A quick look at QUIC

Geoff Huston AM
Chief Scientist , APNIC

November 2023
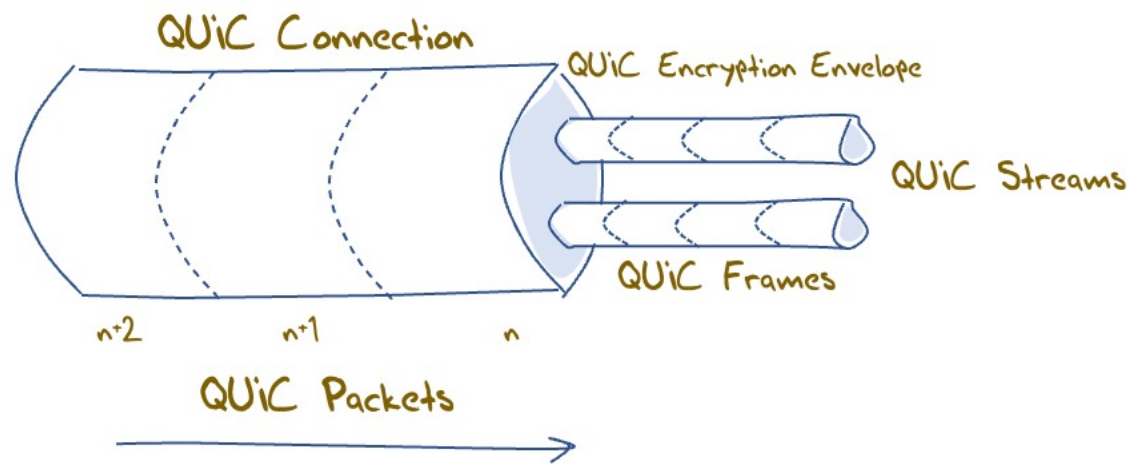
TCP + TLS + HTTP = HTTP/2

HTTP/2

HTTP Multi-stream
TLS Session Encryption
TCP Data stream integrity Congestion Control
IP

e2e encrypted

UDP + QUIC = HTTP/3

HTTP/2

**QUIC
HTTP/3**

e2e encrypted

HTTP/2
HTTP Multi-stream
TLS Session Encryption
TCP Data stream integrity Congestion Control

QUIC HTTP/3
HTTP
QUIC Multi-stream Encryption Data stream integrity Congestion Control
UDP

e2e encrypted

IP
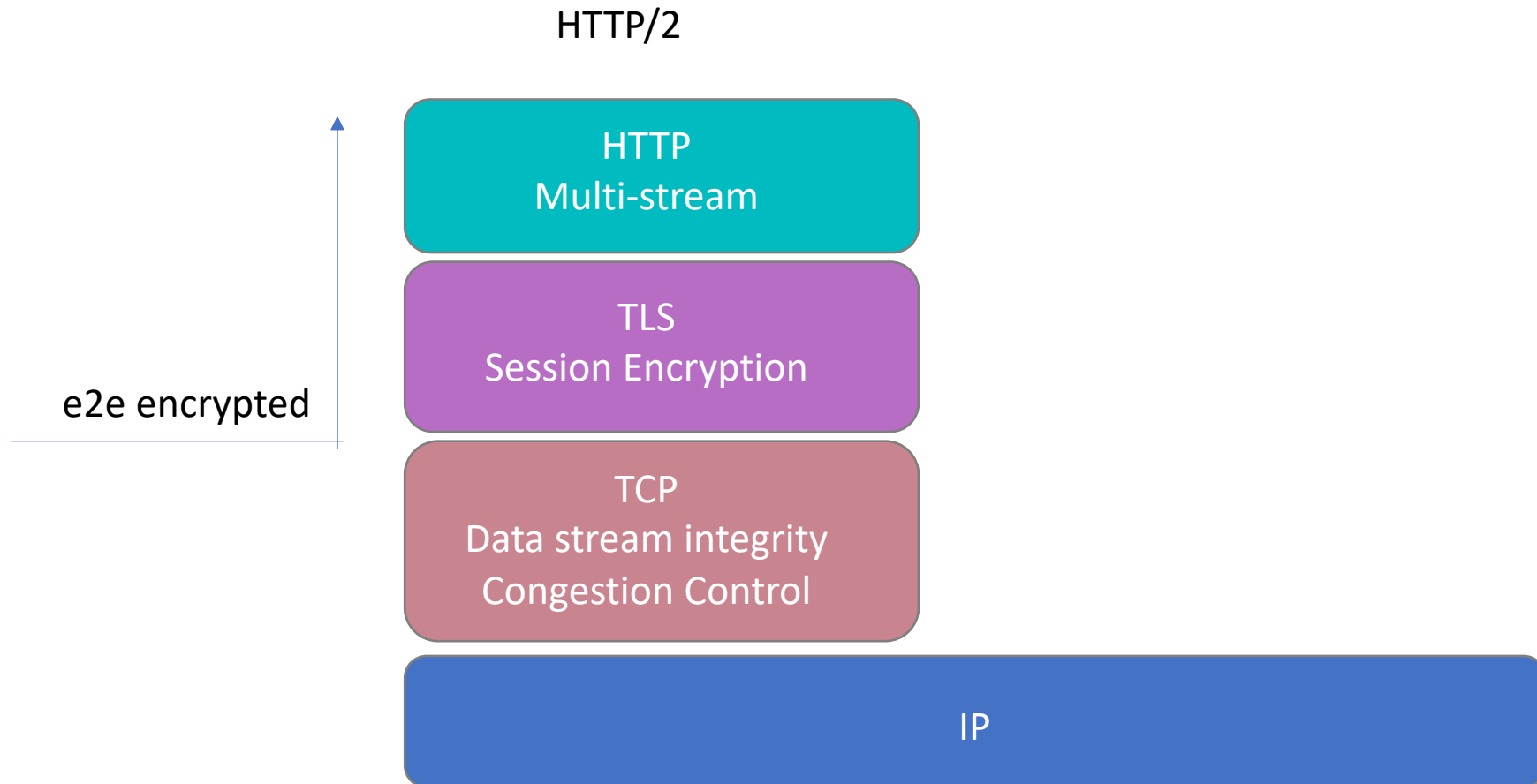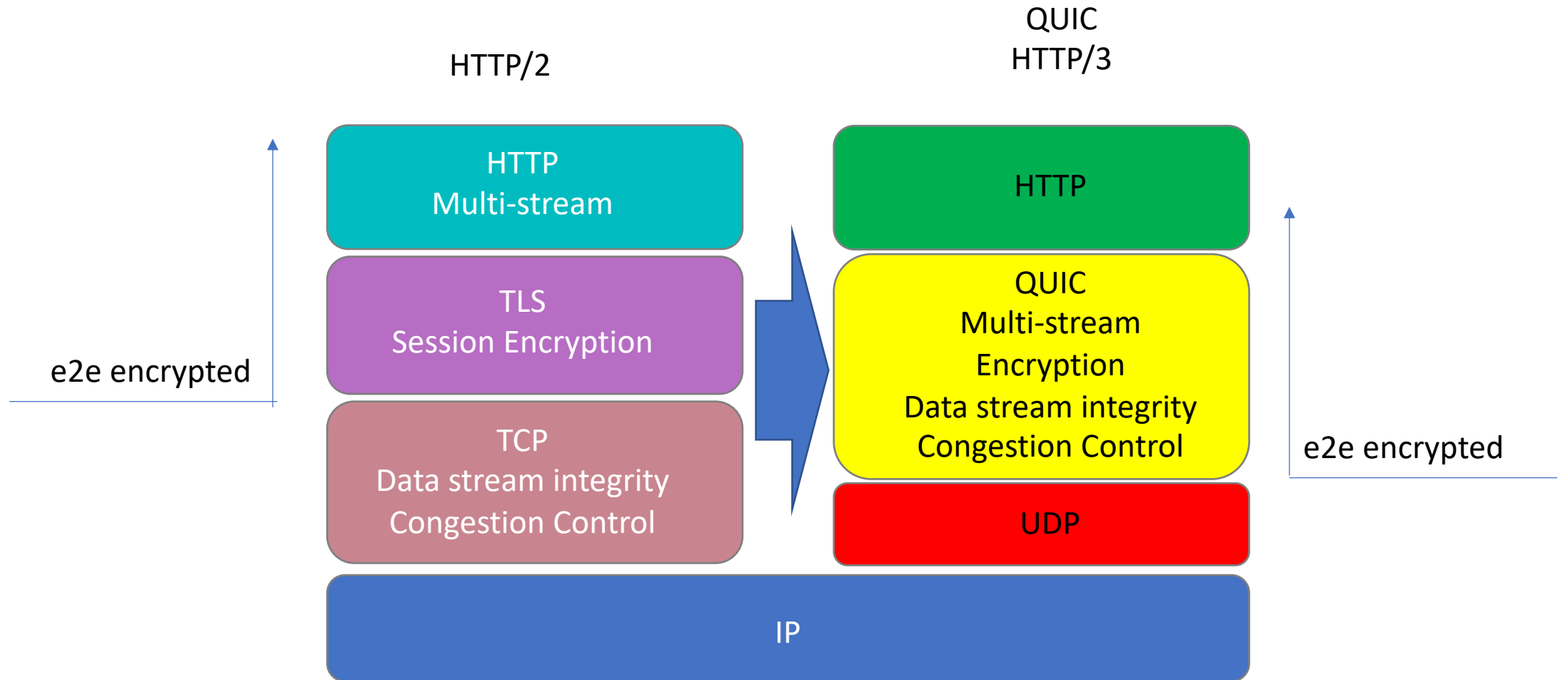
QUIC flow structuring

A QUIC connection is broken into "streams" which are reliable data flows – each stream performs stream-based loss recovery, congestion control, and relative stream scheduling for bandwidth allocation

QUIC also supports unreliable encrypted datagram delivery

QUIC is…

So much more than just "*encrypted TCP over UDP*"

- Support for multi-stream multiplexing that avoids head-of-line blocking and exploits a shared congestion and encryption state
- Faster Startup - Combines transport and encryption setup exchange in a single 3-way exchange at session start, and supports fast reopen
- Customisable - QUIC implementations can use individual flow controllers per flow
- QUIC places its transport control fields inside the encryption envelope, so QUIC features minimal exposure to the network
- Its "address agile" and can work through NATs
- Supports record and Remote Procedure Call service models as well as bit-streaming and datagram services

QUIC is:

- A logical evolutionary step for transport services, providing more flexibility, faster connection setup, and a larger set of transport services

- It's what we should expect from a capable modern transport protocol!

Triggering QUIC in HTTP

Use the DNS to trigger QUIC:

- Set up an HTTPS record for each server name, with value: `alpn="h3"`

Use content-level controls to trigger QUIC:

- Add `Alt-Svc: h3=":443"` to the HTML headers

First Fetch

Second Fetch

Setting Expectations

- Chrome has a dominant share of browser instances - roughly, some 65%*

- And Chrome has been supporting a switch to QUIC via the Alt-Svc directive since 2020

- And Apple Safari is now supporting QUIC, using the DNS `apln` directive

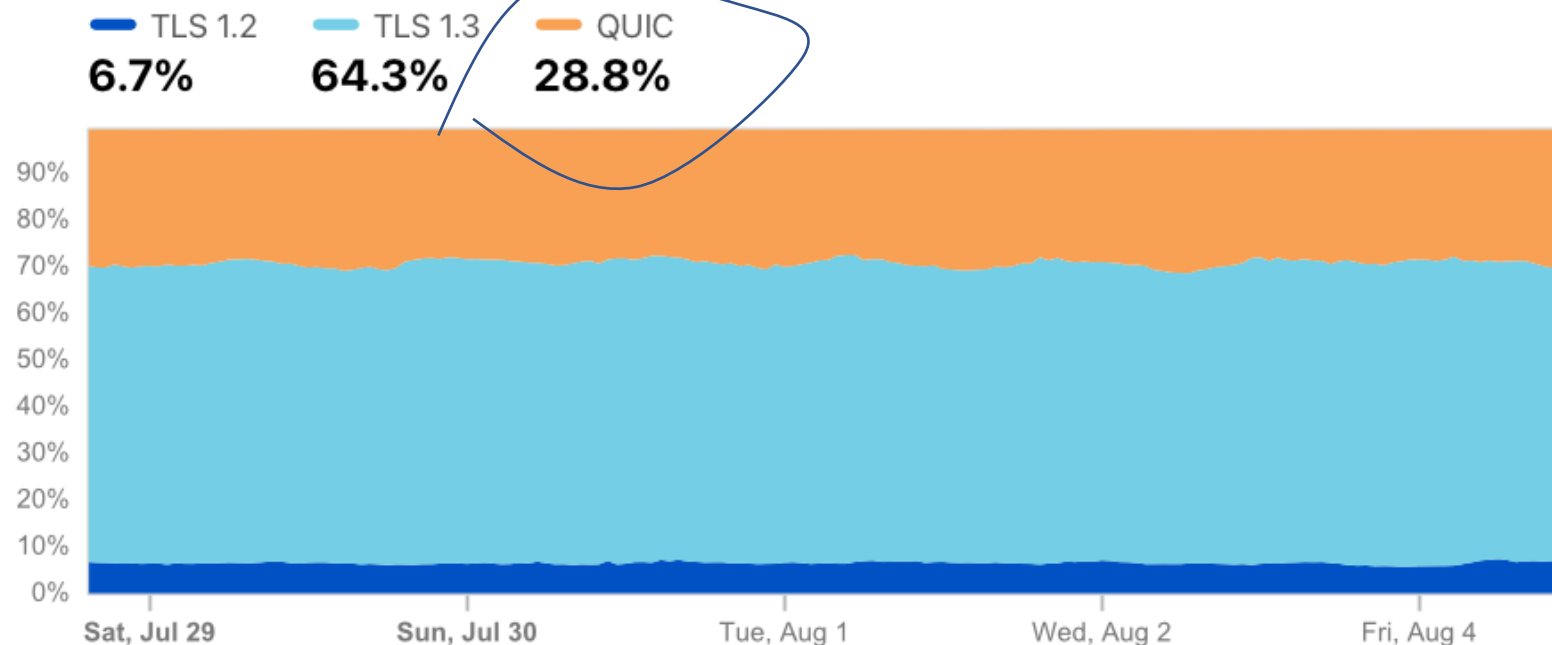- So a QUIC-aware server platform should be seeing some 85% of its sessions using QUIC – right?

Cloudflare's Numbers
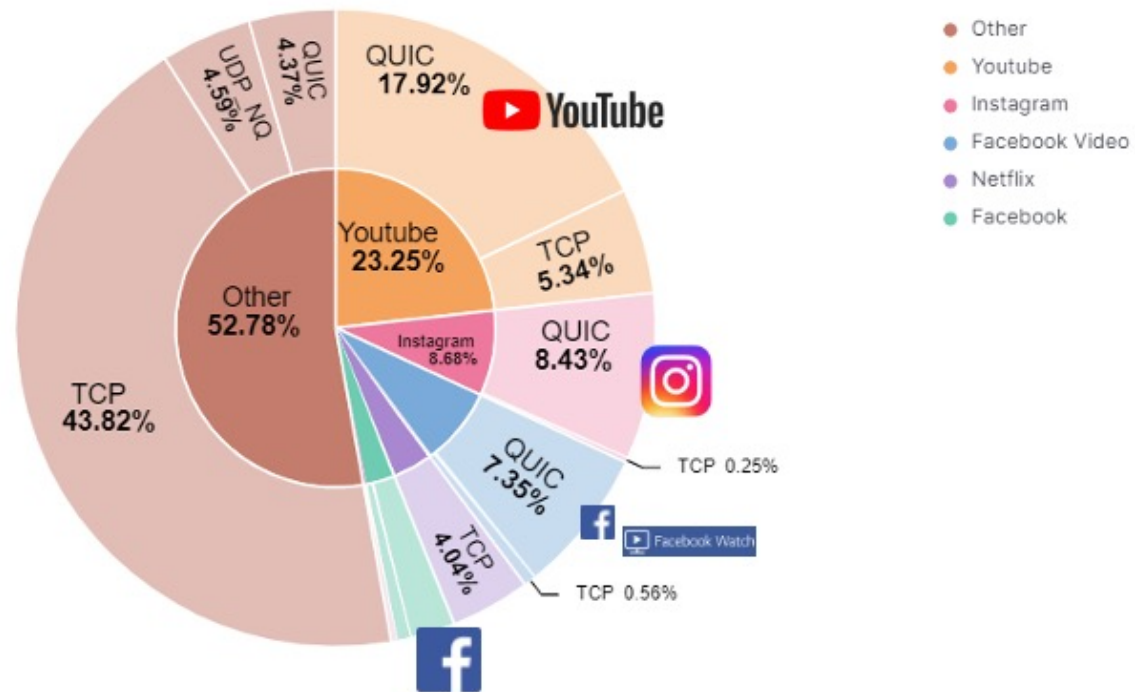
Cloudflare reports a 29% use of QUIC

ISP Numbers

*source EU Operator 2022

APNIC's Measurement of QUIC Use

Zoom: 1h 1d 5d 1w 1m 3m 6m 1y max

● Quic First Fetch : 1.4 ● QUIC Second Fetch : 4.45 | 17:00 July 18, 2022

Subsequent Fetches – mainly Chrome clients

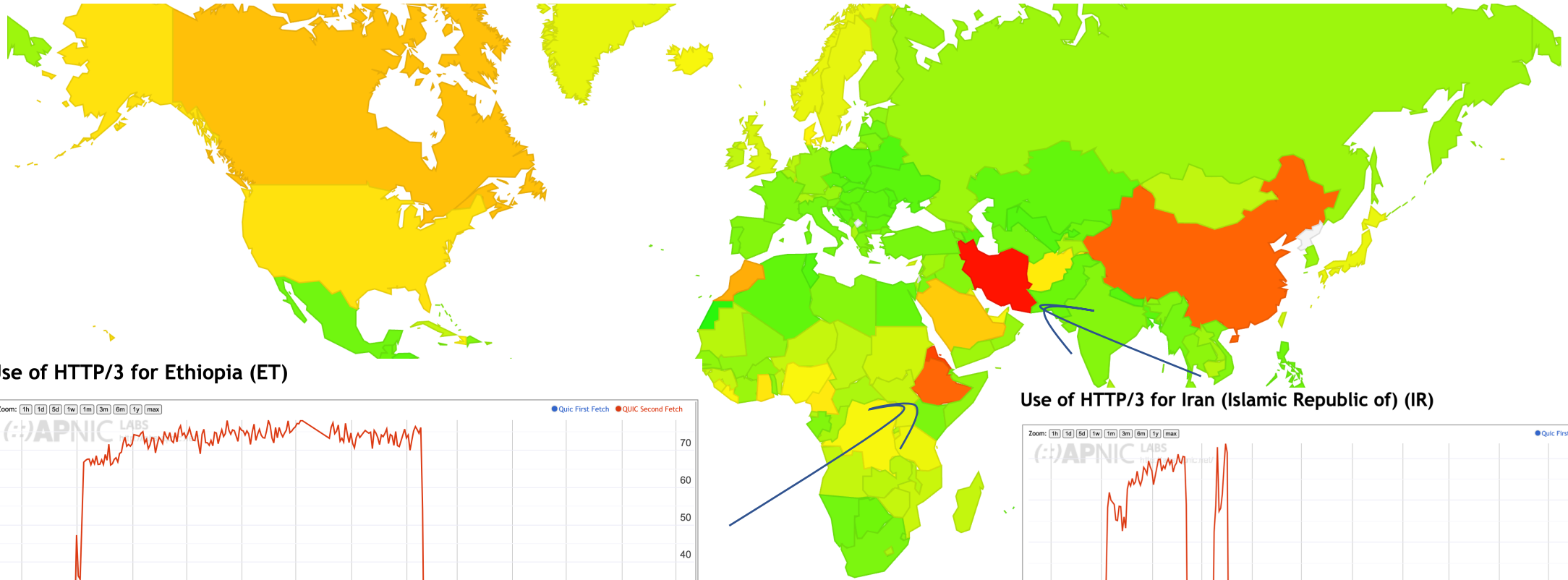First Fetch – mainly Safari clients
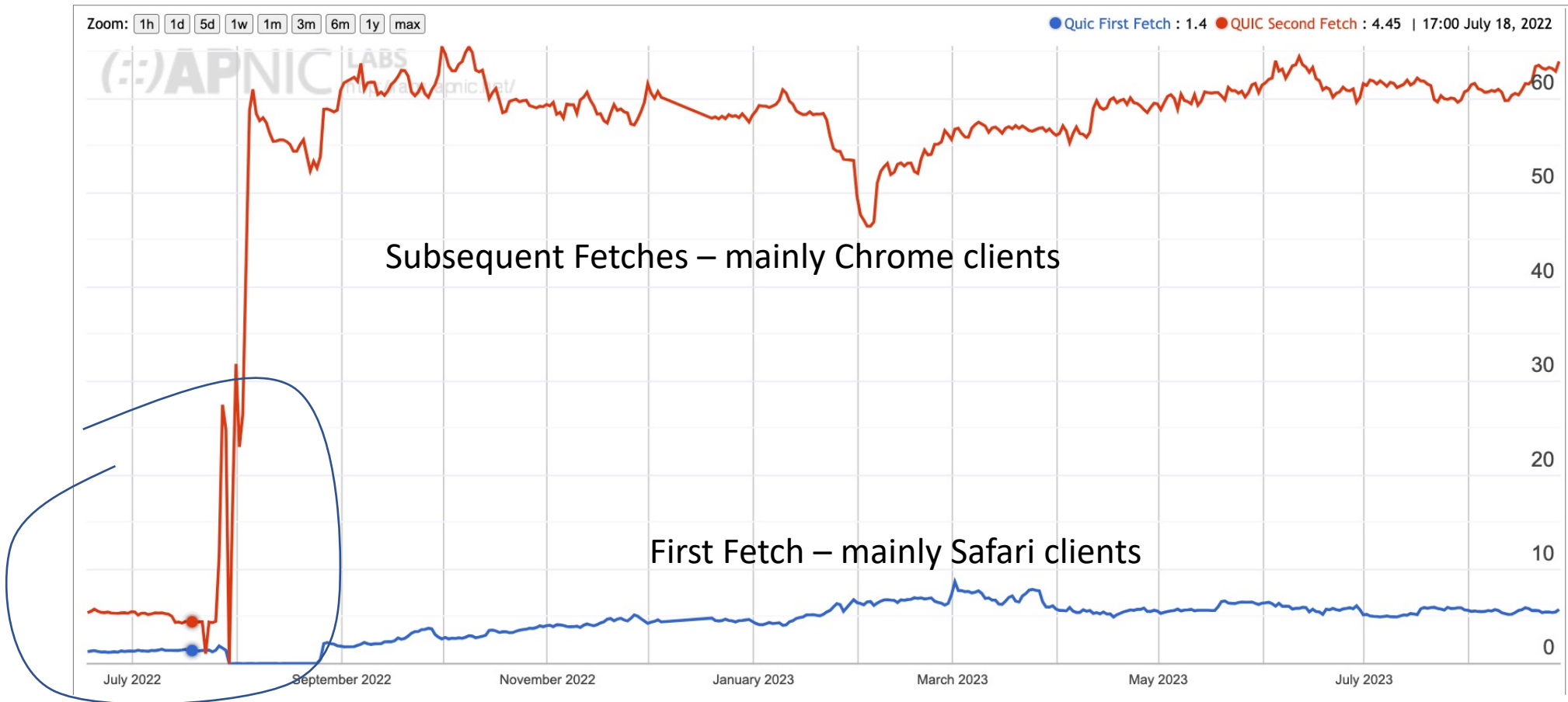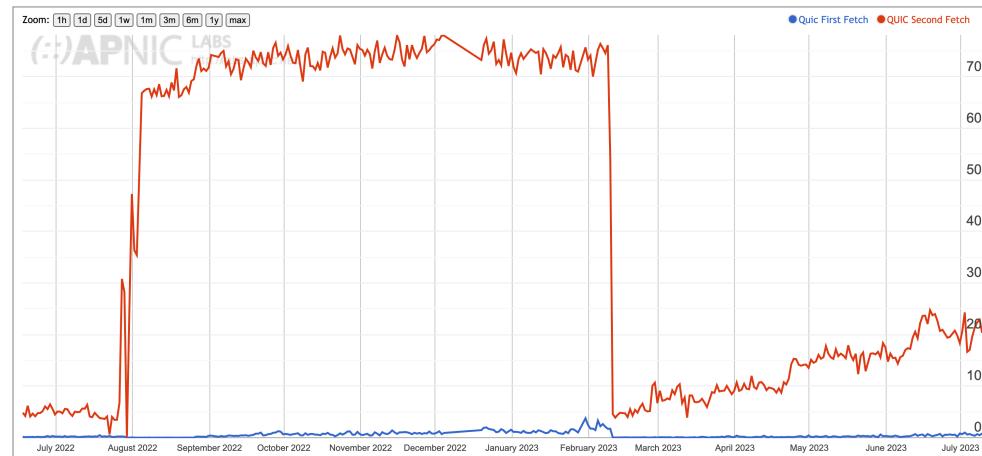
Playing with keepalive parameters!
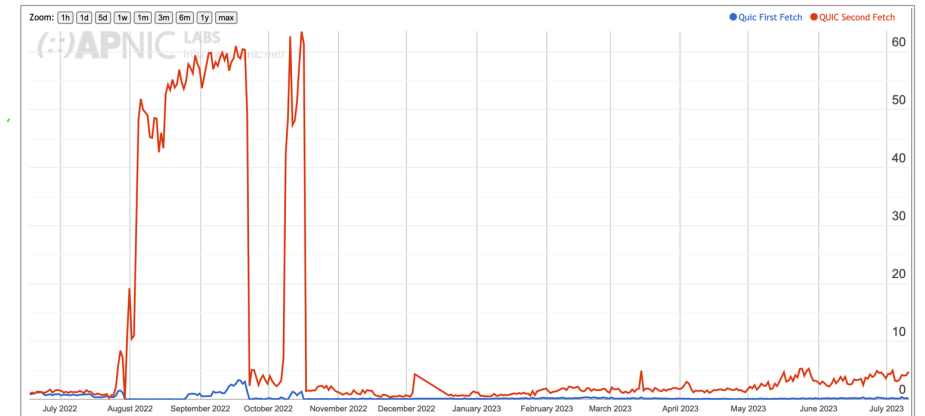
QUIC Use - July 2023
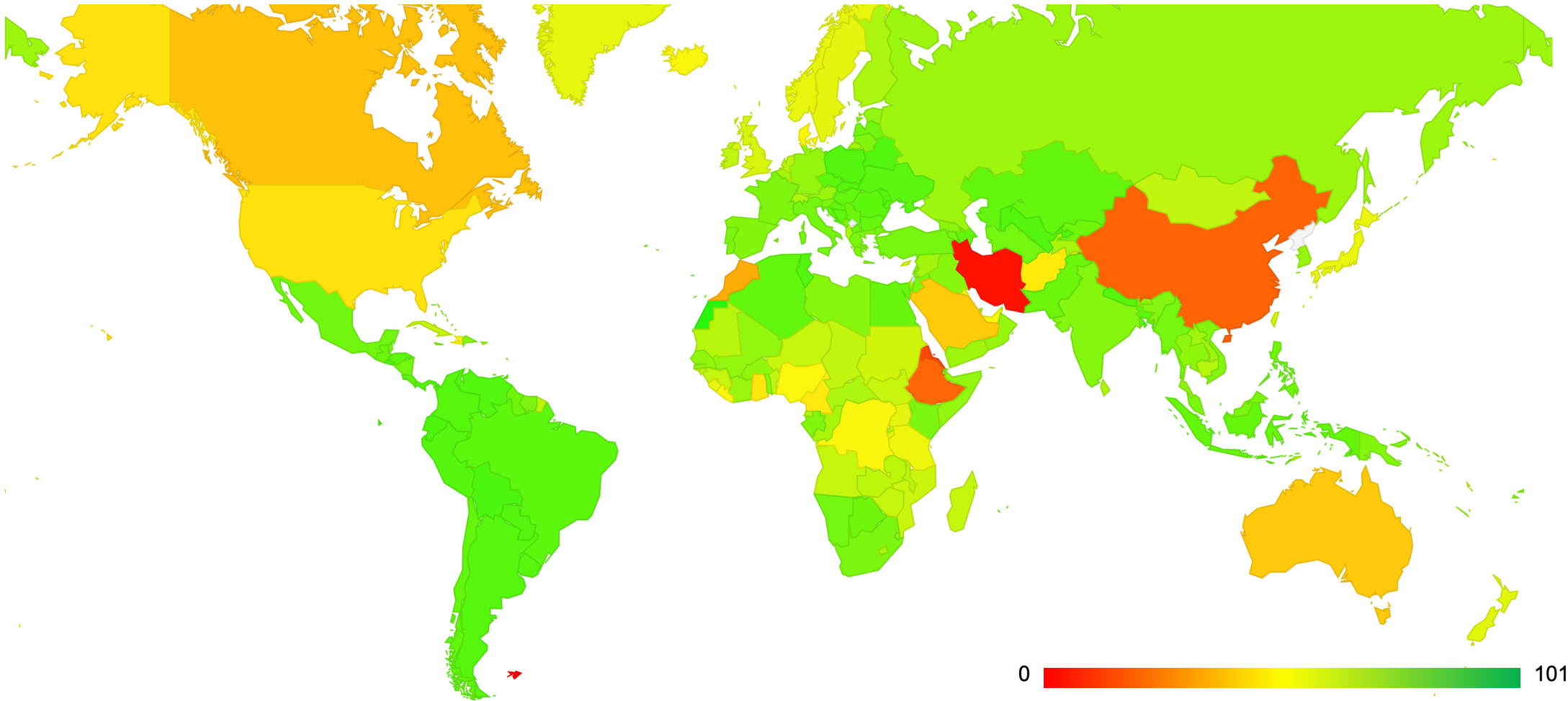
National Filtering of QUIC?
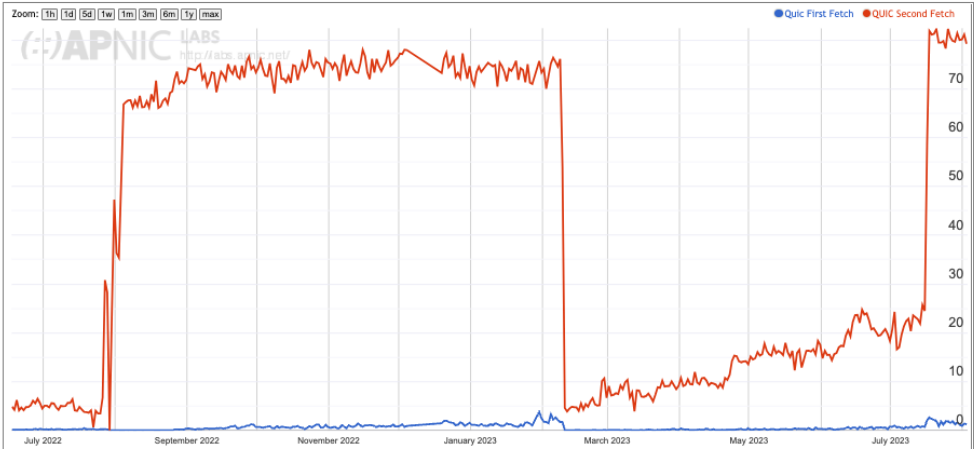

Use of HTTP/3 for Ethiopia (ET)
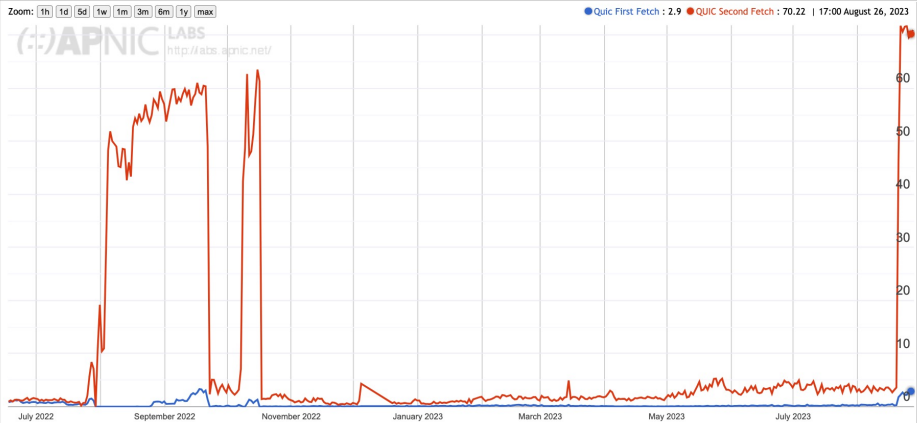
Use of HTTP/3 for Iran (Islamic Republic of) (IR)

National Filtering of QUIC?

Use of HTTP/3 for Ethiopia (ET)

Use of HTTP/3 for Iran (Islamic Republic of) (IR)

W
TTP/3 Use: **87.43%**

Why is QUIC important?

Because QUIC is fast

Because QUIC encrypts everything
- No visible transport control settings
- No visible Server Name Indication in the crypto-setup
- No visible traffic profile other than inter-packet timing
- And if you use a MASQUE-based VPN then there no residual visibility!

Because QUIC is an application capability
- QUIC can interact with the platform through the UDP API, so all of QUIC can be implemented within the application. This gives the application more control over its service outcomes and reduces external dependencies

What does this mean for TCP?

It's not looking all that good for TCP's prospects

- QUIC not only does faster start up, but it supports multi-channel in a frictionless manner

- QUIC resists network operator efforts to perform traffic shaping through direct manipulation of TCP control parameters

- QUIC allows the application service provider to control the congestion behaviour of its sessions

The new Networking Space

And this is why QUIC is so interesting – it is pushing both network carriage and host platform into commodity roles in networking and allowing applications to effectively customize the way in which they want to deliver services and dominating the entire networked environment

QUIC is the application's view of what Transport should be!

QUiC and value transform in the network stack
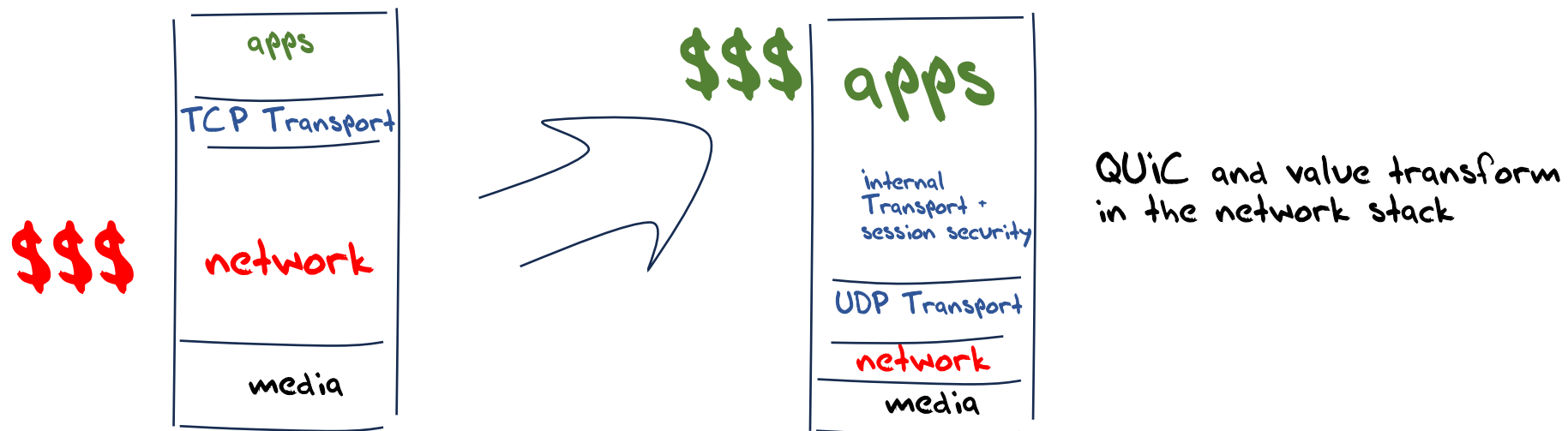
What does this mean for the Internet?

- The relationship between applications, hosts and networks has soured into mutual distrust and suspicion

- The application now defends its integrity by wrapping up as much of the service transaction with encryption and indirection

- QUIC (and MASQUE) is an intrinsic part of this process of wrapping up traffic in encryption and redirection

- For the network operator there is little left to see

- And I suspect that there is no coming back from here!

The new Internet Space

"What you can't dominate, you commoditise*"

- Vertically integrated service providers have faded away into history - the deregulated competitive service industry continues to specialize rather than generalize at every level

- Carriage is no longer an inescapable monopoly - massively replicated content can be used as a substitute for many carriage service elements

- Control over the platform is no longer control over the user. Operating systems have been pushed back into a basic task scheduling role, while functions are being absorbed into the application space

* A related quote is Peter Thiel's "Competition is for losers!"

Thanks!