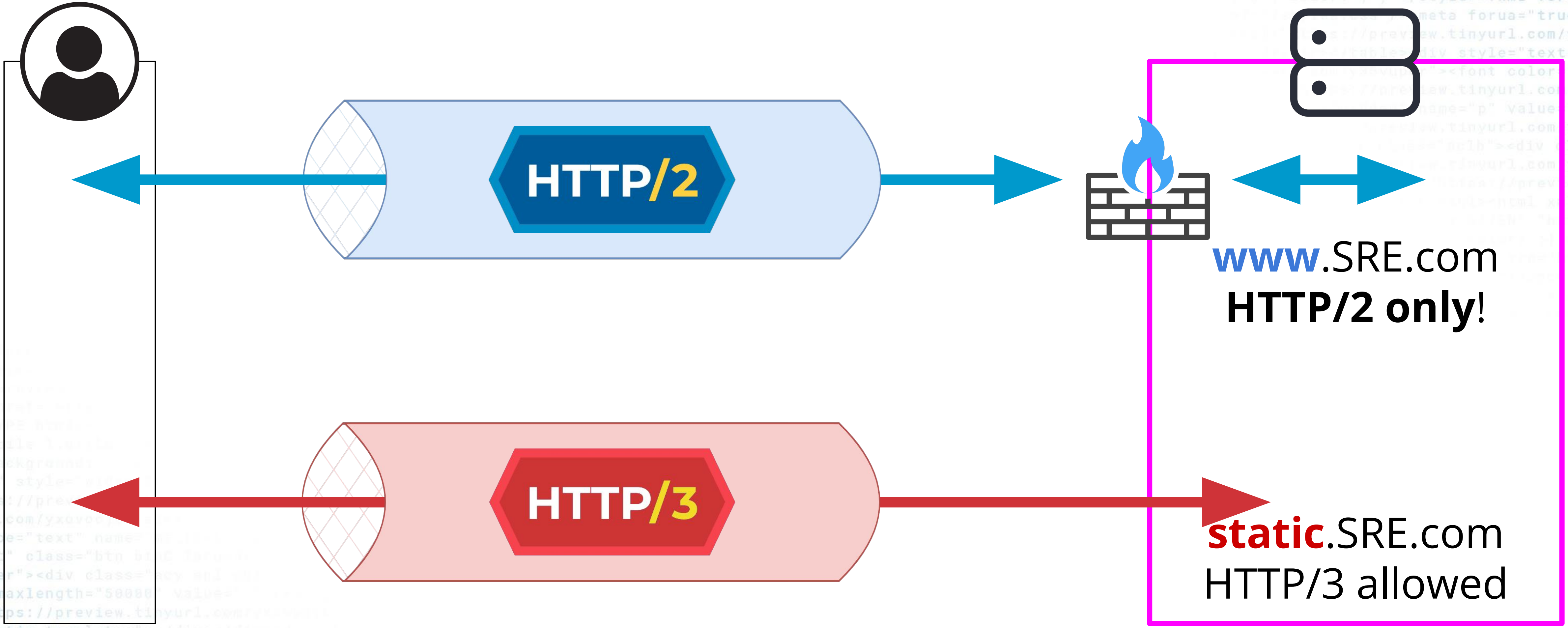
Re-use existing connection if:

- DNS resolves to same IP (*optional*)
- **TLS cert** is authoritative for both (SAN)

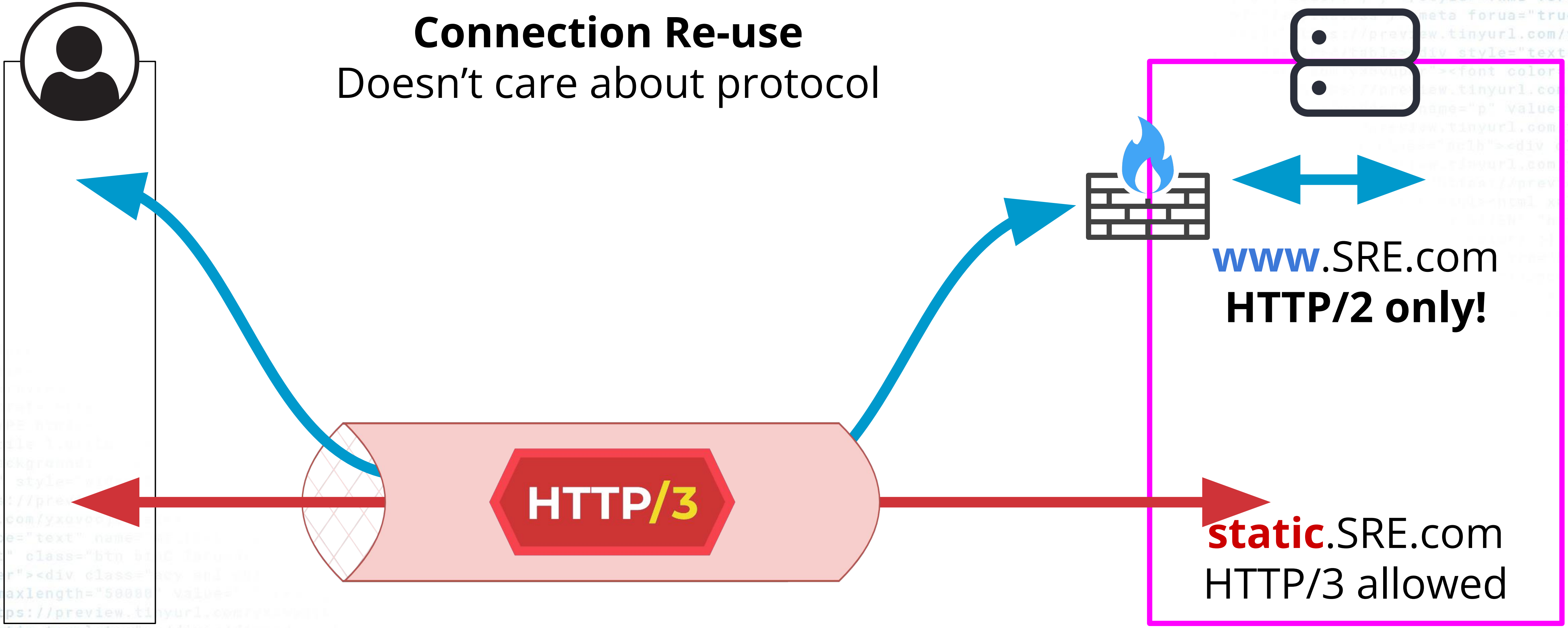
Same Server, Different Protocols



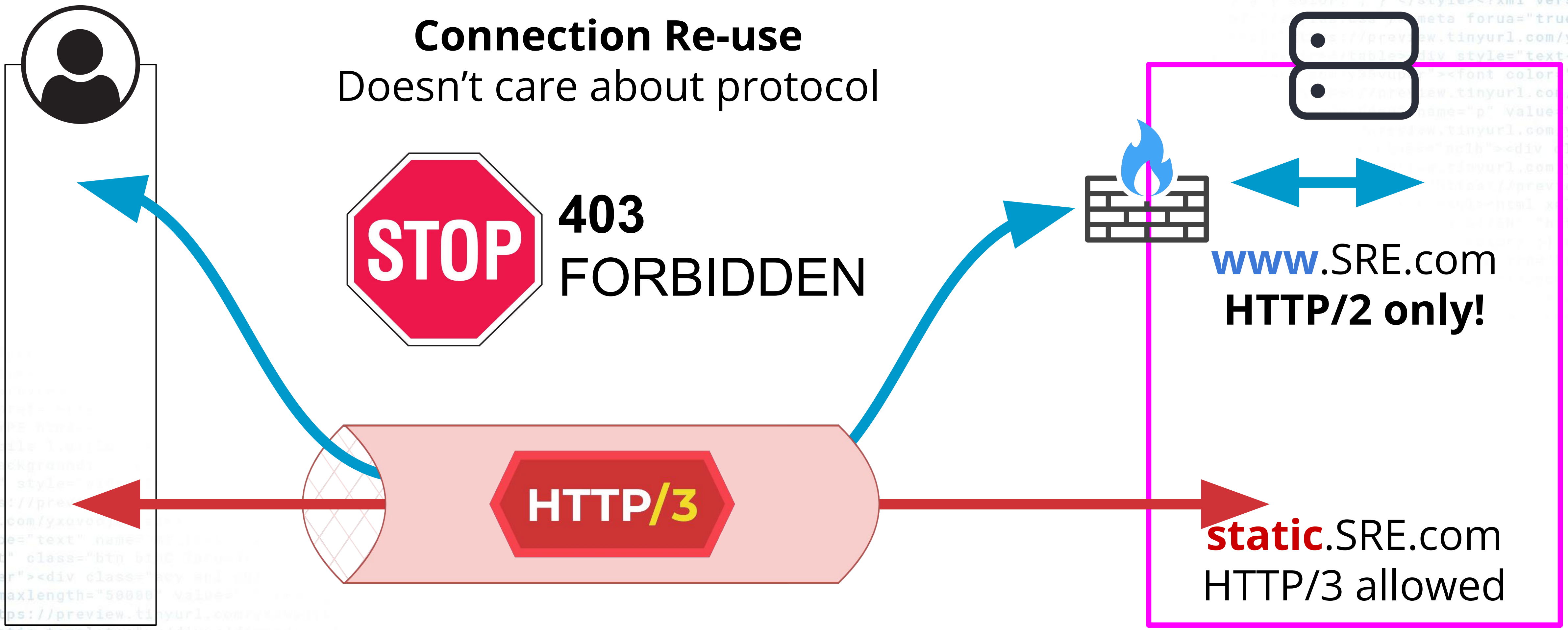
Different Protocols, Different Connections



Different Protocols, Common Connection



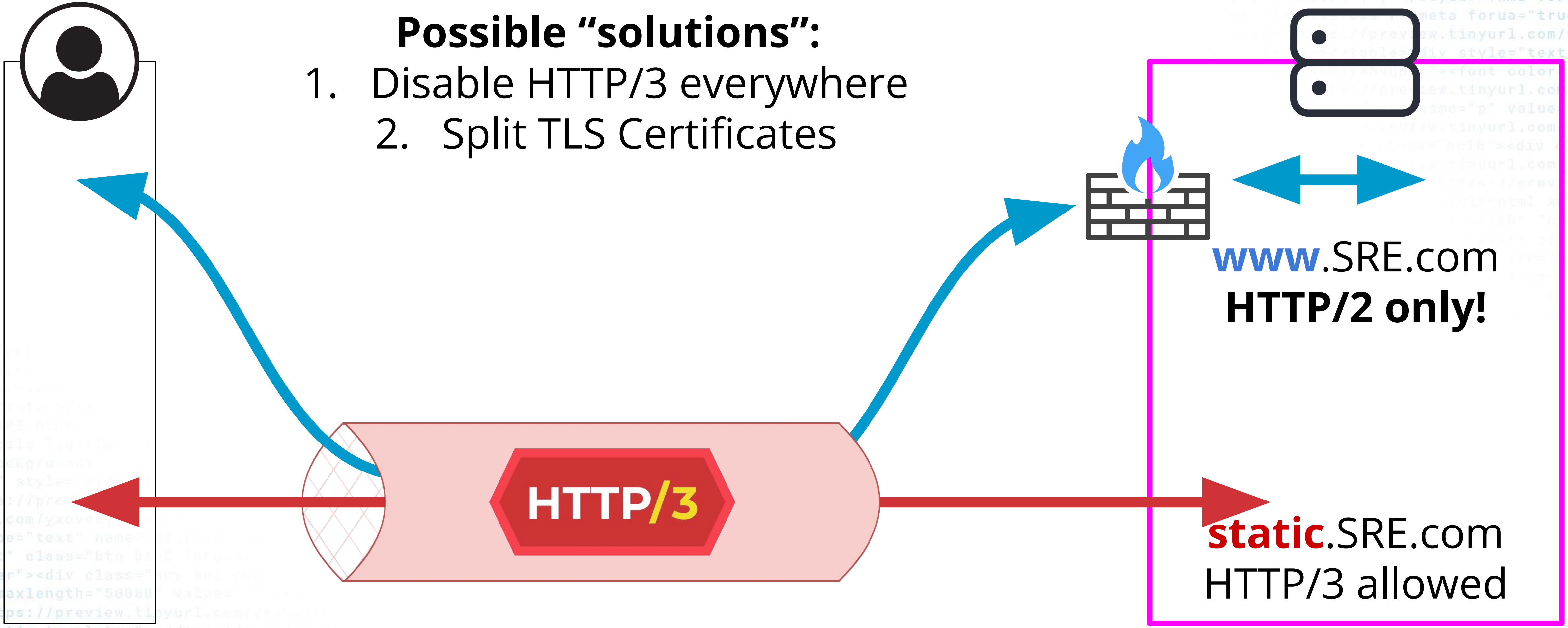
Different Protocols, Common Connection



Different Protocols, Common Connection

Possible "solutions":

1. Disable HTTP/3 everywhere
2. Split TLS Certificates



Multi-CDN with fast failover OR live traffic split

slow.ly

HTTP/2

+

Akamai

HTTP/3

Only on

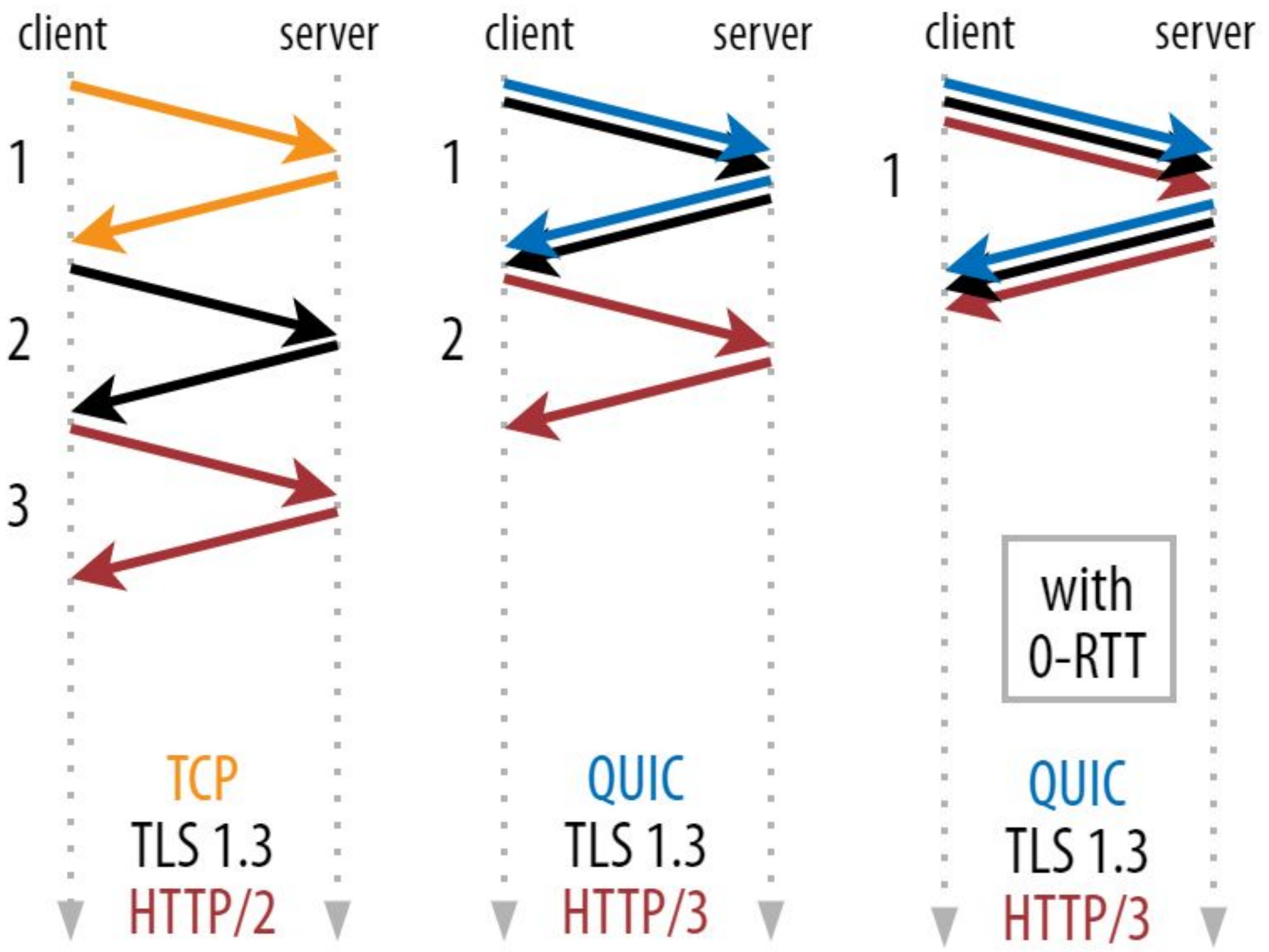
HBO max

November
2024

HTTP/3 is 50% faster than HTTP/2,

but it should be 33% or 66%...

QUIC hates wasting time on handshakes



https://www.smashingmagazine.com/2021/08/http3-performance-improvements-part2/

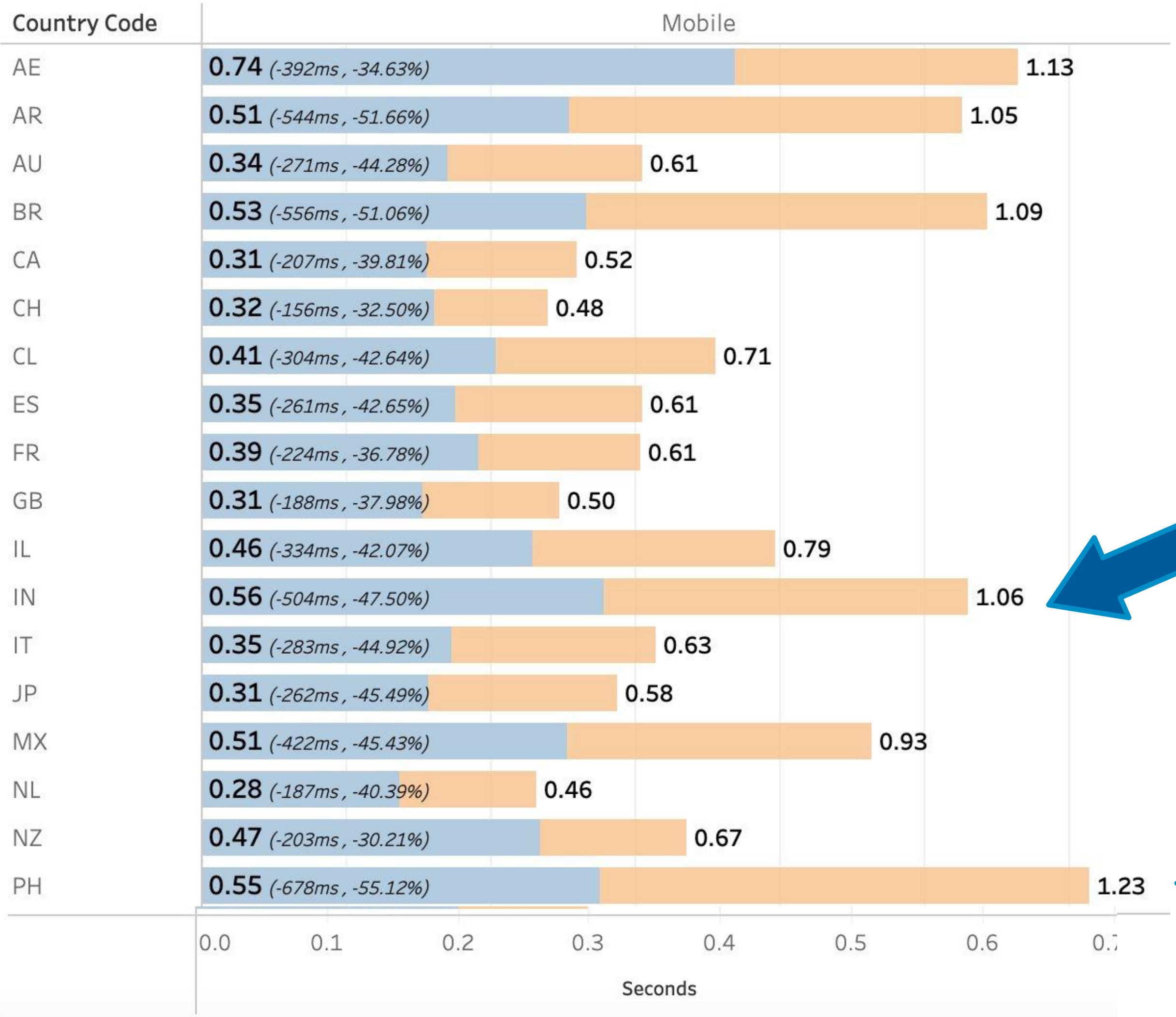
Measured Impact (2021)

p75 Time to First Byte

HTTP/2

HTTP/3

WiX

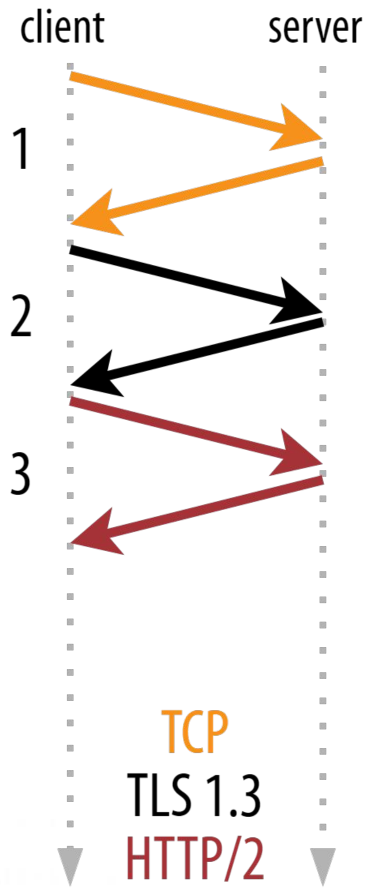


India:
47.50% faster!
1000ms to 560ms!

Philippines:
55% faster!
1230ms to 550ms!

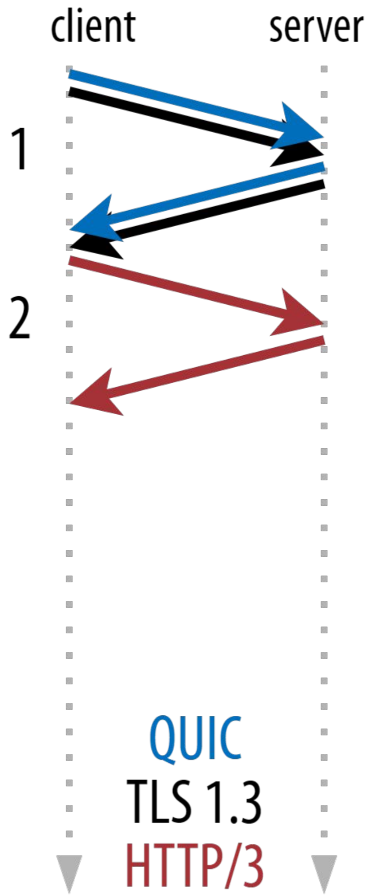
Akamai Experience the Edge

Hold on a minute…

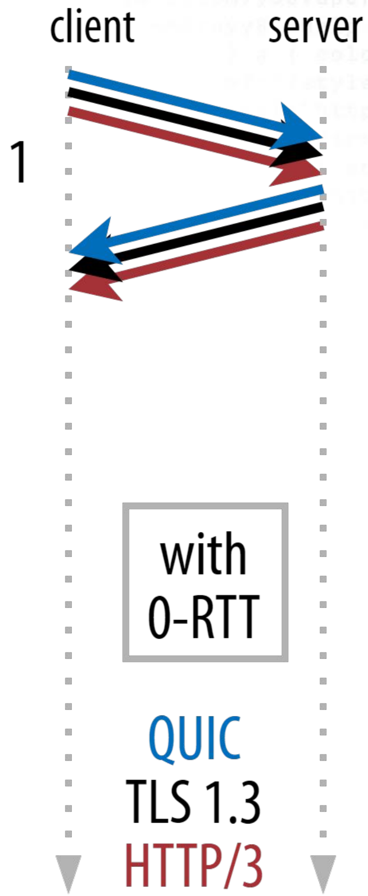


100%

VS



~33% faster



~66% faster

Yet we see ~50%?

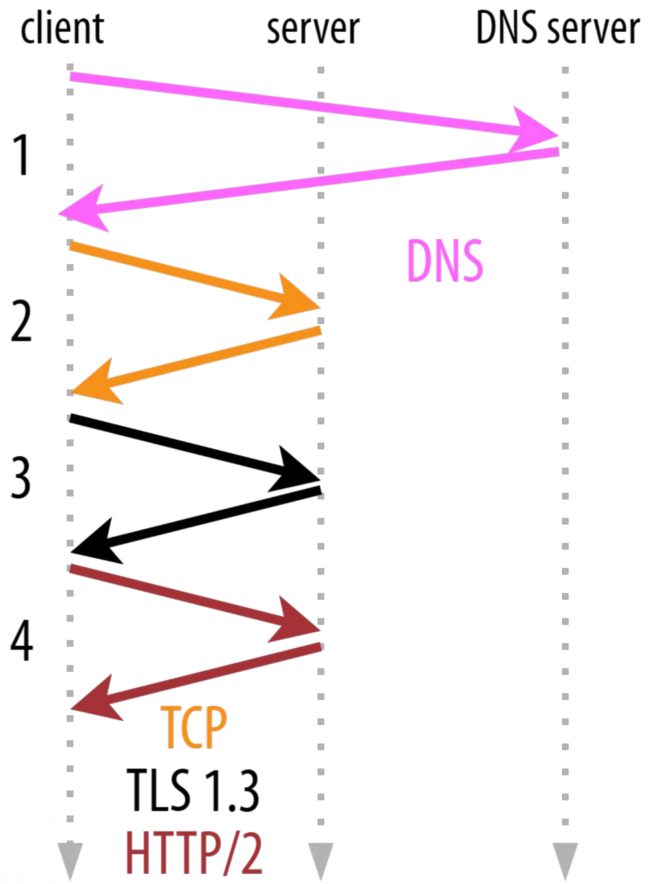
https://www.smashingmagazine.com/2021/08/http3-performance-improvements-part2/

Akamai Experience the Edge

IT'S ALWAYS DNS

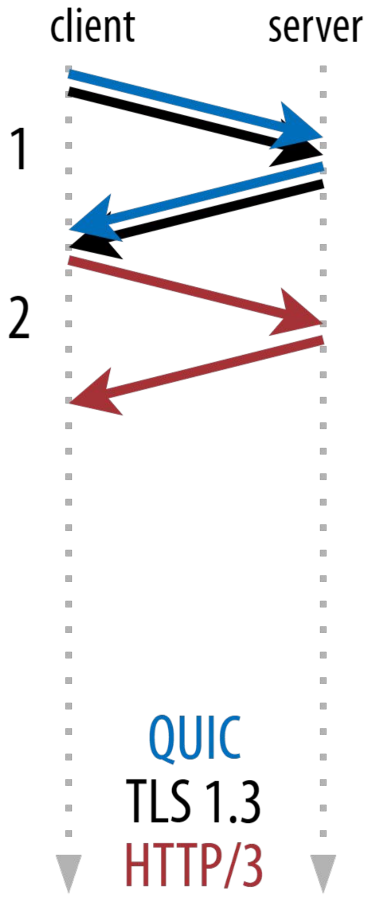
Unless it's BGP...

Hold on a minute…



100%

VS



~50% faster

= unfair comparison

https://www.smashingmagazine.com/2021/08/http3-performance-improvements-part2/

Browser only do HTTP/3 after *discovery*

For a *new* hostname browser doesn't know yet:

1. Browser requests the page over HTTP/1 or 2

2. Server sends back "**alternative services**" header

   ▼ **Response Headers**

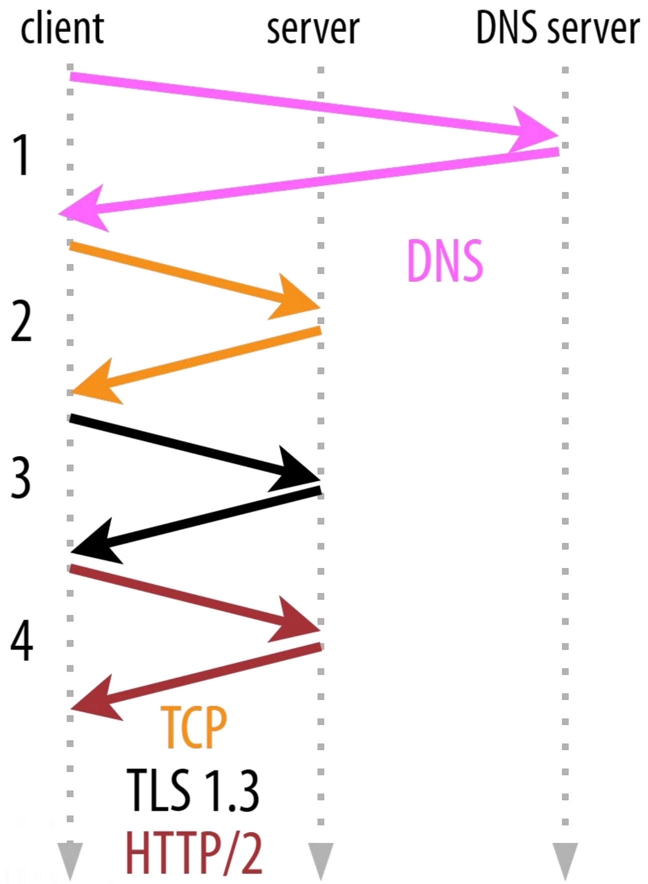
   alt-svc: h3=":443"; ma=86400

   cache-control: private, no-cache, no-store, must-revalidate

3. Browser stores alt-svc info in alt-svc "cache"

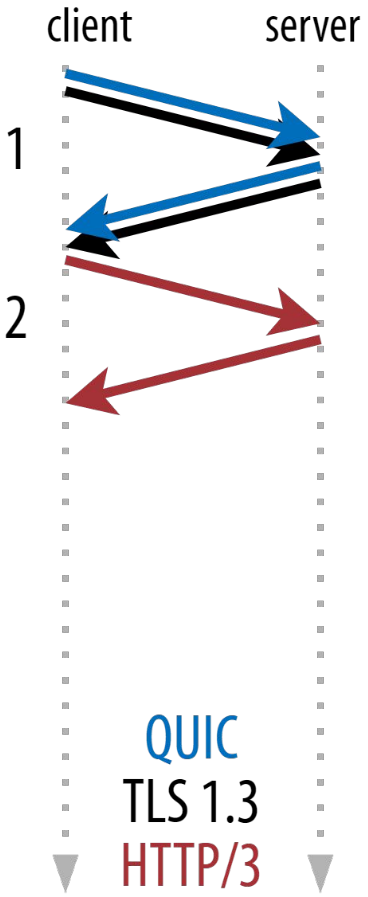
4. Browser *also* tries HTTP/3 from now on, ***in parallel*** with HTTP/1 and 2 ("free" fallback)

Hold on a minute…



100%

VS



~50%
faster

= unfair comparison

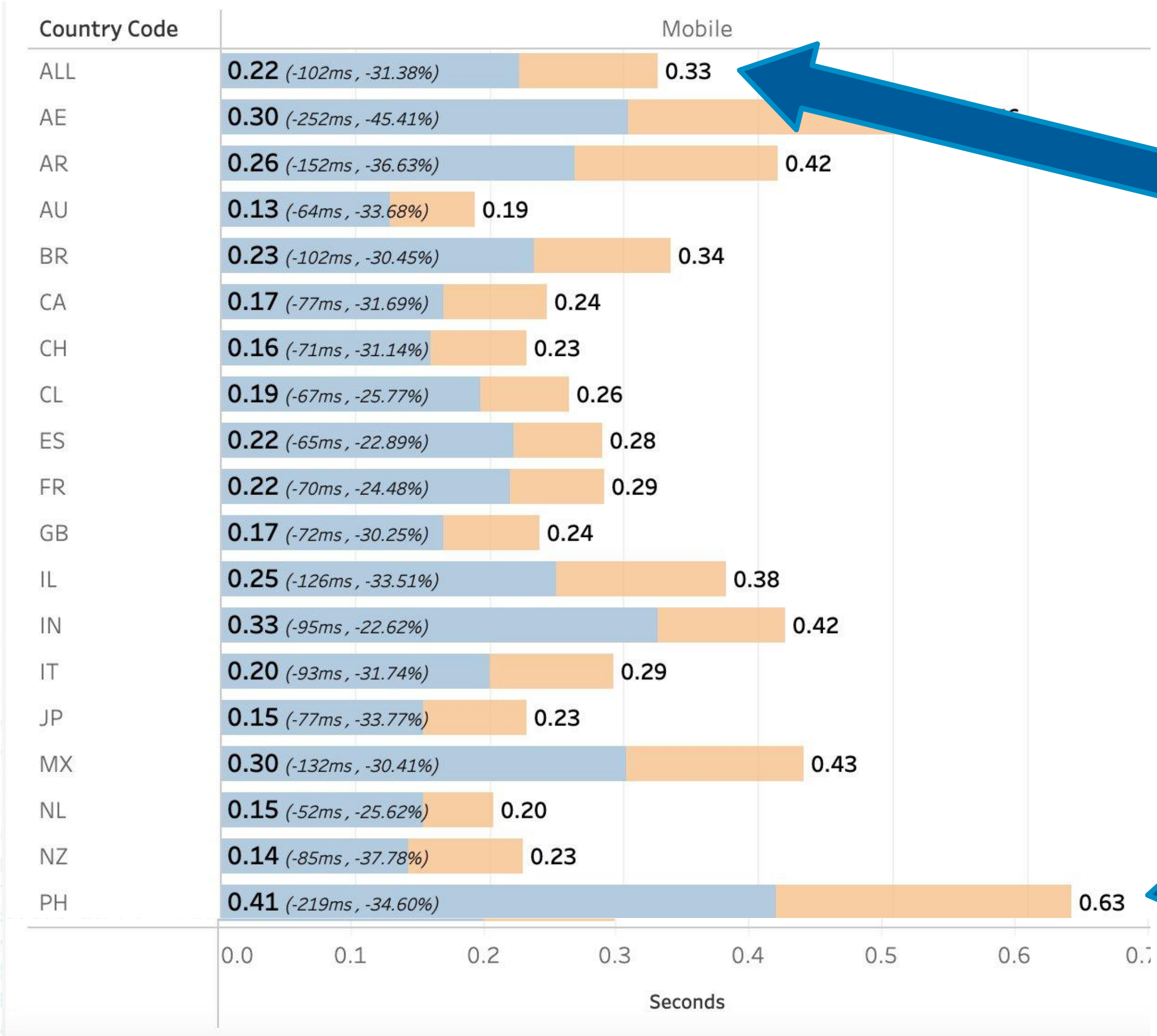
https://www.smashingmagazine.com/2021/08/http3-performance-improvements-part2/

Measured Impact, Fair (2021)

p75 Time to First Byte

HTTP/2

HTTP/3



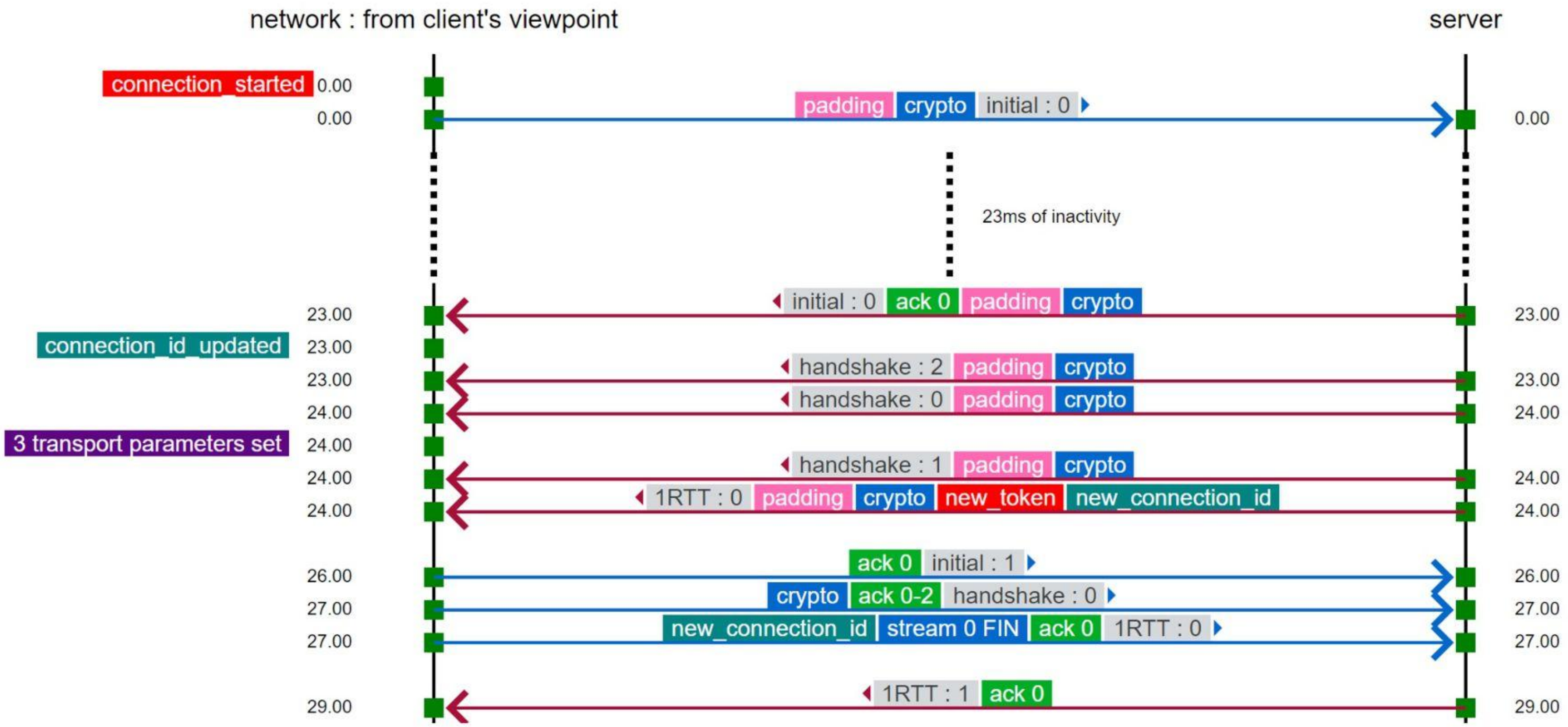
Mean:
31% faster!
330ms to 220ms!

Philippines:
34% faster!
630ms to 410ms!

Akamai Experience the Edge

[qlog] <qvis>

Visual tooling helps me *a lot*



https://qvis.quictools.info

Why can't we just use WIRESHARK ?

- Can't just decrypt QUIC/H3 details: all or nothing

- Don't always have TLS decryption keys
  - Facebook refused to even add SSL_KEYLOGFILE in their stack

- Some features not fully supported
  - HTTP/3 QPACK header decoding was added **just 3 weeks ago**!

- Wireshark JSON/XML output isn't easy to use
  - by default, JSON even contains **duplicate keys**…

- Wire image does not contain all info
  - No **congestion window**, reasons why implementation made decision X

Parsing random application logs is “FUN”

```
I00000036 0xb5080d83e09acbce1e6e4b907633009109 pkt tx pkt 0 dcid=0x108c2996a1d18a8bb1f7611937eb5f30 scid=0xb5080d83e09acbc
I00000036 0xb5080d83e09acbce1e6e4b907633009109 frm tx 0 Short(0x00) STREAM(0x13) id=0x0 fin=1 offset=0 len=16 uni=0
I00000036 0xb5080d83e09acbce1e6e4b907633009109 rcv loss_detection_timer=1541515004932932352 last_hs_tx_pkt_ts=154151500486
I00000090 0xb5080d83e09acbce1e6e4b907633009109 con recv packet len=63
I00000090 0xb5080d83e09acbce1e6e4b907633009109 pkt rx pkt 2 dcid=0xb5080d83e09acbce1e6e4b907633009109 scid=0x108c2996a1d18
I00000090 0xb5080d83e09acbce1e6e4b907633009109 frm rx 2 Handshake(0x7d) ACK(0x1a) largest_ack=0 ack_delay=6(863) ack_block
I00000090 0xb5080d83e09acbce1e6e4b907633009109 frm rx 2 Handshake(0x7d) ACK(0x1a) block=[0..0] block_count=0
I00000090 0xb5080d83e09acbce1e6e4b907633009109 rcv latest_rtt=47 min_rtt=32 smoothed_rtt=34.076 rttvar=15.920 max_ack_dela
I00000090 0xb5080d83e09acbce1e6e4b907633009109 rcv packet 0 acked, slow start cwnd=13370
I00000090 0xb5080d83e09acbce1e6e4b907633009109 pkt read packet 63 left 0
I00000092 0xb5080d83e09acbce1e6e4b907633009109 rcv loss detection timer fired
I00000092 0xb5080d83e09acbce1e6e4b907633009109 rcv handshake_count=0 tlp_count=1 rto_count=0
I00000092 0xb5080d83e09acbce1e6e4b907633009109 con transmit probe pkt left=1
I00000092 0xb5080d83e09acbce1e6e4b907633009109 pkt tx pkt 1 dcid=0x108c2996a1d18a8bb1f7611937eb5f30 scid=0xb5080d83e09acbc
I00000092 0xb5080d83e09acbce1e6e4b907633009109 frm tx 1 Short(0x00) PING(0x07)
I00000092 0xb5080d83e09acbce1e6e4b907633009109 con probe pkt size=35
I00000103 0xb5080d83e09acbce1e6e4b907633009109 con recv packet len=169
I00000103 0xb5080d83e09acbce1e6e4b907633009109 pkt rx pkt 0 dcid=0xb5080d83e09acbce1e6e4b907633009109 scid=0x type=Short(0x00) len=0
I00000103 0xb5080d83e09acbce1e6e4b907633009109 frm rx 0 Short(0x00) CRYPTO(0x18) offset=0 len=130
Ordered CRYPTO data
00000000  04 00 00 3d 00 00 1c 20  db 3d 0e 65 08 00 00 00  |...=... .=.e....|
00000010  00 00 00 00 00 00 20 da  41 9b 6d 9d d0 6b 98 4f  |...... .A.m..k.O|
00000020  bc bc 57 57 7a eb 74 3e  a2 11 ea fd e4 cd 1b d5  |..WWz.t>........|
00000030  5b 1b 75 f3 51 1a 09 00  08 00 2a 00 04 ff ff ff  |[.u.Q.....*.....|
00000040  ff 04 00 00 3d 00 00 1c  20 06 2e 42 d3 08 00 00  |....=... ..B....|
00000050  00 00 00 00 00 01 00 20  25 05 93 85 08 6b e5 0f  |....... %....k..|
00000060  43 63 a9 b7 5b c4 e9 d4  9b 63 9d 27 1f 16 67 68  |Cc..[....c.'..gh|
00000070  78 a0 42 3f cb b2 77 f8  00 08 00 2a 00 04 ff ff  |x.B?..w....*....|
00000080  ff ff                                             |..|
00000082
```

```
9 con recv
9 pkt rx pk
9 frm rx 2
9 frm rx 2
9 rcv lates
9 rcv packe
```

qlog examples

```
{
"time": 15000,
"name": "transport:packet_received",
"data": {
    "header": {
        "packet_type": "1rtt",
        "packet_number": 25
    },
    "frames": [
    {
        "frame_type": "ack",
        "acked_ranges": [
            [10,15],
            [17,20]
        ]
     }]
}}
```

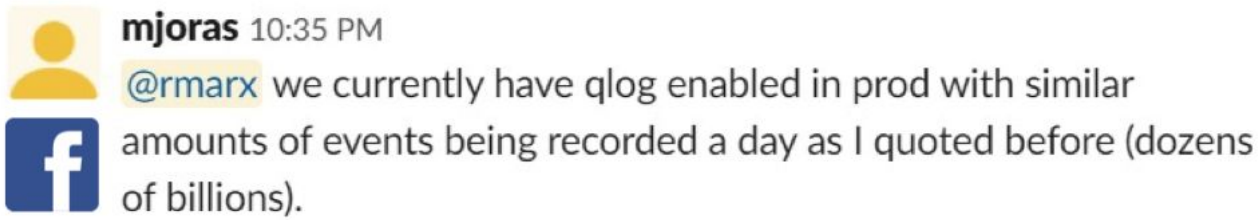
```
{
"time": 15001,
"name": "recovery:metrics_updated",
"data": {
  "min_rtt": 25,
  "smoothed_rtt": 30,
  "latest_rtt": 25,

  "congestion_window": 60,
  "bytes_in_flight": 77000,
}
```

[qlog] adoption

>70% of QUIC implementations have (partial) support:

- aioquic
- quic-go
- quiche
- mvfst
- picoquic
- haskell
- ngtcp2
- ...



Others do something similar:

- msquic
- google quiche

Standardization in-progress @
https://github.com/quicwg/qlog

https://interop.seemann.io
https://qvis.edm.uhasselt.be
https://qlog.edm.uhasselt.be/anrw
https://blog.cloudflare.com/cubic-and-hystart-support-in-quiche
https://huitema.wordpress.com/2020/07/12/parsing-quic-logs-and-assessing-packet-losses

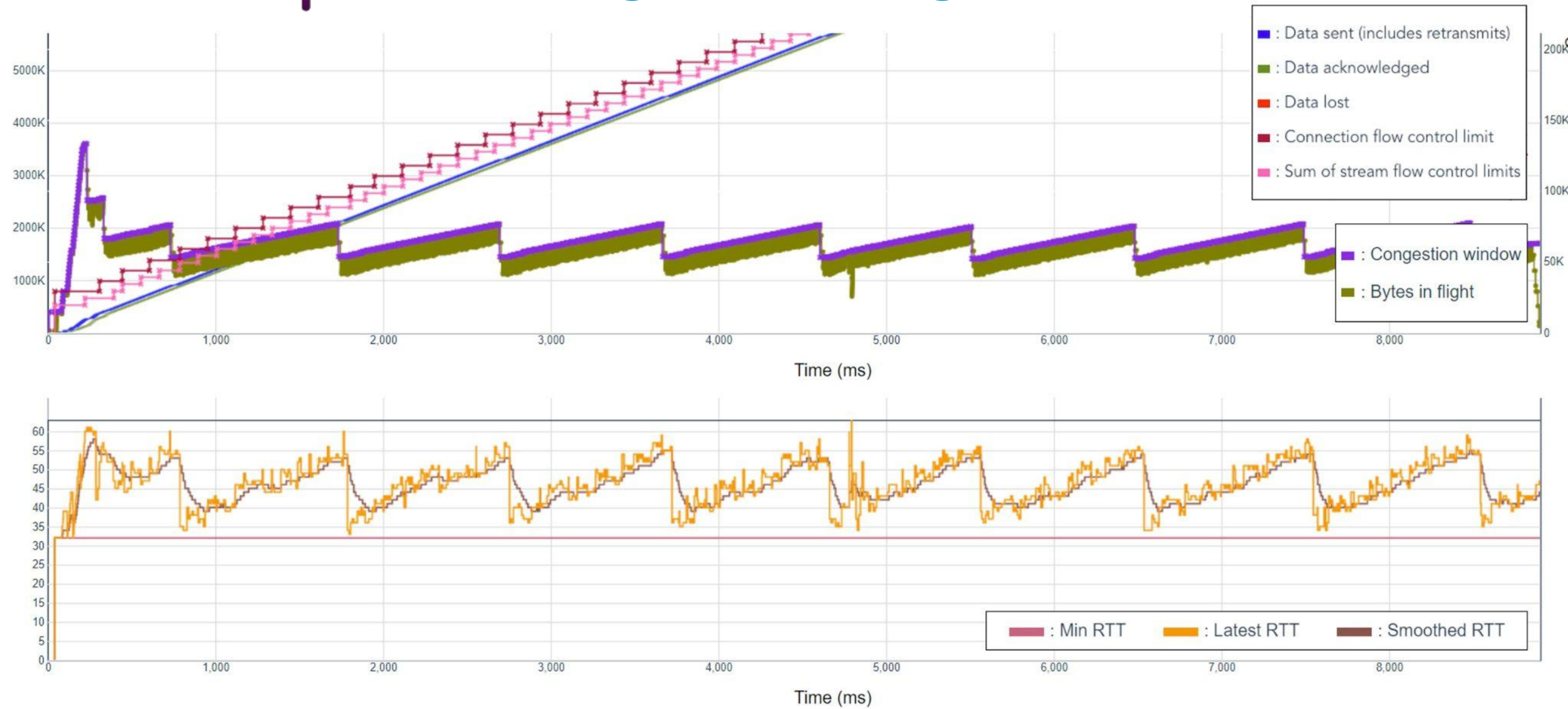
qvis Sequence diagram

Too many Selective Acknowledgements?

https://qvis.quictools.info

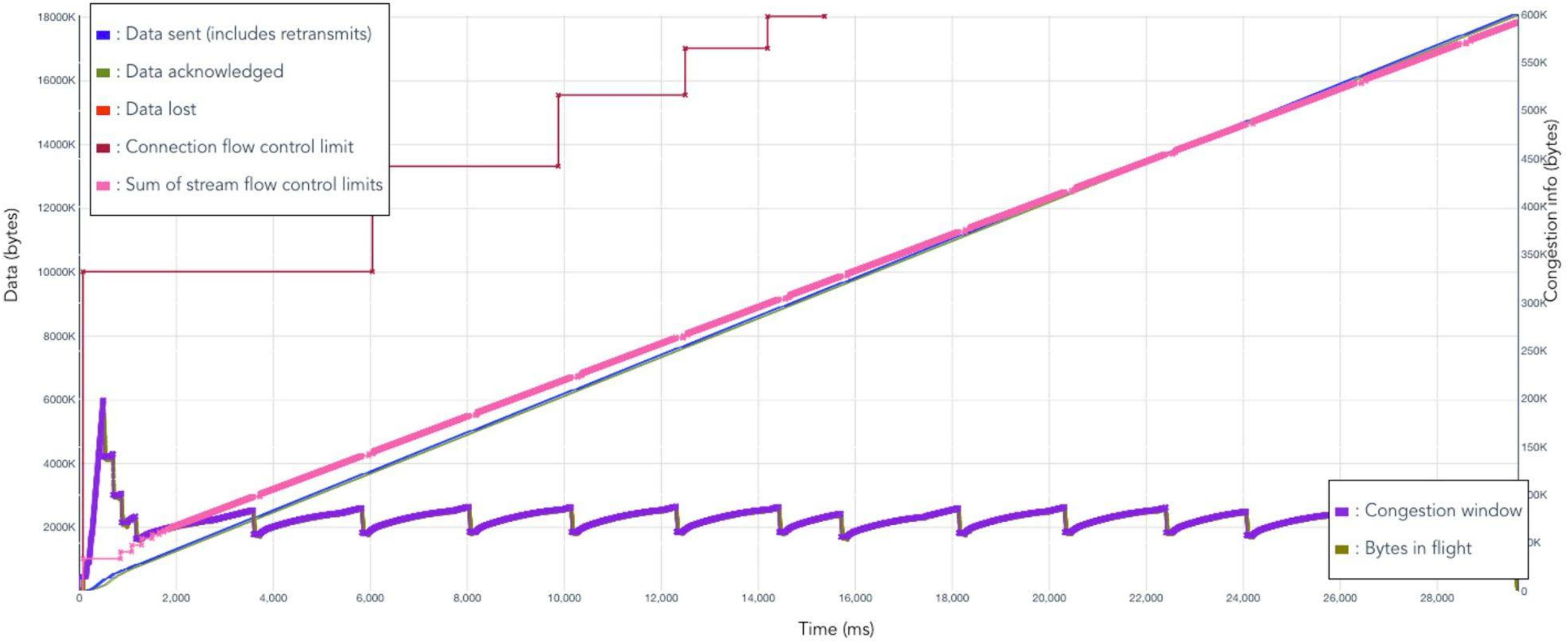
Akamai Experience the Edge

<qvis> Congestion diagram



https://qvis.quictools.info

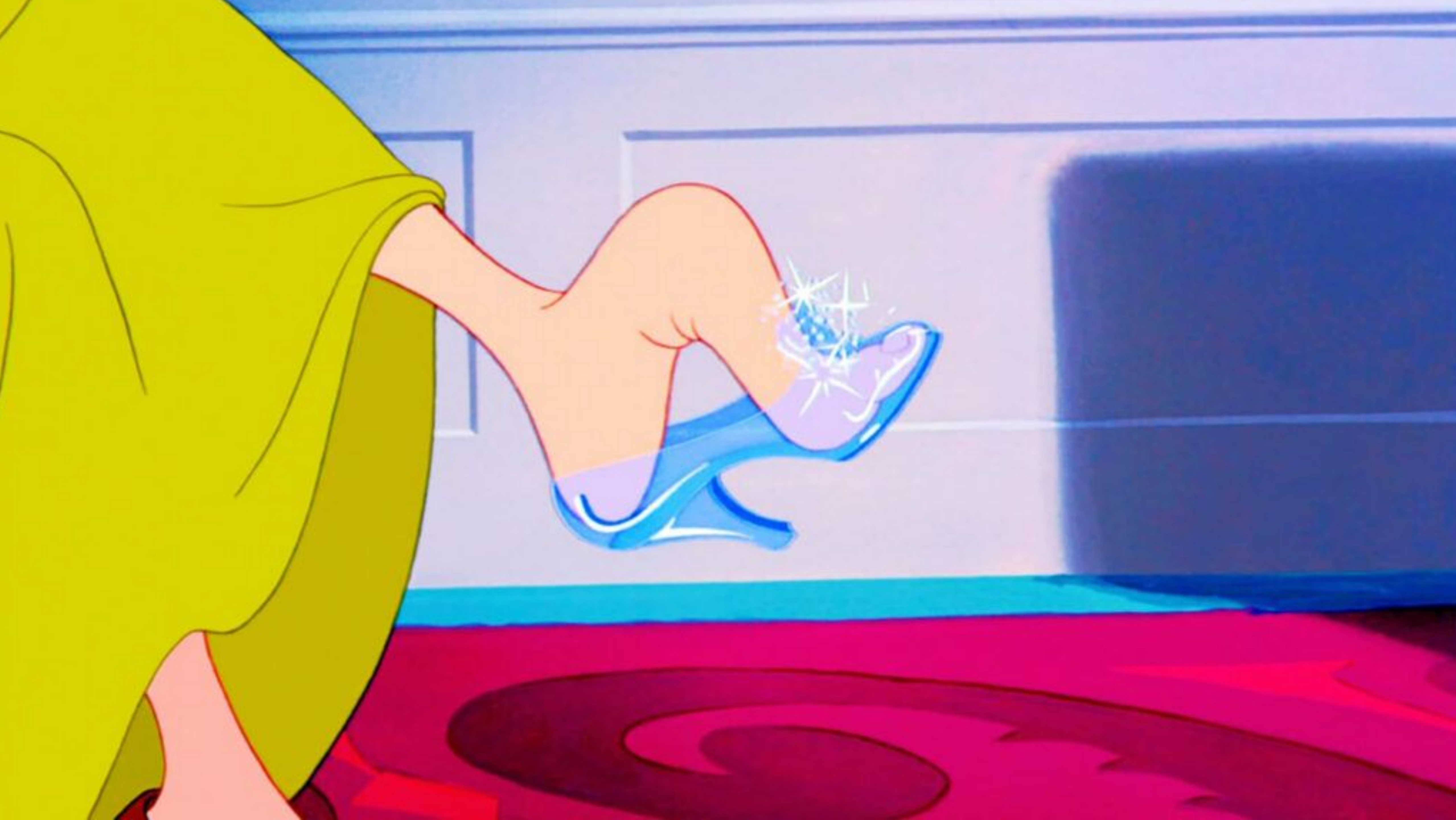
<qvis> Congestion diagram



https://qvis.quictools.info

Akamai Experience the Edge

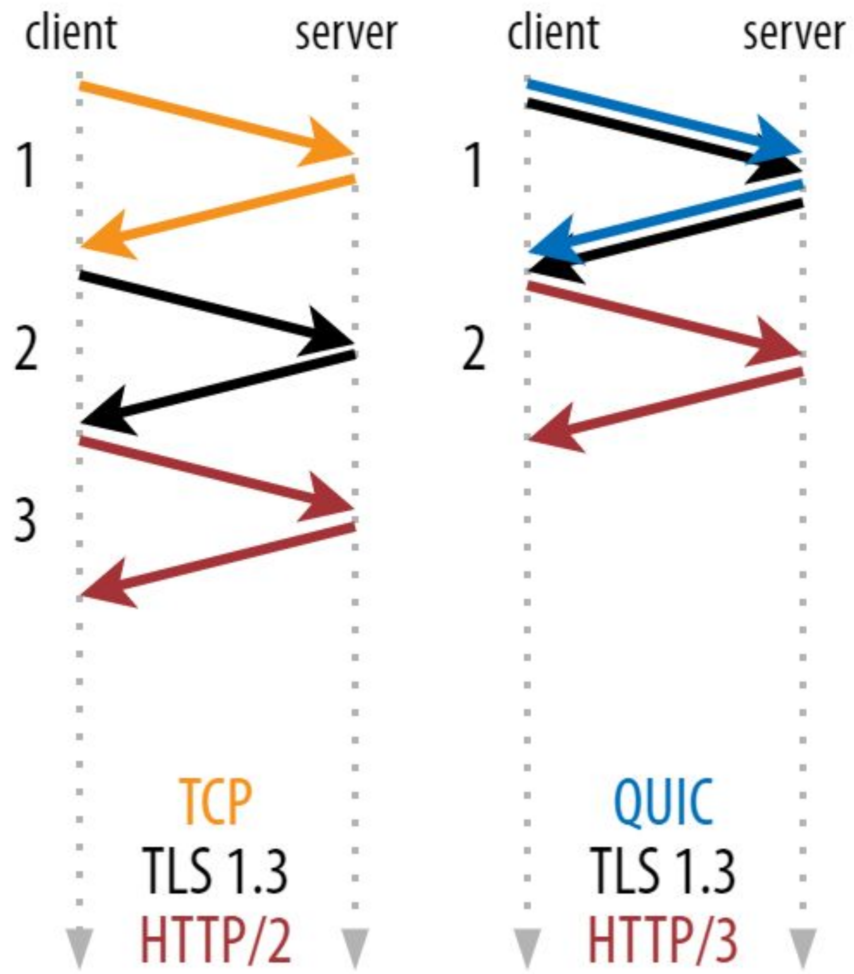

[qlog]
<qvis>



HTTP/3 not faster than HTTP/2 at all,

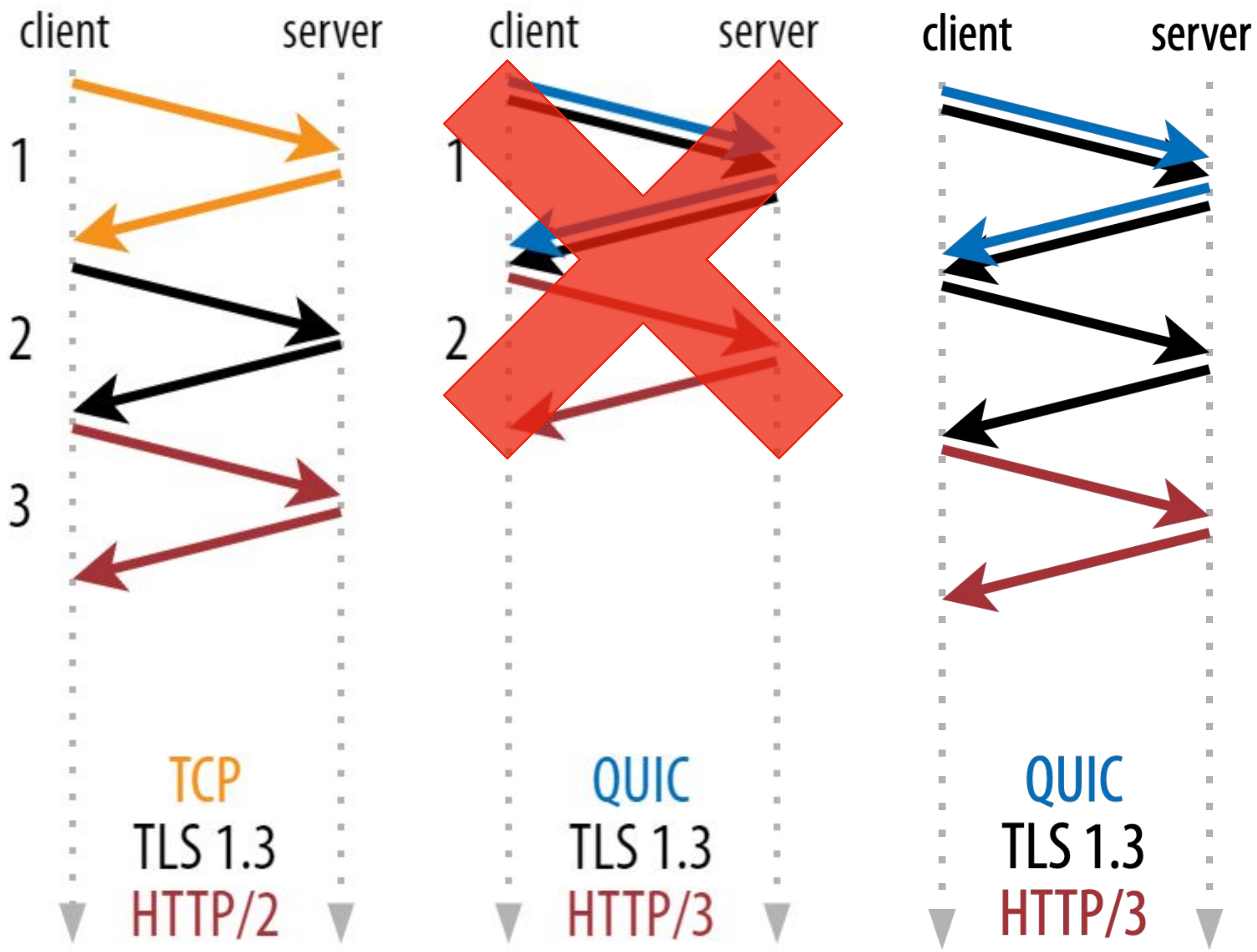
we want our money back!

Faster Handshake *Theory*



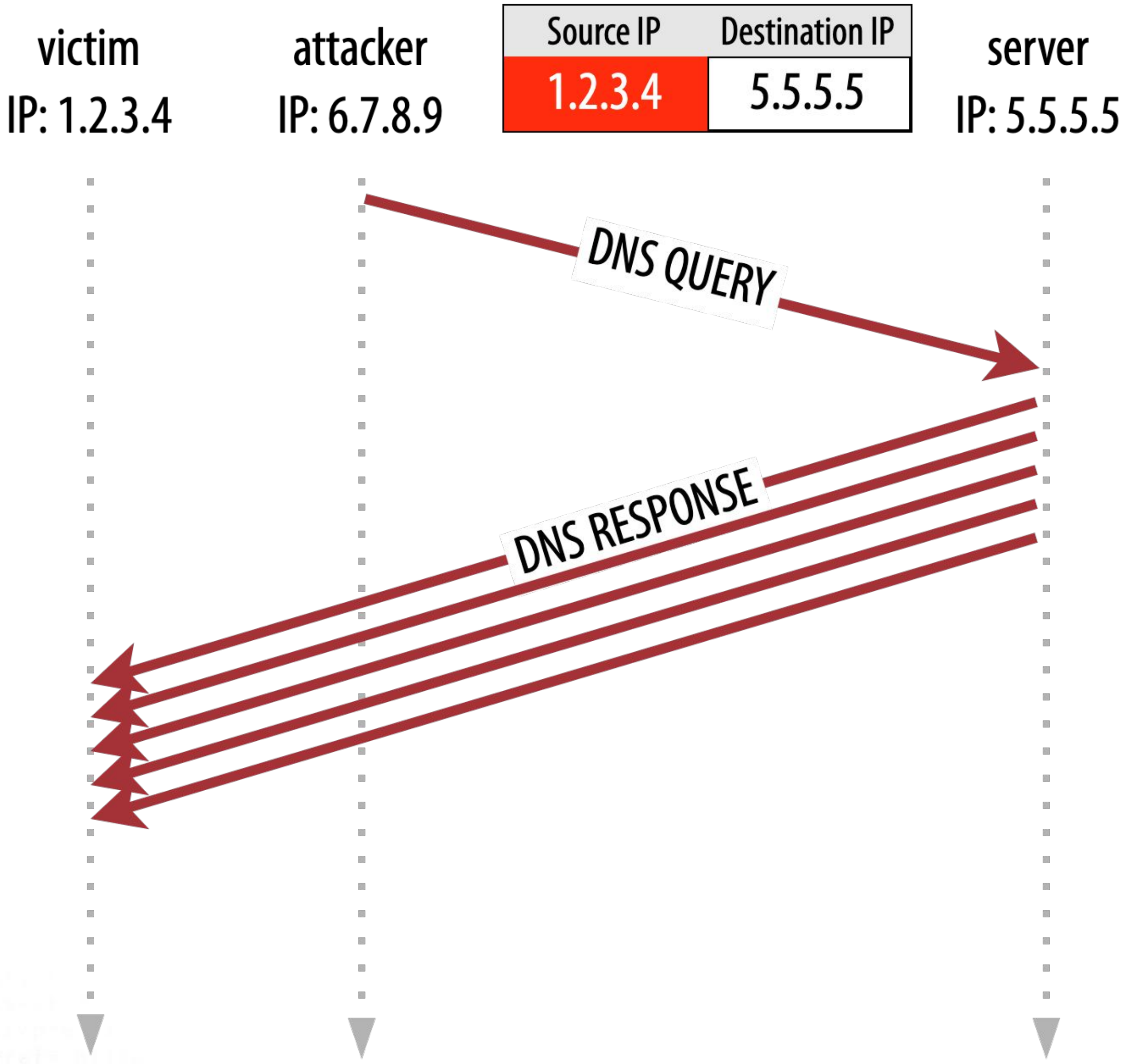
https://understanding-quic.net/
https://blog.apnic.net/2023/01/16/on-the-interplay-between-tls-certificates-and-quic-performance/

Faster Handshake in *Practice*…



https://understanding-quic.net/
https://blog.apnic.net/2023/01/16/on-the-interplay-between-tls-certificates-and-quic-performance/

UDP reflection / amplification attack

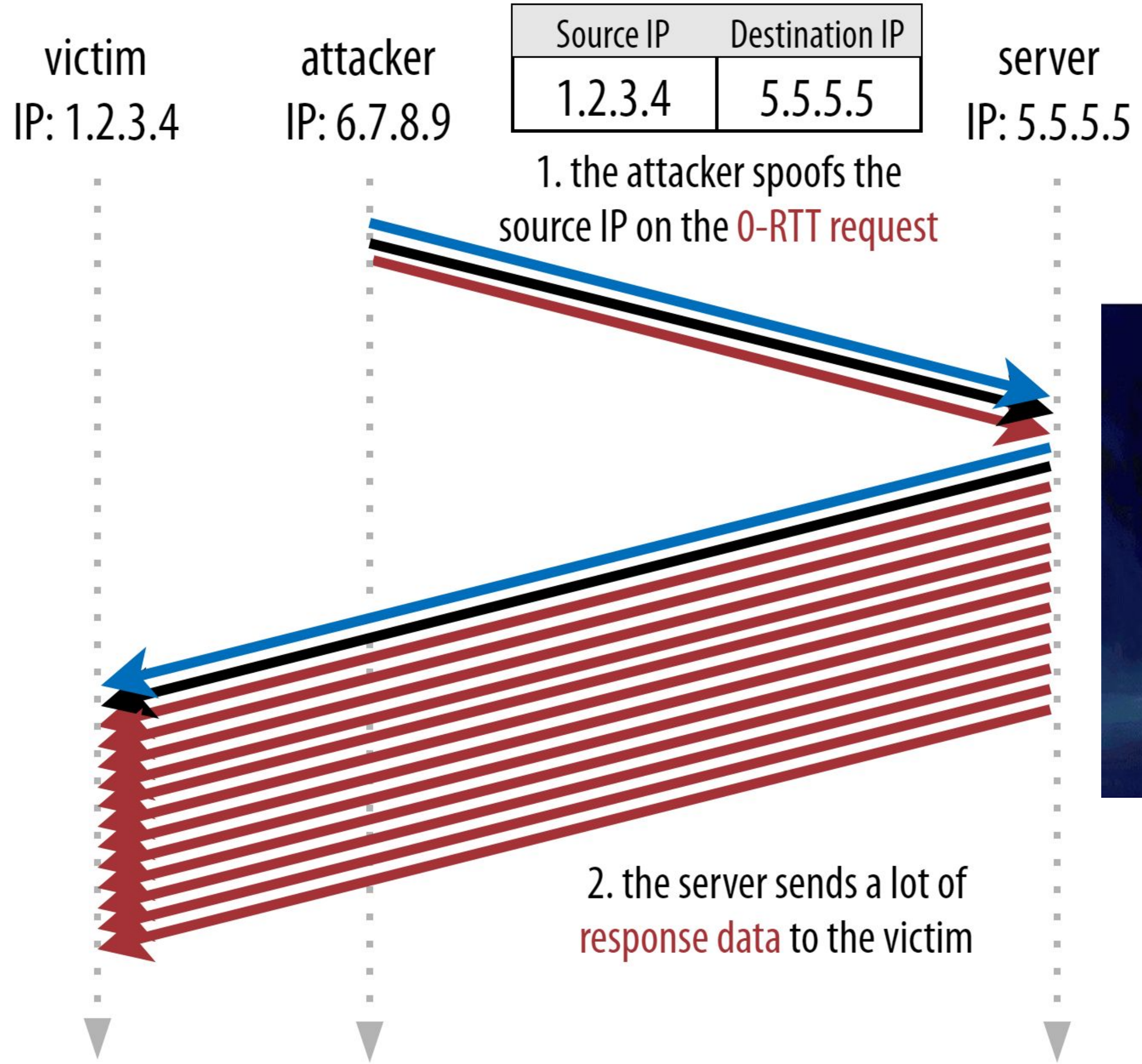


5x bandwidth amplification towards victim

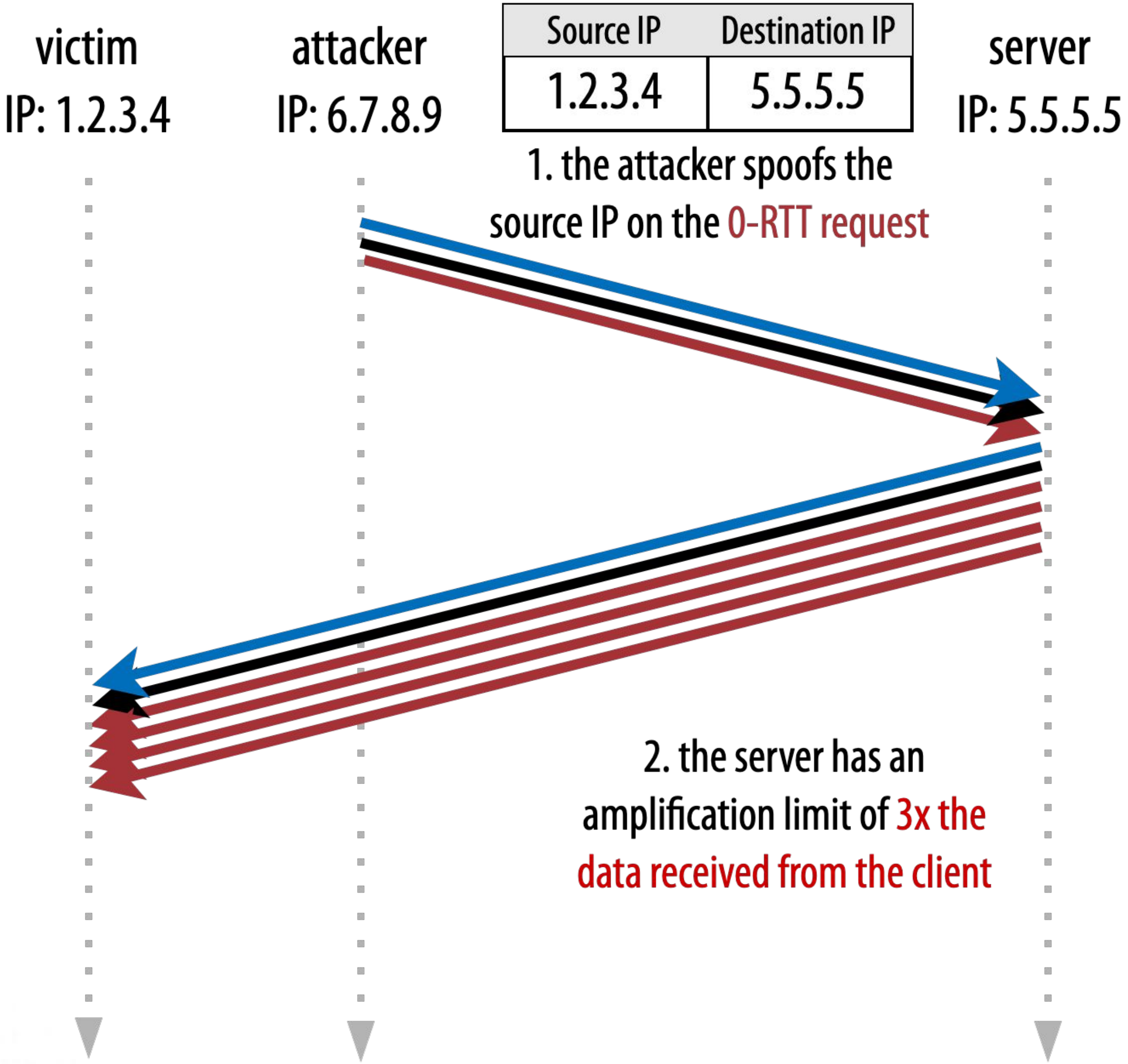
Memcrashed:
51000x amplification

https://blog.cloudflare.com/memcrashed-major-amplification-attacks-from-port-11211/

QUIC vulnerable *during* handshake: 0-RTT



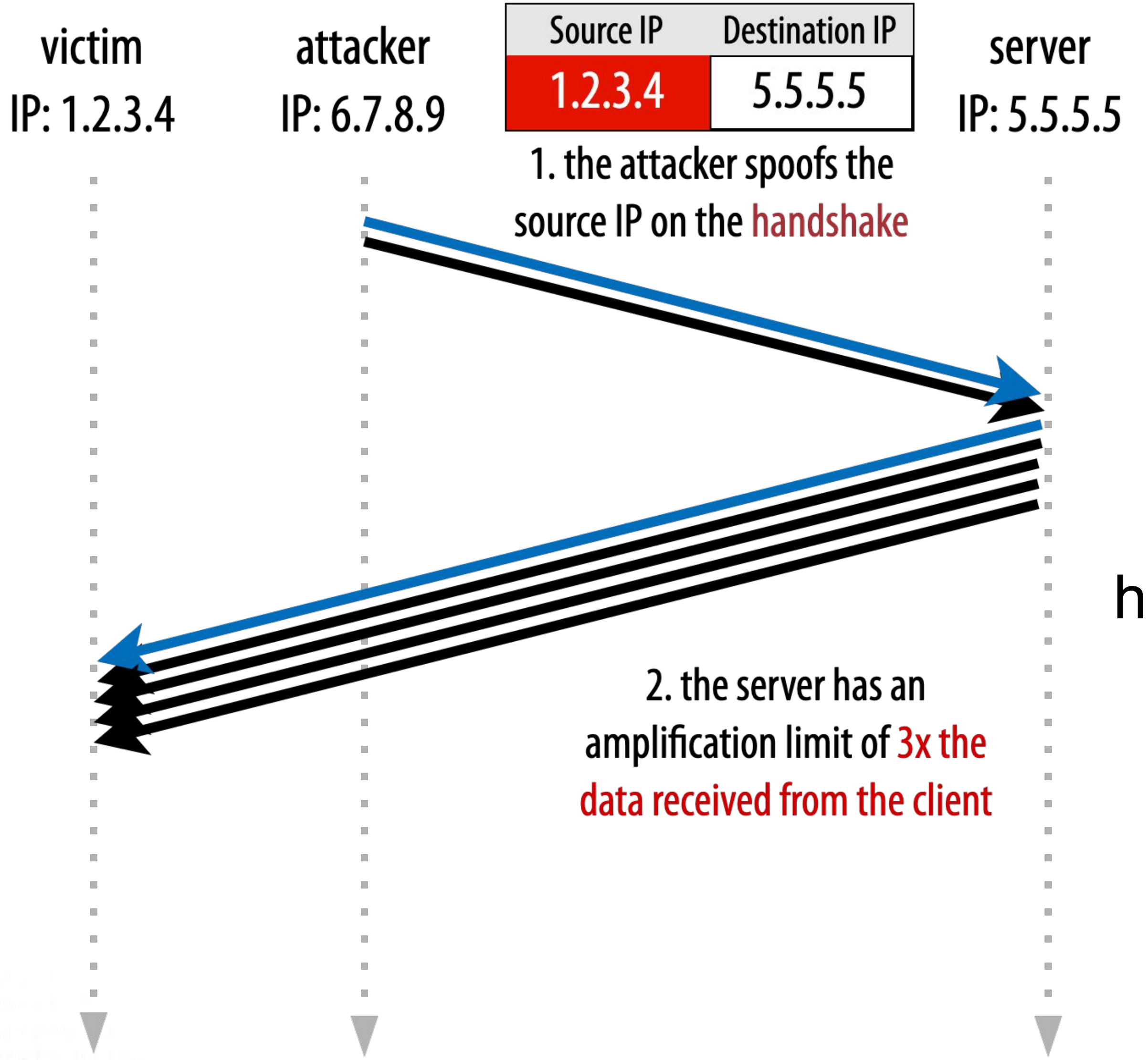
Amplification prevention in QUIC



3x

amplification limit

Not just 0-RTT, also “normal” handshake

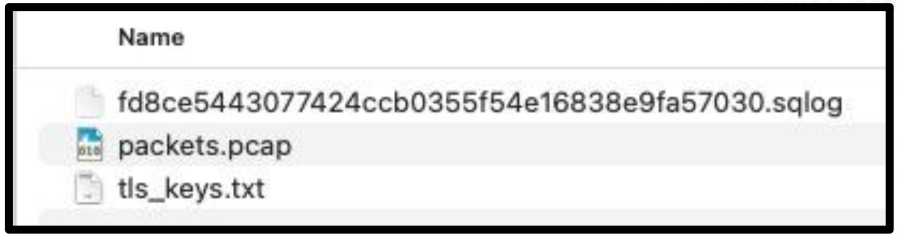


TLS certificate can push handshake size > 3x limit

curl-http3 docker image

Supports both pcaps (with TLS keys) and qlog output!

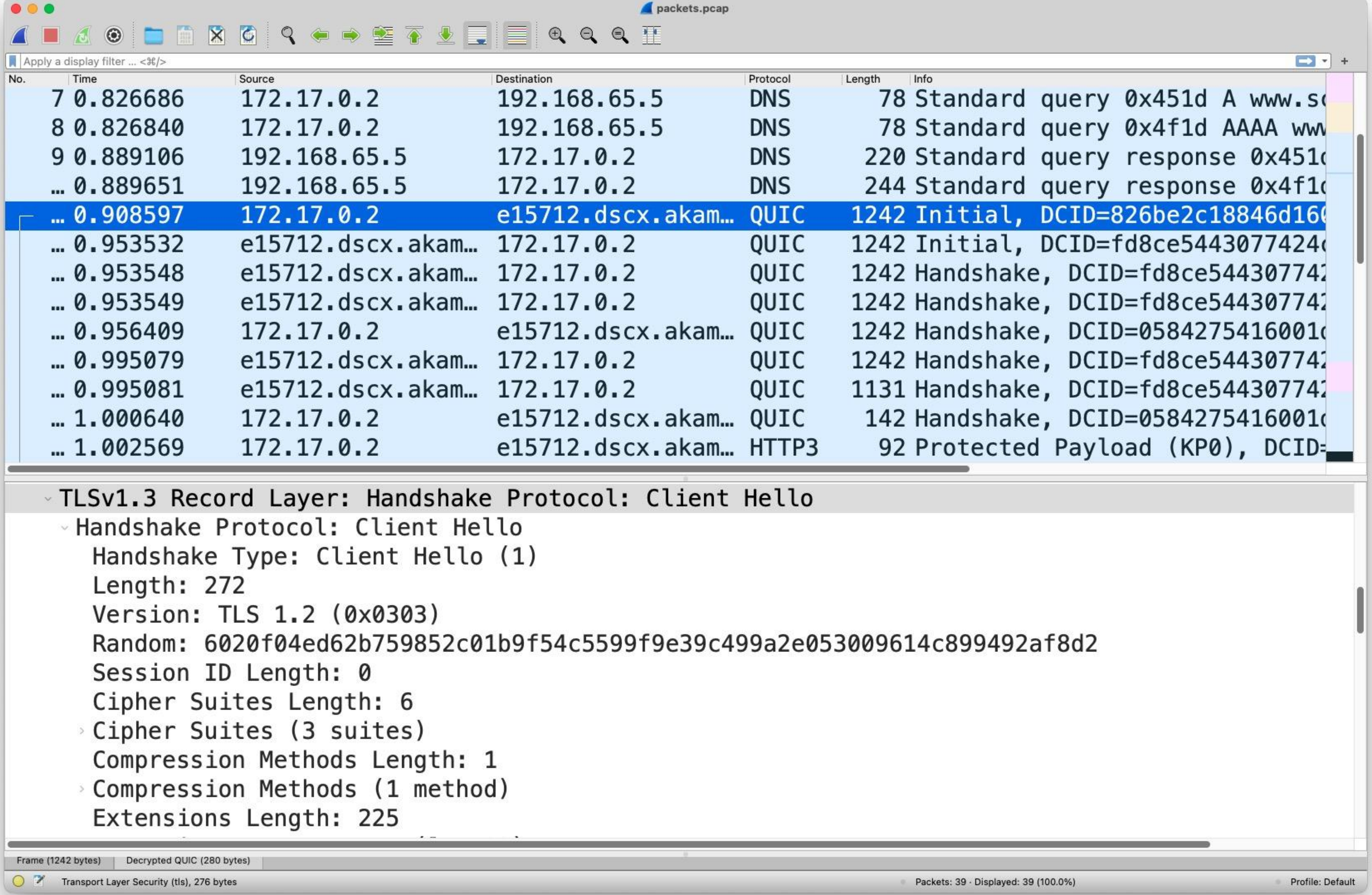
```
docker run -it --rm
    --volume $(pwd)/pcaps_on_host:/srv
    --env QLOGDIR=/srv
    --env SSLKEYLOGFILE=/srv/tls_keys.txt
```



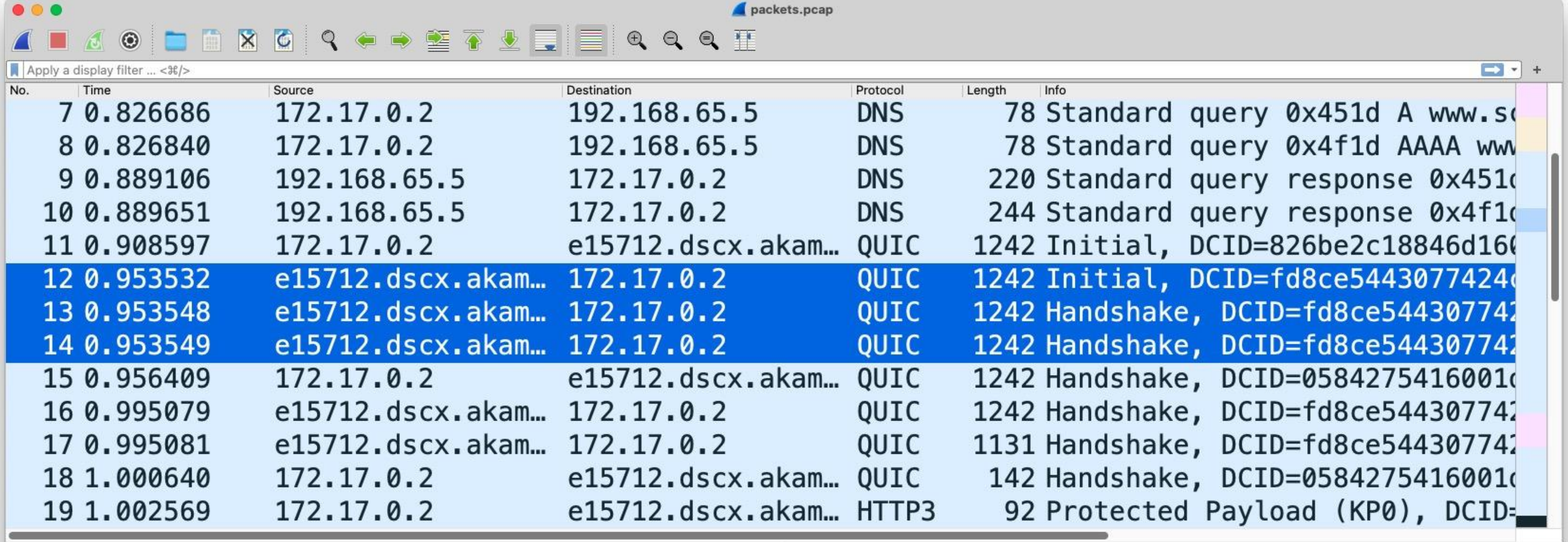

```
rmarx/curl-http3

    bash -c "tcpdump -w /srv/packets.pcap -i eth0 & sleep 1;
    curl -IL https://www.sre.com --http3;
    sleep 2; pkill tcpdump; sleep 2"
```

https://github.com/rmarx/curl-http3


packets.pcap
Apply a display filter … <⌘/>
No. Time Source Destination Protocol Length Info
7 0.826686 172.17.0.2 192.168.65.5 DNS 78 Standard query 0x451d A www.s
8 0.826840 172.17.0.2 192.168.65.5 DNS 78 Standard query 0x4f1d AAAA www
9 0.889106 192.168.65.5 172.17.0.2 DNS 220 Standard query response 0x451
… 0.889651 192.168.65.5 172.17.0.2 DNS 244 Standard query response 0x4f1
… 0.908597 172.17.0.2 e15712.dscx.akam… QUIC 1242 Initial, DCID=826be2c18846d16
… 0.953532 e15712.dscx.akam… 172.17.0.2 QUIC 1242 Initial, DCID=fd8ce5443077424
… 0.953548 e15712.dscx.akam… 172.17.0.2 QUIC 1242 Handshake, DCID=fd8ce54430774
… 0.953549 e15712.dscx.akam… 172.17.0.2 QUIC 1242 Handshake, DCID=fd8ce54430774
… 0.956409 172.17.0.2 e15712.dscx.akam… QUIC 1242 Handshake, DCID=0584275416001
… 0.995079 e15712.dscx.akam… 172.17.0.2 QUIC 1242 Handshake, DCID=fd8ce54430774
… 0.995081 e15712.dscx.akam… 172.17.0.2 QUIC 1131 Handshake, DCID=fd8ce54430774
… 1.000640 172.17.0.2 e15712.dscx.akam… QUIC 142 Handshake, DCID=0584275416001
… 1.002569 172.17.0.2 e15712.dscx.akam… HTTP3 92 Protected Payload (KP0), DCID=
TLSv1.3 Record Layer: Handshake Protocol: Client Hello
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 272
Version: TLS 1.2 (0x0303)
Random: 6020f04ed62b759852c01b9f54c5599f9e39c499a2e053009614c899492af8d2
Session ID Length: 0
Cipher Suites Length: 6
Cipher Suites (3 suites)
Compression Methods Length: 1
Compression Methods (1 method)
Extensions Length: 225
Frame (1242 bytes)
Decrypted QUIC (280 bytes)
Transport Layer Security (tls), 276 bytes
Packets: 39 · Displayed: 39 (100.0%)
Profile: Default

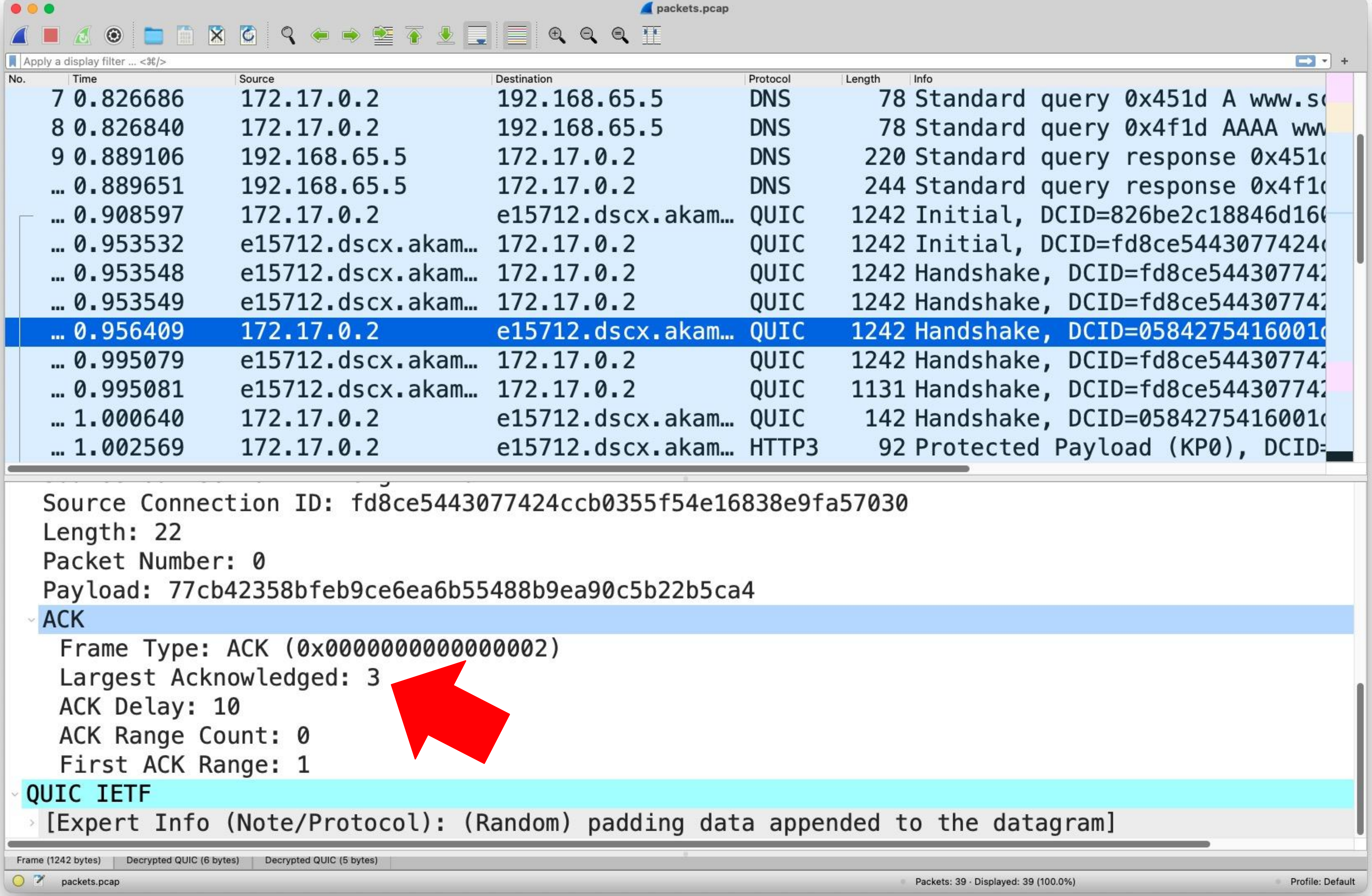
e Edge



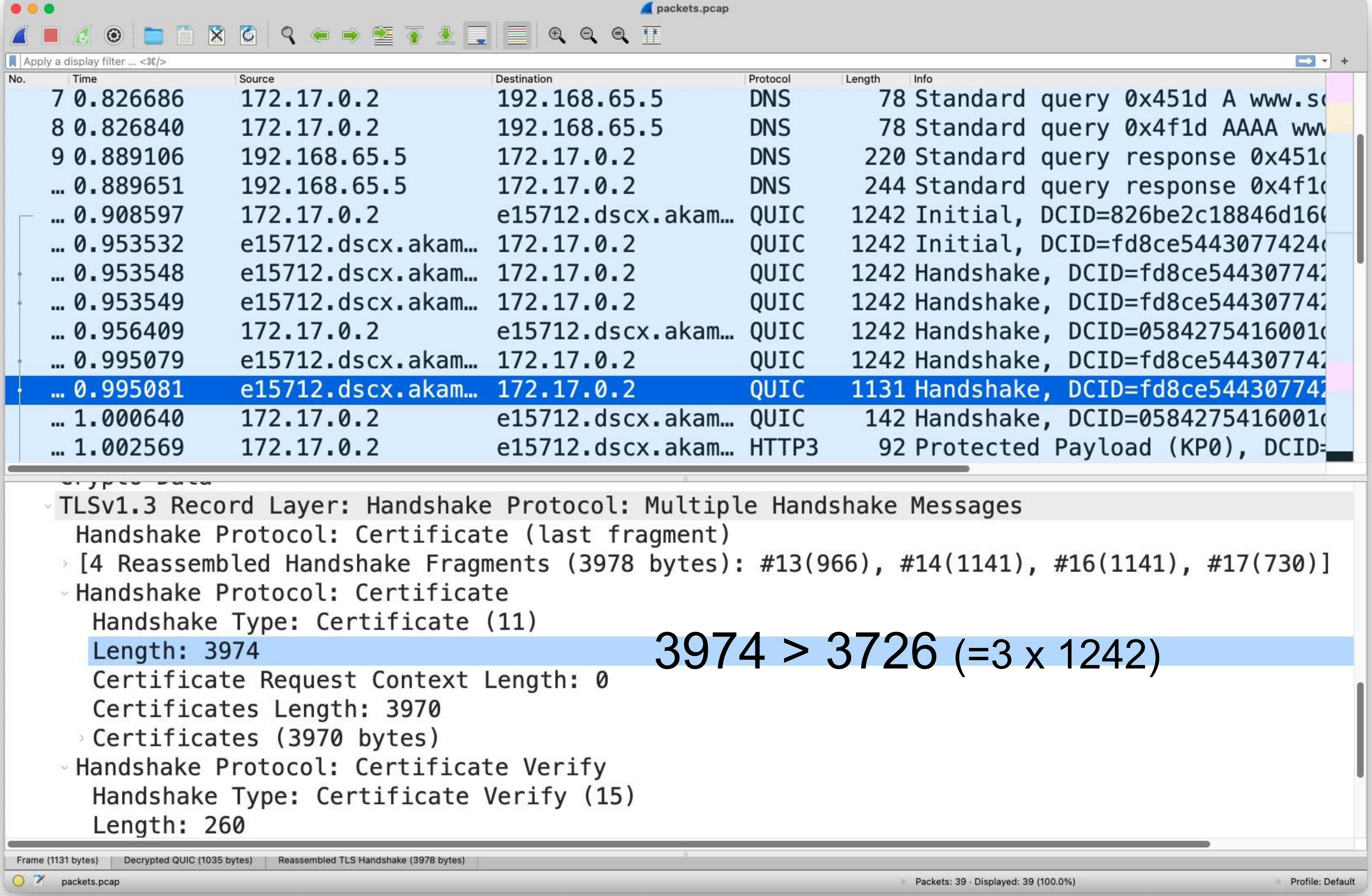
Client sent 1x 1242 bytes

Server is limited to 3x 1242 bytes

packets.pcap Packets: 39 · Displayed: 39 (100.0%) · Selected: 3 (7.7%) Profile: Default


packets.pcap
Apply a display filter … <⌘/>
No. Time Source Destination Protocol Length Info
7 0.826686 172.17.0.2 192.168.65.5 DNS 78 Standard query 0x451d A www.s
8 0.826840 172.17.0.2 192.168.65.5 DNS 78 Standard query 0x4f1d AAAA ww
9 0.889106 192.168.65.5 172.17.0.2 DNS 220 Standard query response 0x451
… 0.889651 192.168.65.5 172.17.0.2 DNS 244 Standard query response 0x4f1
… 0.908597 172.17.0.2 e15712.dscx.akam… QUIC 1242 Initial, DCID=826be2c18846d16
… 0.953532 e15712.dscx.akam… 172.17.0.2 QUIC 1242 Initial, DCID=fd8ce5443077424
… 0.953548 e15712.dscx.akam… 172.17.0.2 QUIC 1242 Handshake, DCID=fd8ce54430774
… 0.953549 e15712.dscx.akam… 172.17.0.2 QUIC 1242 Handshake, DCID=fd8ce54430774
… 0.956409 172.17.0.2 e15712.dscx.akam… QUIC 1242 Handshake, DCID=058427541600
… 0.995079 e15712.dscx.akam… 172.17.0.2 QUIC 1242 Handshake, DCID=fd8ce54430774
… 0.995081 e15712.dscx.akam… 172.17.0.2 QUIC 1131 Handshake, DCID=fd8ce54430774
… 1.000640 172.17.0.2 e15712.dscx.akam… QUIC 142 Handshake, DCID=058427541600
… 1.002569 172.17.0.2 e15712.dscx.akam… HTTP3 92 Protected Payload (KP0), DCID=
Source Connection ID: fd8ce5443077424ccb0355f54e16838e9fa57030
Length: 22
Packet Number: 0
Payload: 77cb42358bfeb9ce6ea6b55488b9ea90c5b22b5ca4
ACK
Frame Type: ACK (0x0000000000000002)
Largest Acknowledged: 3
ACK Delay: 10
ACK Range Count: 0
First ACK Range: 1
QUIC IETF
[Expert Info (Note/Protocol): (Random) padding data appended to the datagram]
Frame (1242 bytes) Decrypted QUIC (6 bytes) Decrypted QUIC (5 bytes)
packets.pcap
Packets: 39 · Displayed: 39 (100.0%)
Profile: Default

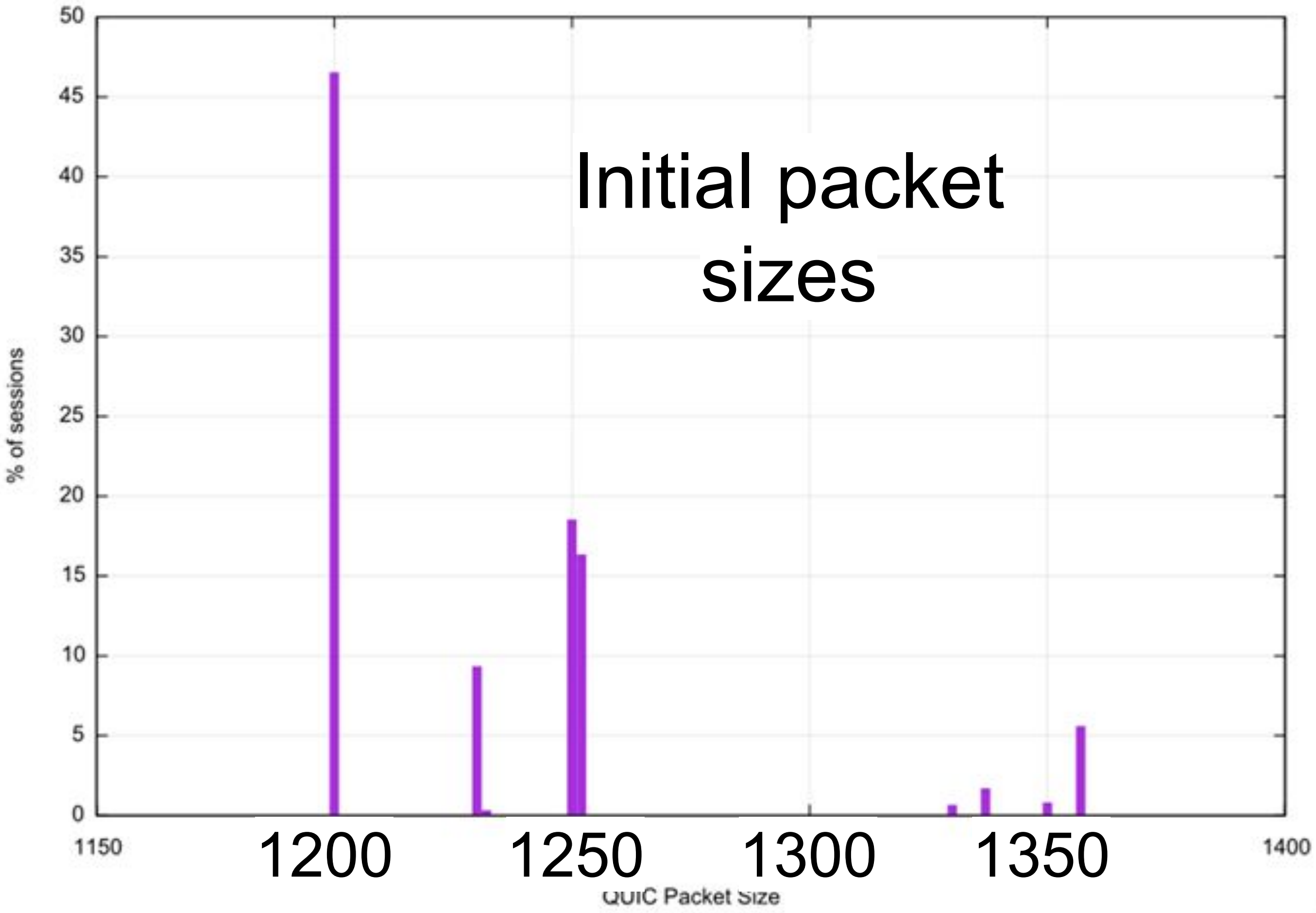
e Edge



3974 > 3726 (=3 x 1242)

e Edge

Lots of magic numbers here



Many deployments ignore 3X and go to **4, 5 or 6X** just to get handshake done in 1 RTT

https://2023.apricot.net/assets/files/APPS314/2023-03-01-quic-apri_1677636425.pdf
https://blog.apnic.net/2020/05/28/does-the-quic-handshake-require-compression-to-be-fast/
https://blog.apnic.net/2023/01/16/on-the-interplay-between-tls-certificates-and-quic-performance/


I THOUGHT THINGS WERE BAD
UNDER MUFASA.

Deploy your own HTTP/3

Reminder:
- Not all code is usually open source
- Open source configuration != actual deployment

(Almost) full support:

Caddy H2O Hypercorn

Most mature, most complicated:

mvfst ts QUICHE MsQuic

In beta/testing:

NGINX HAPROXY

Months/years out:

APACHE HTTP SERVER PROJECT node JS

https://github.com/quicwg/base-drafts/wiki/Implementations

Deploy your own HTTP/3

Reminder:

ce

ual deployment

(Almost) f ... ed:

Caddy

MsQuic

In beta/te

OpenSSL *refuses* to provide sensible APIs for QUIC integration…

Use QuicTLS, BoringSSL, LibreSSL, or WolfSSL

https://github.com/haproxy/wiki/wiki/SSL-Libraries-Support-Status

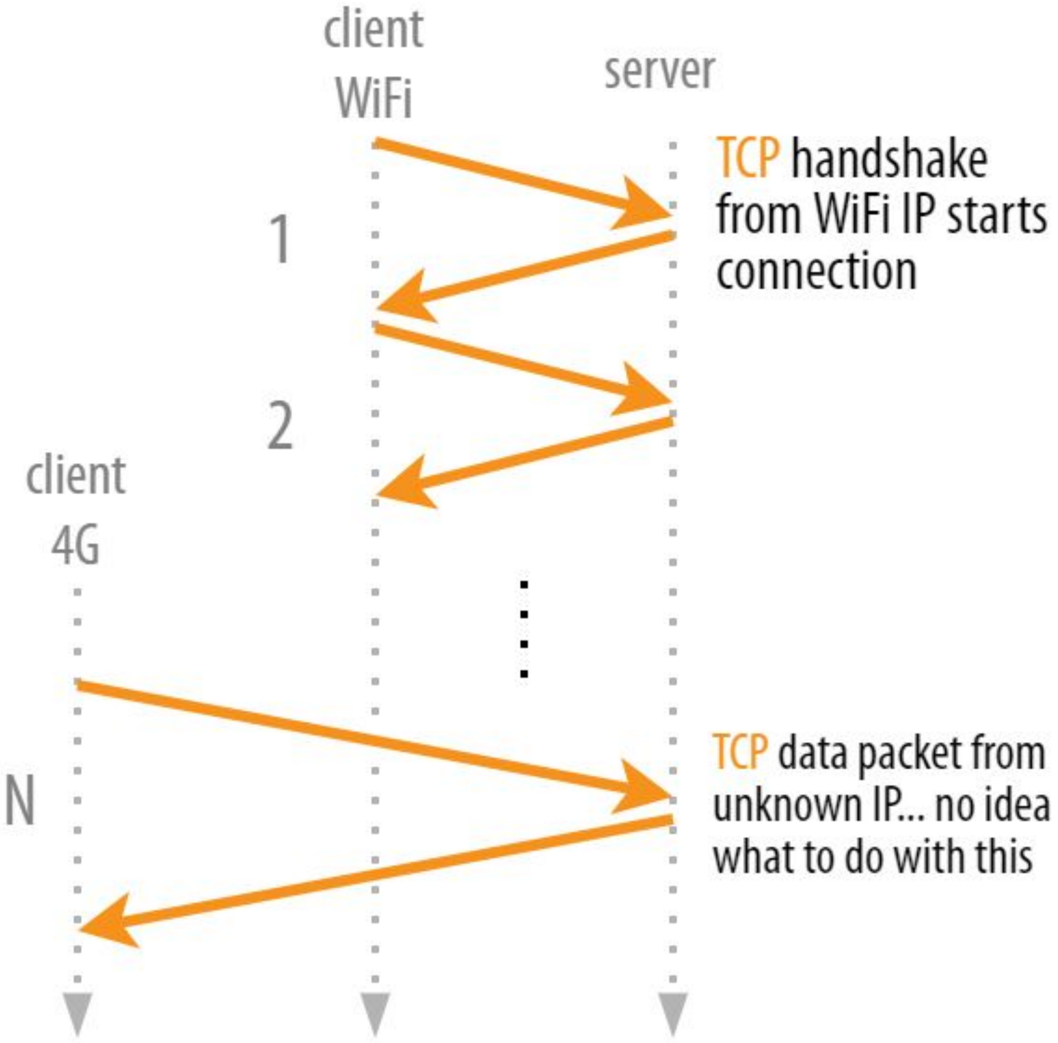
https://github.com/quicwg/base-drafts/wiki/Implementations

But wait, there’s more…

Content servers are 1 thing, what about:

- Load balancers?
- Terminating/reverse proxies?
- Firewalls?
- Observability?
- TLS key/certificate management?
- DNS????
- …

TCP “Connection”



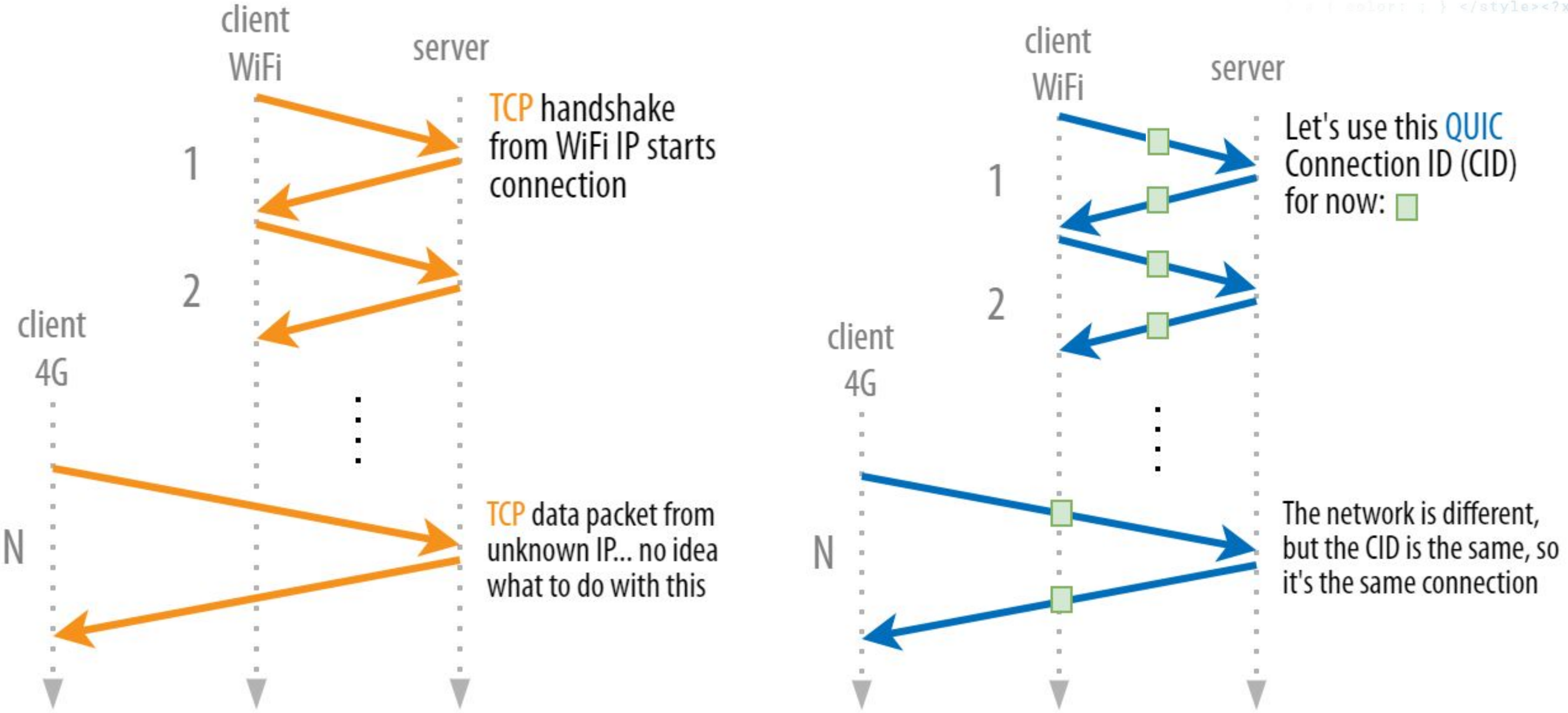
All client TCP connections need to be re-established after 4-tuple changes

- Active migration (wifi to 4G)
- NAT rebinding after timeout

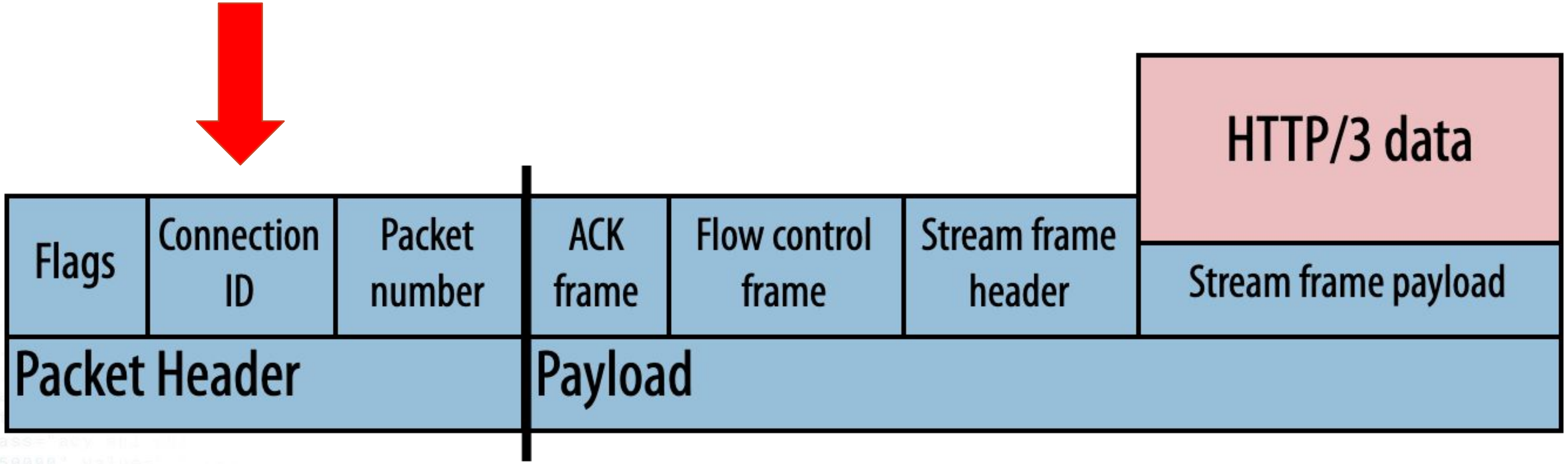
⇒ Connection handshake delay
⇒ Loss of HTTP/TLS context

Gets load balanced to different server/cluster?

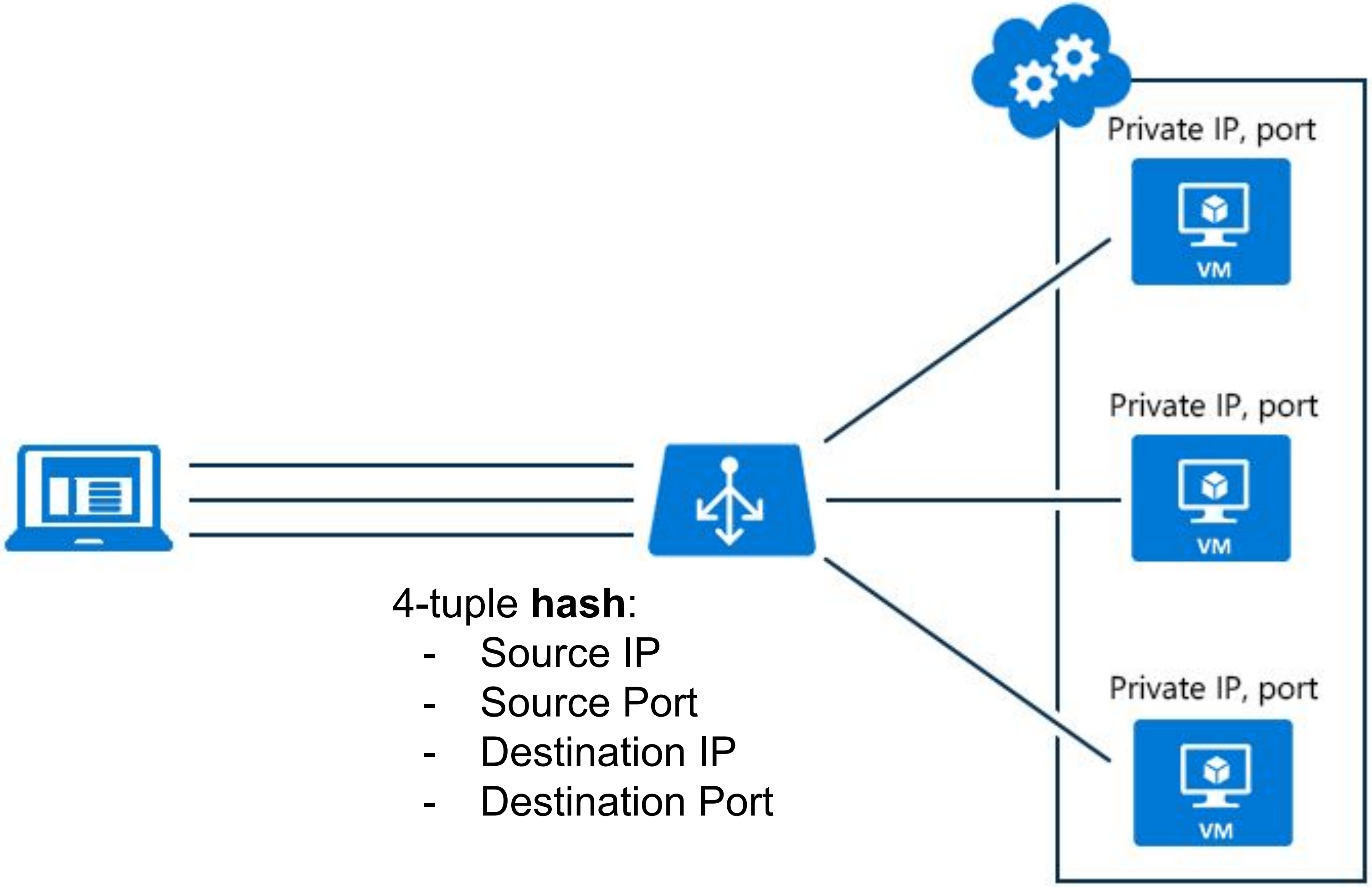
QUIC's "Connection ID"



CID included in each QUIC packet!

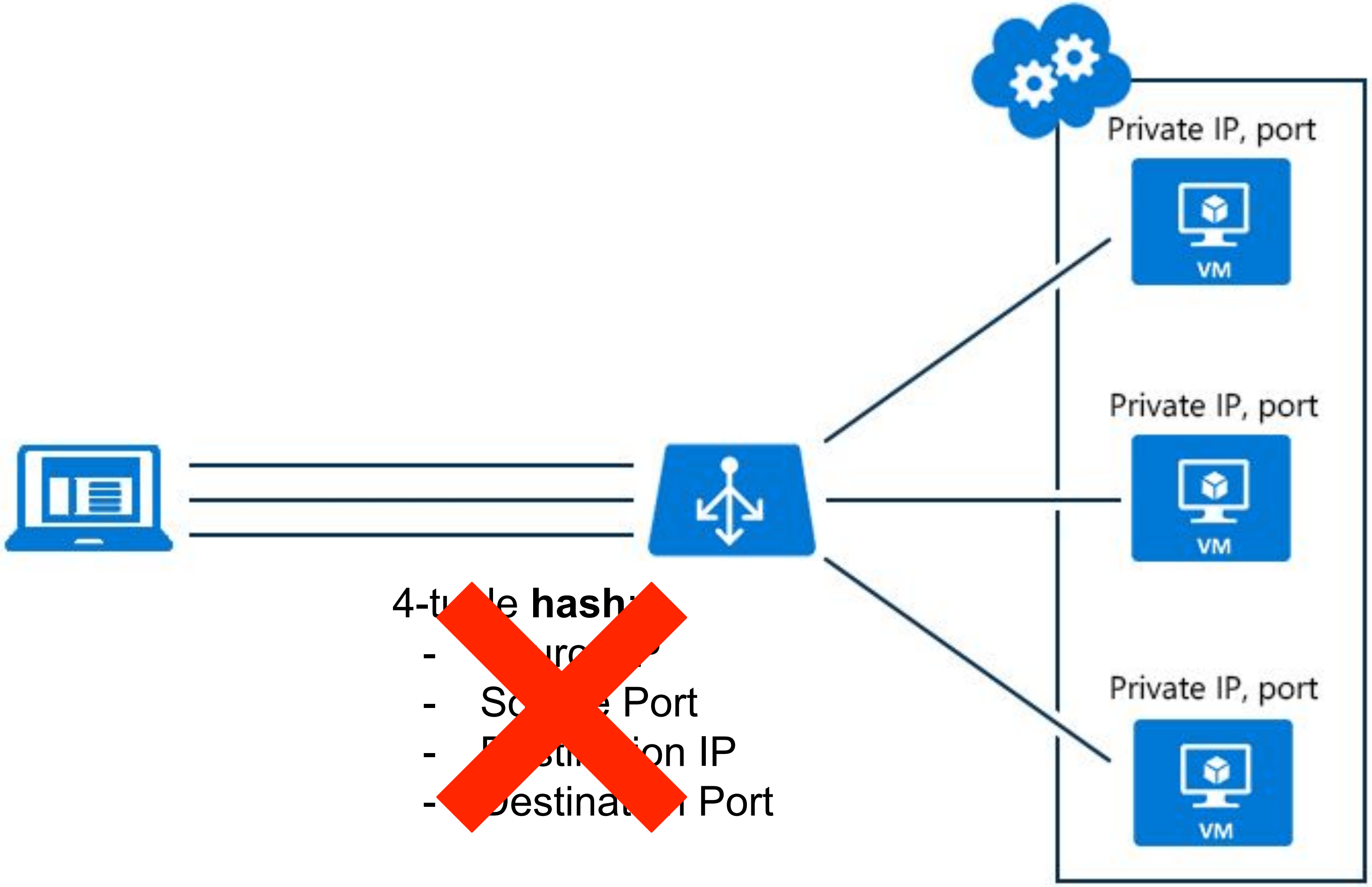


Stateless Load Balancing changes



4-tuple **hash**:
- Source IP
- Source Port
- Destination IP
- Destination Port

Stateless Load Balancing changes

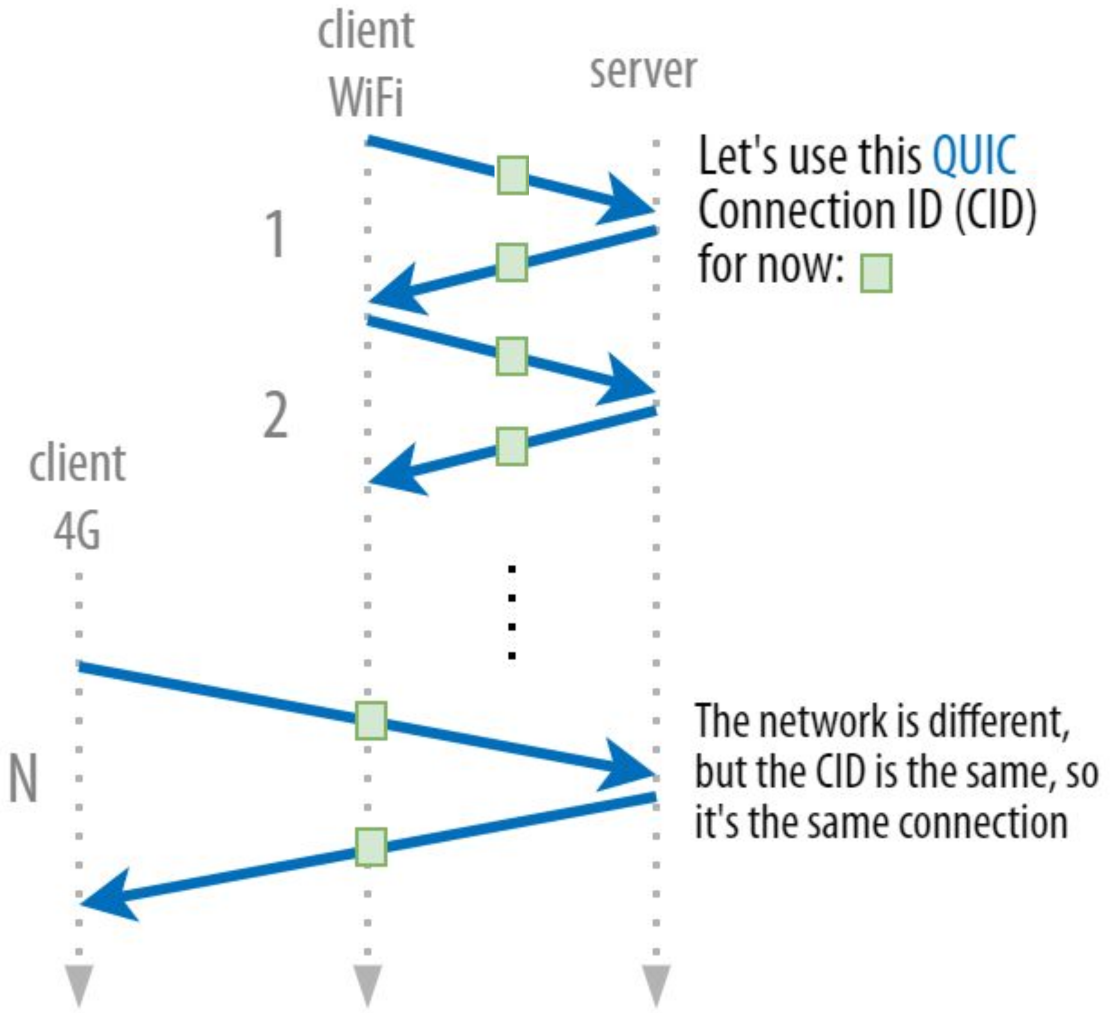


~~**4-tuple hash:**~~

- ~~Source IP~~
- ~~Source Port~~
- ~~Destination IP~~
- ~~Destination Port~~

Connection ID

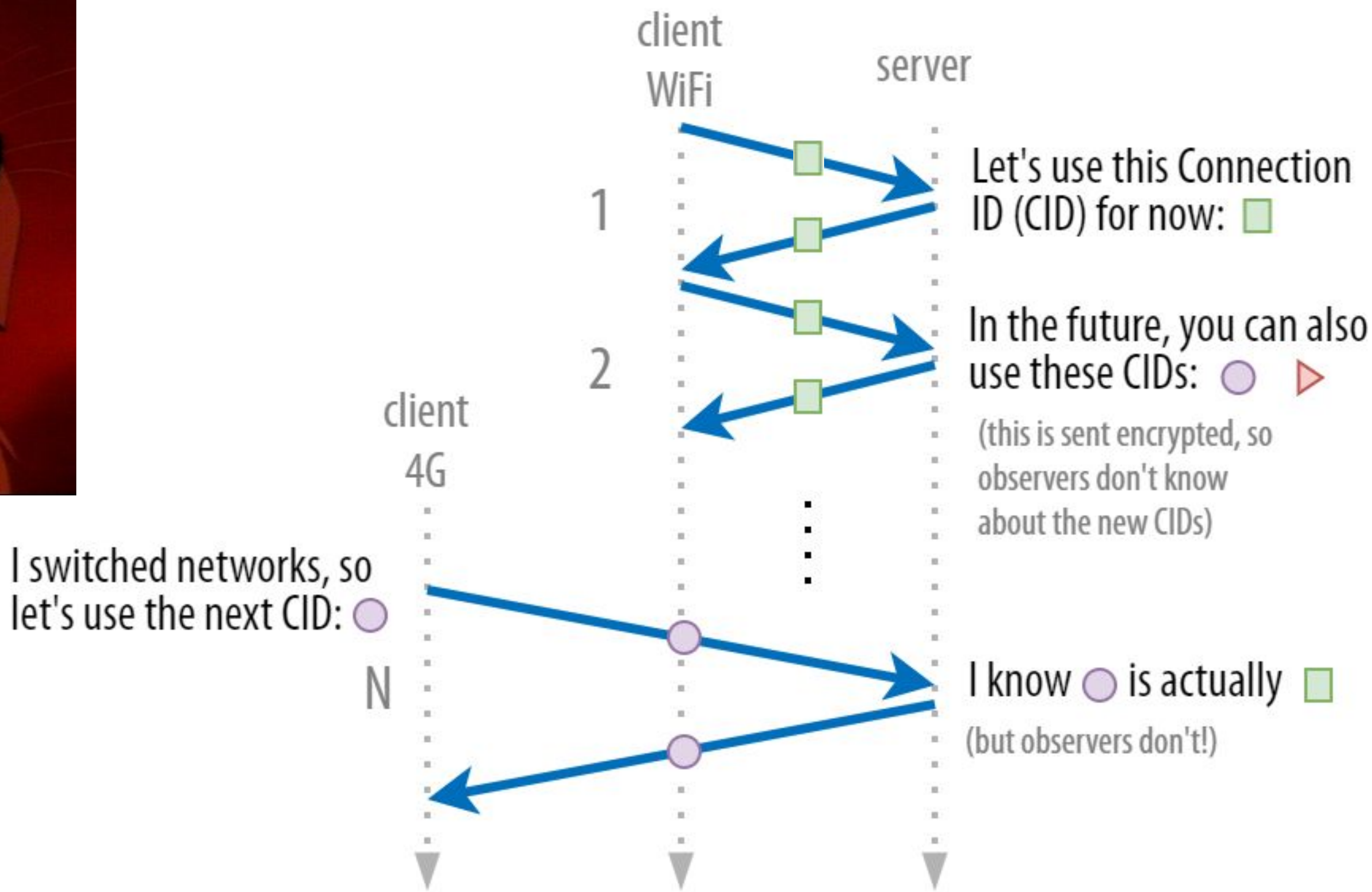
CIDs as a Geographical Tracking Vector



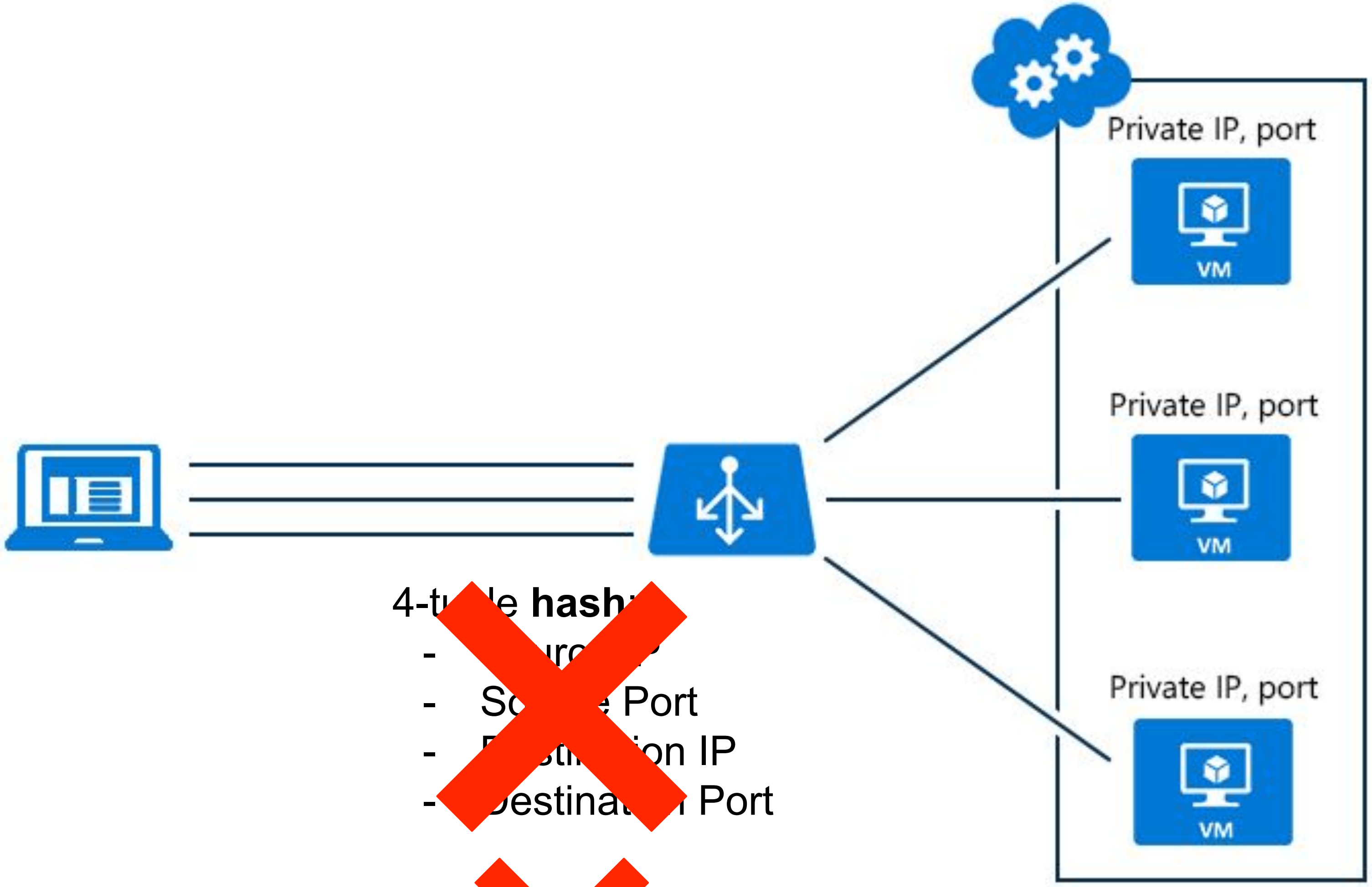
Linkability problem:

CID can be used to track users across different networks!

Solution: Change CIDs when network changes!



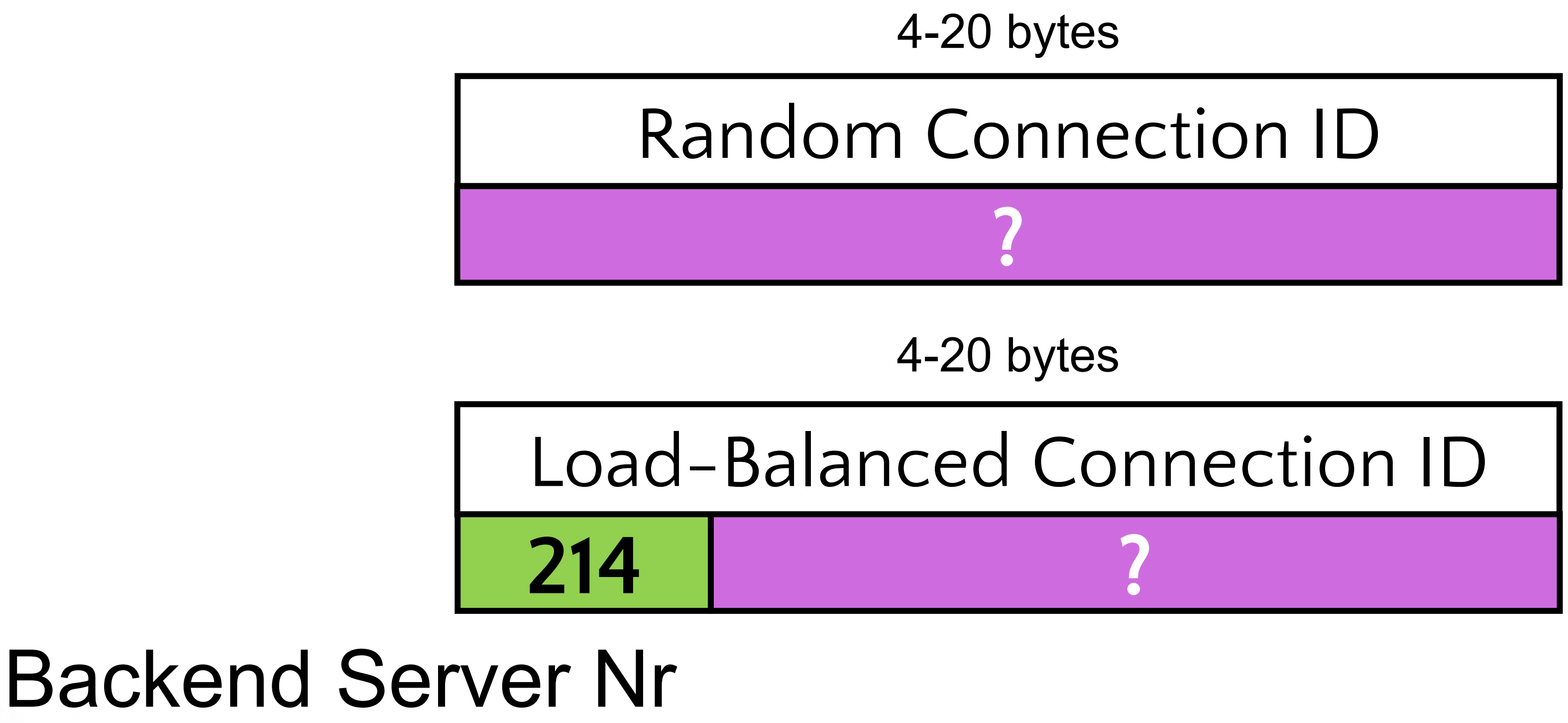
Problem: Breaks stateless load balancing…



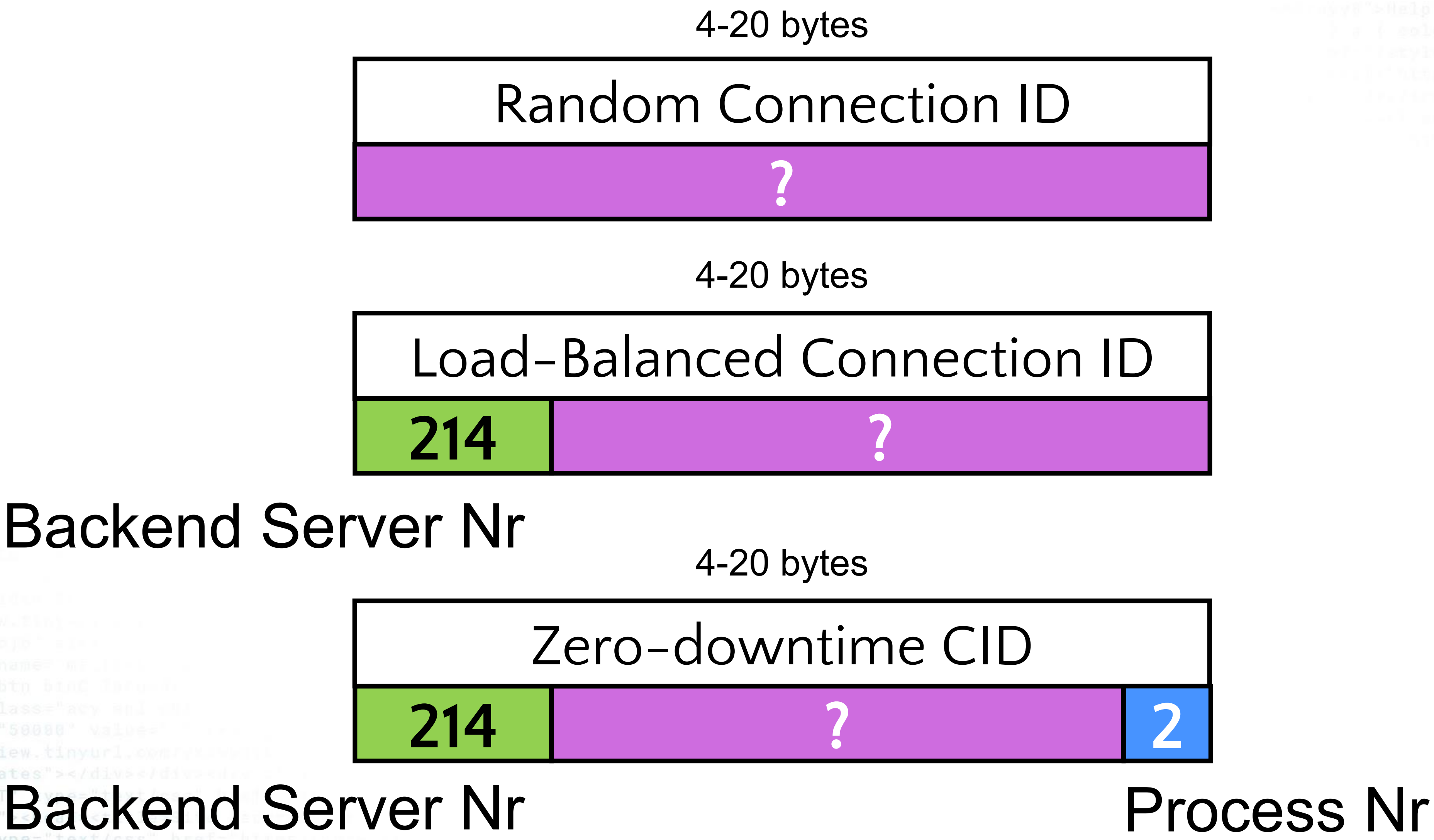
4-tuple hash:
- Source IP
- Source Port
- Destination IP
- Destination Port

Connection ID

Solution: encode LB information in CID



Solution: encode LB information in CID

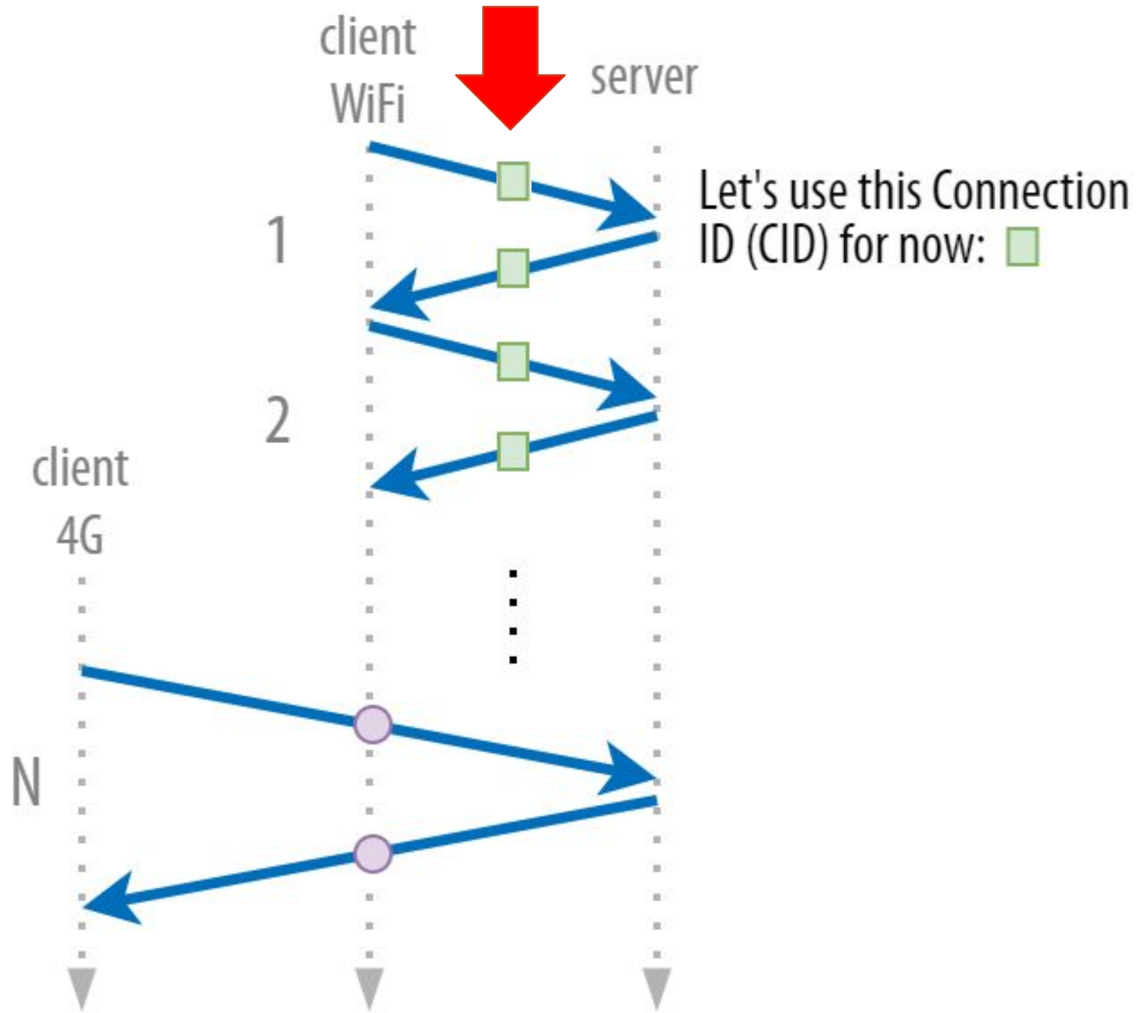


EPIQ 18 Facebook Keynote: https://www.youtube.com/watch?v=8lYHNzoPS2o

Problem: Who chooses the CID?

Client chooses CID

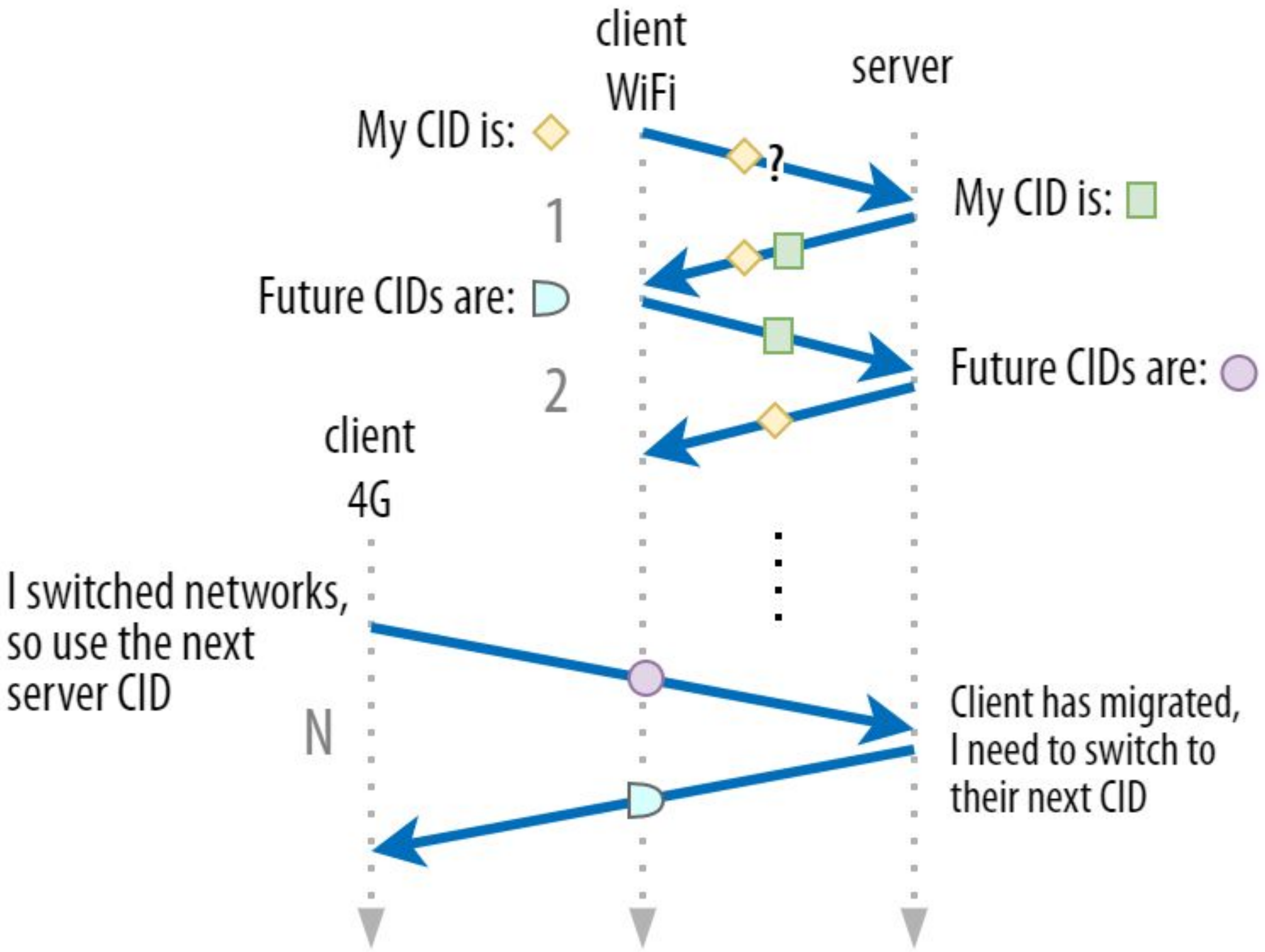
But client doesn't have Load Balancing info…



Akamai Experience the Edge

Solution: Separate CIDs for Client and Server

Different directions use different CIDs

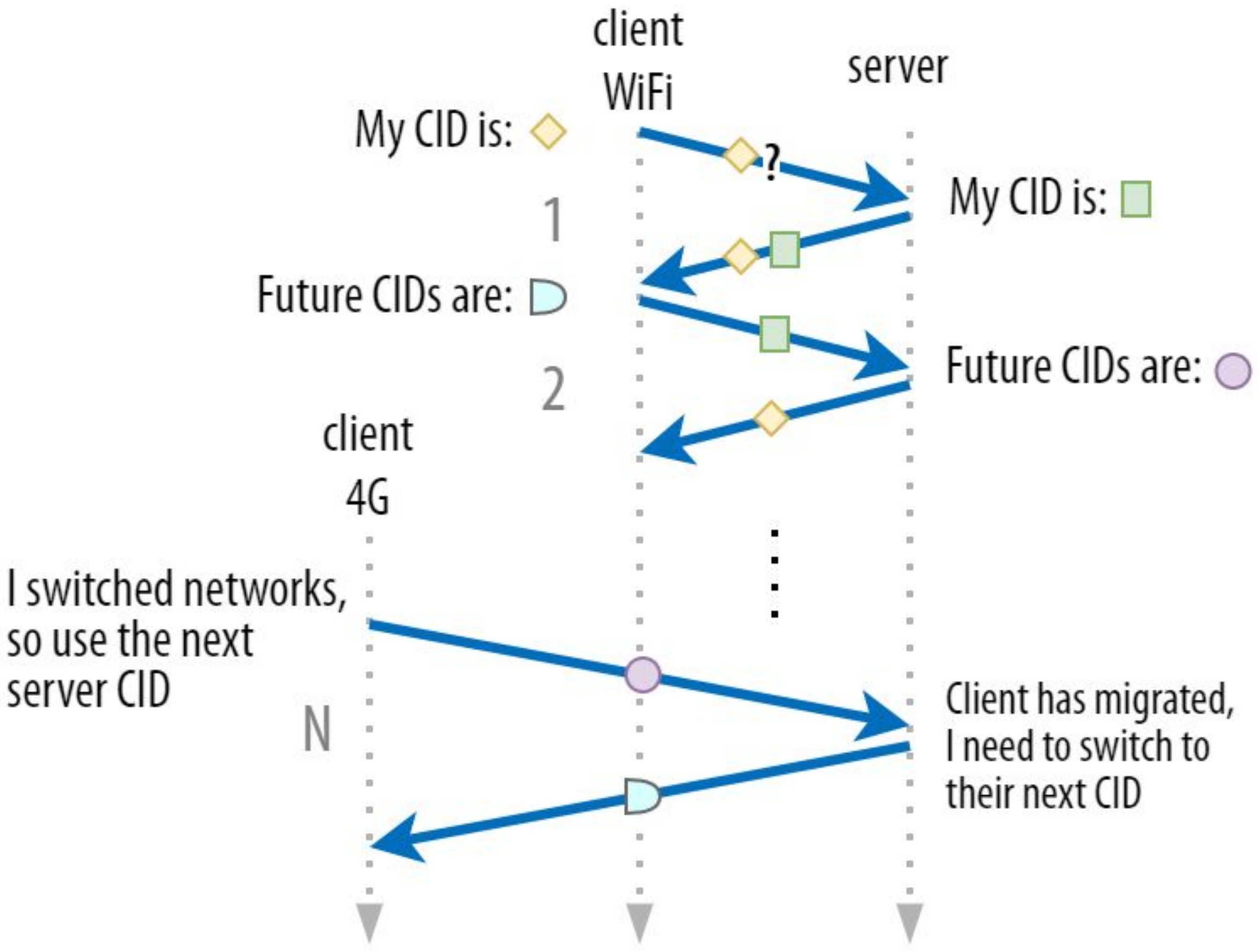


Solution: Separate CIDs for Client and Server

Different directions
use different CIDs

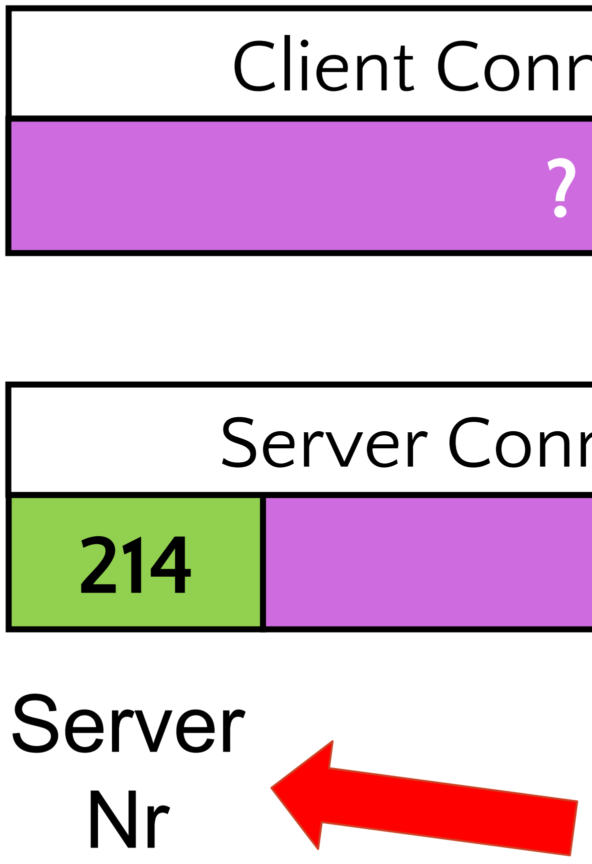
```
Handshake Packet {
  Header Form (1) = 1,
  Fixed Bit (1) = 1,
  Long Packet Type (2) = 2,
  Reserved Bits (2),
  Packet Number Length (2),
  Version (32),
  Destination Connection ID Length (8),
  Destination Connection ID (0..160),
  Source Connection ID Length (8),
  Source Connection ID (0..160),


  Length (i),
  Packet Number (8..32),
  Packet Payload (8..),
}
```



Akamai Experience the Edge

Problem: Leaking internal deployment info in CID



- Figure out #servers / LB logic
- **DDoS 1 specific server**

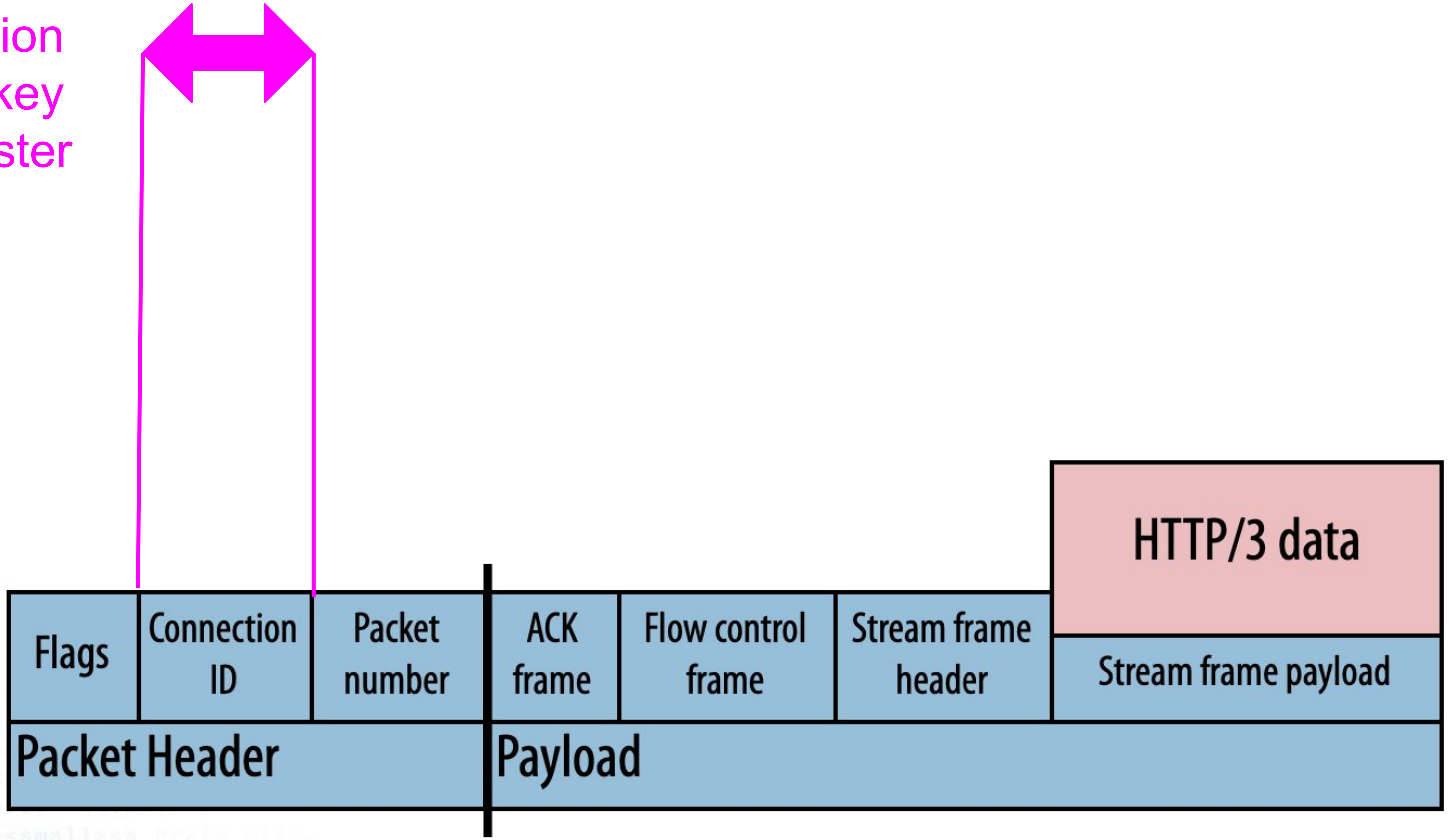
“Facebook network and load balancing internals from CID observations” : https://arxiv.org/abs/2209.00965

Solution: Encrypt CID

Encrypt CID at the server, decrypt at Load Balancer

- Can use **single** shared private key across deployment/cluster

CID encryption
- Single key
- Per cluster



QUIC Load Balancing RFC candidate: https://quicwg.org/load-balancers/draft-ietf-quic-load-balancers.html

Tip of the Iceberg

- server migration / preferred address
- retiring connection IDs
- CID and key updates
- Transport parameter negotiation
- Long vs Short packet headers
- zero-length CIDs
- MULTIPATH?!?!?
- Packet coalescing and chaos protection/GREASE
- RETRY and STATELESS RESET
- Happy eyeballs
- 0-RTT ticket re-use
- Replay attacks
- Session resumption, address validation and STEKs
- Encrypted Client Hello (ECH)
- DNS HTTPS records
- ...

Key challenges:

- Load balancing
- Firewalling
- 0-RTT + session resumption
- TLS key/certificate management

Akamai also does Cloud now

End-to-end qlog extraction

HTTP/3 test page - all options

Process qlog

{"qlog_format":"JSON","qlog_version":"0.3","traces":[{"common_fields":{"ODCID":"e16e8b769bc5e801"},"events":[{"data":{"count":1,"raw":[{"length":1258,"payload_length":1250}]},"name":"transport:datagrams_received","time":1692887275647.8293},{"data":{"owner":"local","original_destination_connection_id":"e16e8b769bc5e801","max_idle_timeout":60000,"stateless_reset_token":"b957081308025907d52ee833fa52dff7","initial_max_data":1048576,"initial_max_stream_data_bidi_local":1048576,"initial_max_stream_data_bidi_remote":1048576,"initial_max_stream_data_uni":1048576,"initial_max_streams_bidi":128,"initial_max_streams_uni":128,"ack_delay_exponent":3,"max_ack_delay":25,"disable_active_migration":false,"active_connection_id_limit":8,"initial_source_connection_id":"b968b91175caa7c1","max_datagram_frame_size":65536},"name":"transport:parameters_set","time":1692887275650.6865},{"data":{"key_type":"client_initial_secret","trigger":"tls"},"name":"security:key_updated","time":1692887275651.8425},{"data":{"key_type":"server_initial_secret","trigger":"tls"},"name":"security:key_updated","time":1692887275652.1306},{"data":{"frames":[{"frame_type":"padding"},{"frame_type":"crypto","length":114,"offset":331},{"frame_type":"ping"},{"frame_type":"crypto","length":291,"offset":31},{"frame_type":"ping"},{"frame_type":"padding"},{"frame_type":"crypto","length":31,"offset":0},{"frame_type":"padding"},{"frame_type":"ping"},{"frame_type":"padding"},

https://github.com/http3-prioritization/prioritization-test-page

Conclusion


HUNGRY


QUIC
YOU
HTTP/3

Conclusion

HTTP/3

Complex, yes, but mostly just **different**

Ease yourself into it:

https://www.youtube.com/playlist?list=PL3tsOU35YefabIEzJa_cq6vjkojNr0UZh
https://www.smashingmagazine.com/2021/08/http3-core-concepts-part1/
https://calendar.perfplanet.com/2022/http-3-prioritization-demystified/
https://calendar.perfplanet.com/2020/head-of-line-blocking-in-quic-and-http-3-the-details/
https://www.researchgate.net/profile/Robin-Marx-2

Would you like to know more?

Robin Marx

@programmingart

linkedin.com/in/rmarx

rmarx@akamai.com

Akamai Experience the Edge

Image sources

Crying dwarf: https://www.pinterest.com/pin/241998179947602806/

Crocodile: https://static.wikia.nocookie.net/villains/images/0/09/Crocodile_Disney.jpg/revision/latest?cb=20180430043230

Captain hook: https://i.ytimg.com/vi/85dY-6AMcec/maxresdefault.jpg

Snow white bed: https://www.1828.org.uk/2020/04/10/what-snow-white-and-the-seven-dwarfs-can-teach-us-about-economic-recoveries/

Grumpy: https://insidethemagic.net/2021/06/grumpy-snow-white-backstory-ad1/

Dopey: https://www.etsy.com/uk/listing/948566803/im-a-little-dopeydisney-snow-white-the

Icepick: https://japaneseknifecompany.com/product/ice-pick-with-axe-240mm/

Peter Pan and the gang: https://movies.disney.com/peter-pan

Peter pan and the lost boys: https://disney.fandom.com/wiki/Lost_Boys

Pan vs Hook: https://www.disneyclips.com/images/peterpan4.html

Evil pan: https://screenshot-media.com/culture/entertainment/peter-pan-slasher-film/

Vampire Pan: https://goodimprov.com/2020/03/26/peter-pan-is-a-vampire/

Hook: https://chisholmcountry.com/2022/08/17/getting-hooked-on-adventure/

Step sister foot: https://yourfriendinreykjavik.com/icelandic-cinderella-mjadveig/

Cinderella stairs: https://www.pinterest.com/pin/784470828828625826/

Step sister bloody: https://oss.adm.ntu.edu.sg/nurjanna001/goldberg-machine-the-grimm-brothers-cinderella/

Pumpkin carriage: https://gifdb.com/gif/cinderella-pumpkin-transformation-s1h6y2ddi3cdzh96.html

memory lane: https://medium.com/@lachachi.ghizlene/googling-a-trip-down-memory-lane-2aa3e7499c54

Lucifer cup: https://i0.wp.com/www.caps.media/195/0-cinderella/full/cinderella-disneyscreencaps.com-2180.jpg?strip=all

Dead Mufasa: https://www.cheekiemonkie.net/2012/04/i-am-no-superman.html

Scar smiling: https://news.disney.com/scar-quotes-lion-king

Scar scared: https://m.imdb.com/title/tt0110357/mediaviewer/rm1097297920/

Canary Icon: https://www.flaticon.com/free-icon/canary_3362347

Skull Icon: https://visualpharm.com/assets/914/Skull-595b40b85ba036ed117dbabc.svg

Sever Icon: https://creazilla.com/nodes/3254772-server-icon

Firewall Icon: https://iconduck.com/icons/207872/firewall

Slowly logo: https://dribbble.com/shots/5559292-slow

Peter Pan thinking: https://paintbynumbers.uk/products/peter-pan-new-paint-by-numbers/

Tinkerbell angry: https://www.pinterest.com/pin/835628905834514765/

Mufasa: https://disney.fandom.com/wiki/Mufasa

Scar Kills mufasa: https://www.imdb.com/news/ni64059642/

Simba vs scar: https://www.youtube.com/watch?v=7TeZY6Nvn9U

Robin Hood rhinos: https://www.reddit.com/r/meme/comments/jrw2ox/the_internet_will_get_it/

Robin Hood standing: https://dmk.fandom.com/wiki/Robin_Hood?file=Cp-robin_hood.png