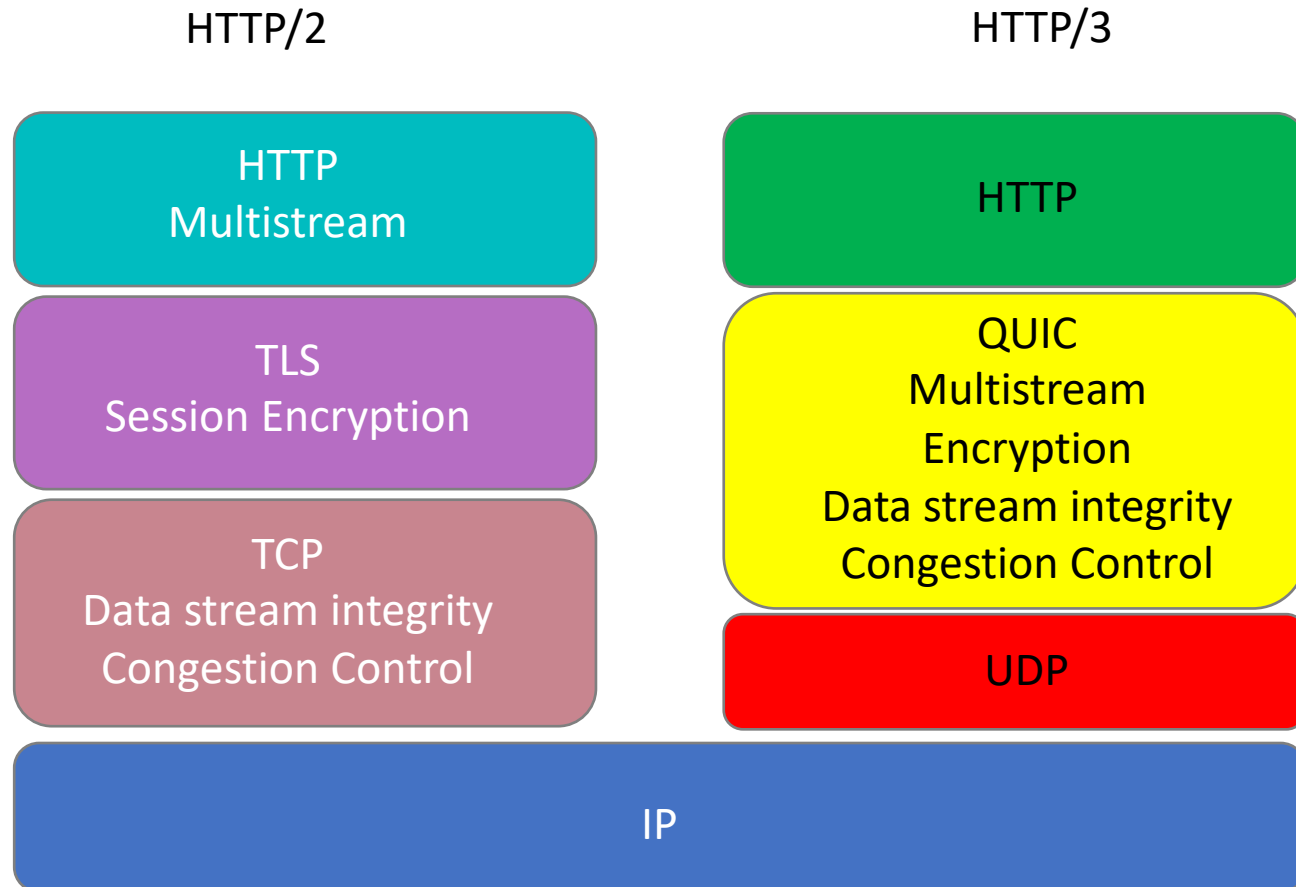


A Quick look at QUIC

Geoff Huston, Joao Damas,

APNIC Labs

QUIC is...



QUIC is...

A transport level framing protocol that offers applications access to the basic IP datagram services offered by IP through the use of UDP

All other transport services (data integrity, session control, congestion control) are shifted to the application

QUIC also places the transport control fields inside the encryption envelope, so QUIC has minimal exposure to the network

Looking for QUIC

- At APNIC we use Ads to perform large scale measurements of network service capabilities as seen by users
 - IPv6 deployment
 - DNSSEC validation
 - Fragmentation
- Can we use this measurement platform to see the level of use of QUIC in today's network?

Setting up QUIC

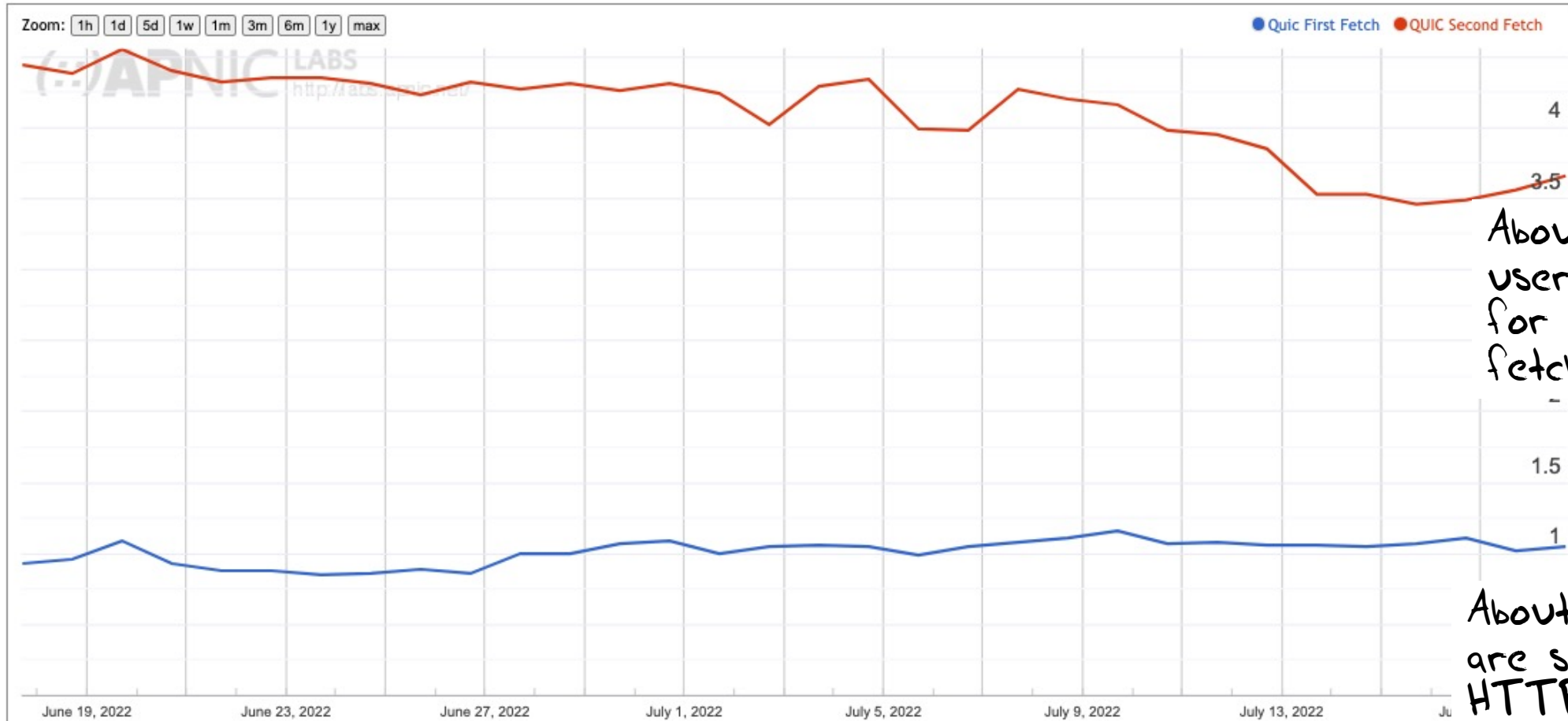
- Server:
 - nginx v1.21.7 with QUIC functions included
- DNS:
 - Set up an HTTPS record for each URL with value: **alpn="h3"**
- Content:
 - **Alt-Svc: h3=" :443"**

(This second method requires a subsequent query to allow the client to use the Alt-Svc capability. We perform a 2-second delayed second query for this URL in the measurement experiment approximately one fifth of the time. We keep the domain name constant and vary the URL args to detect the second fetch.)

Setting up QUIC

- Server:
 - nginx v1.21.7 with QUIC functions included
 - DNS:
 - Set up an HTTPS record for each URL with value: **alpn="h3"**
 - Content:
 - **Alt-Svc: h3=":443"**
- First Fetch
- Second Fetch

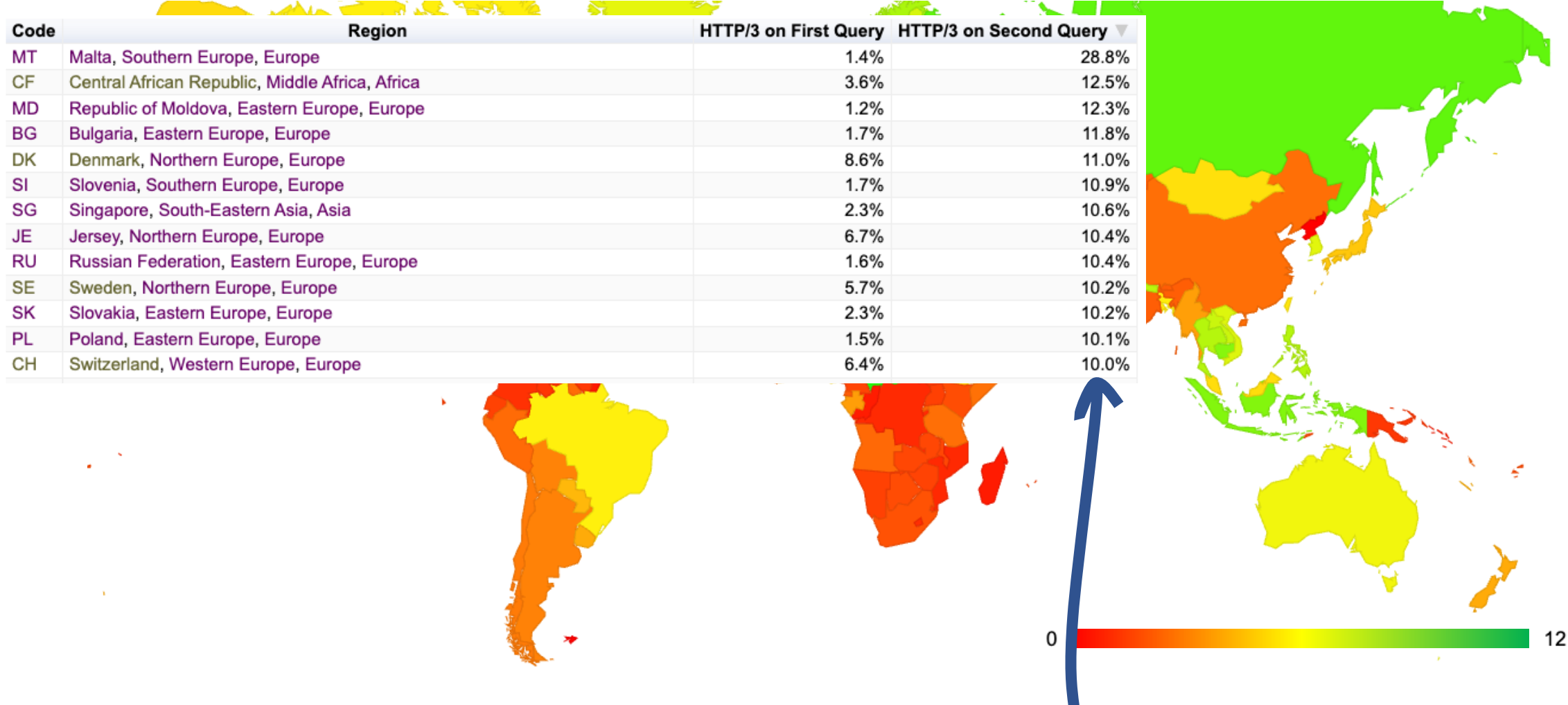
QUIC Use - June/July 2022



About 3.5÷ of users use HTTP/3 for the second fetch

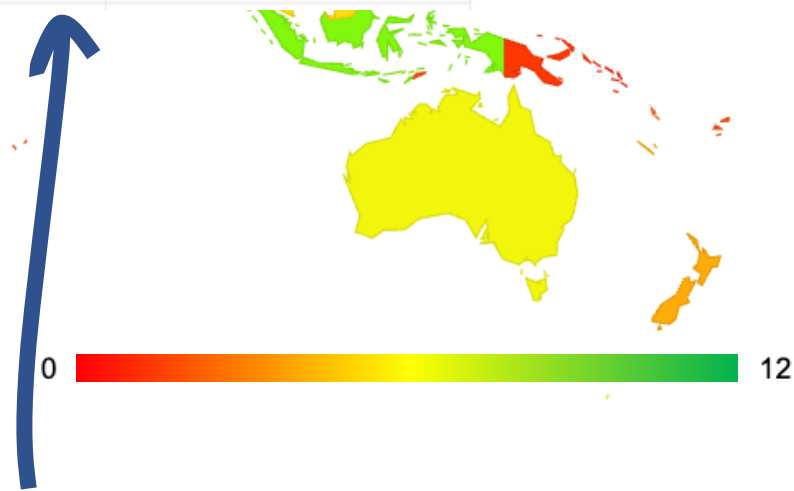
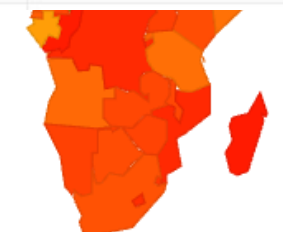
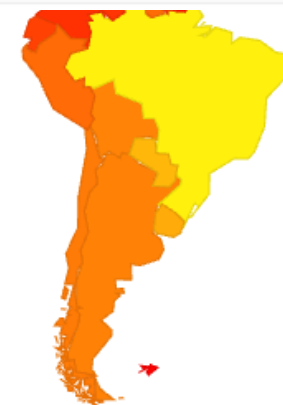
About 1÷ of users are seen to use HTTP/3 on first fetch

QUIC Use - June/July 2022



QUIC Use - June/July 2022

Code	Region	HTTP/3 on First Query	HTTP/3 on Second Query
DK	Denmark, Northern Europe, Europe	8.6%	11.0%
JE	Jersey, Northern Europe, Europe	6.7%	10.4%
CH	Switzerland, Western Europe, Europe	6.4%	10.0%
SE	Sweden, Northern Europe, Europe	5.7%	10.2%
FO	Faeroe Islands, Northern Europe, Europe	5.5%	7.9%
GI	Gibraltar, Southern Europe, Europe	5.2%	8.8%
MC	Monaco, Western Europe, Europe	5.2%	8.2%
GB	United Kingdom of Great Britain and Northern Ireland, Northern Europe, Europe	5.0%	9.1%
IM	Isle of Man, Northern Europe, Europe	4.8%	5.8%
MO	Macao Special Administrative Region of China, Eastern Asia, Asia	4.4%	6.1%
BH	Bahrain, Western Asia, Asia	4.0%	6.4%
GG	Guernsey, Northern Europe, Europe	3.9%	4.9%



Some Questions:

1. Which clients are performing QUIC and why?
2. What are the QUIC MSS values?
3. What is the QUIC connection failure rate?
4. Is QUIC faster than HTTP/2 + TLS?

1. OS Clients* performing QUIC

	TCP/TLS	QUIC on First Fetch	QUIC on Second Fetch
iOS	5.5%	↑ 93.3%	16.1%
Mac OS	1.0%	2.8%	0.6%
Android	84.5%	1.7%	↑ 77.9%
Win	5.5%	1.4%	4.3%
Linux	0.4%	0.2%	0.2%
Others	3.1%	0.6%	0.9%
	100.0%	100.0%	100.0%

* Based on reported browser string

1. Browser Clients* performing QUIC

	TCP/TLS	QUIC on First Fetch	QUIC on Second Fetch
Chrome	91.8%	4.1%	↑ 81.7%
Safari	4.3%	↑ 93.3%	↑ 16.1%
Firefox	0.8%	2.4%	1.0%
Edge	0.7%	0.0%	0.5%
Opera	0.2%	0.1%	0.6%
Others	2.2%	0.1%	0.1%
	100.0%	100.0%	100.0%

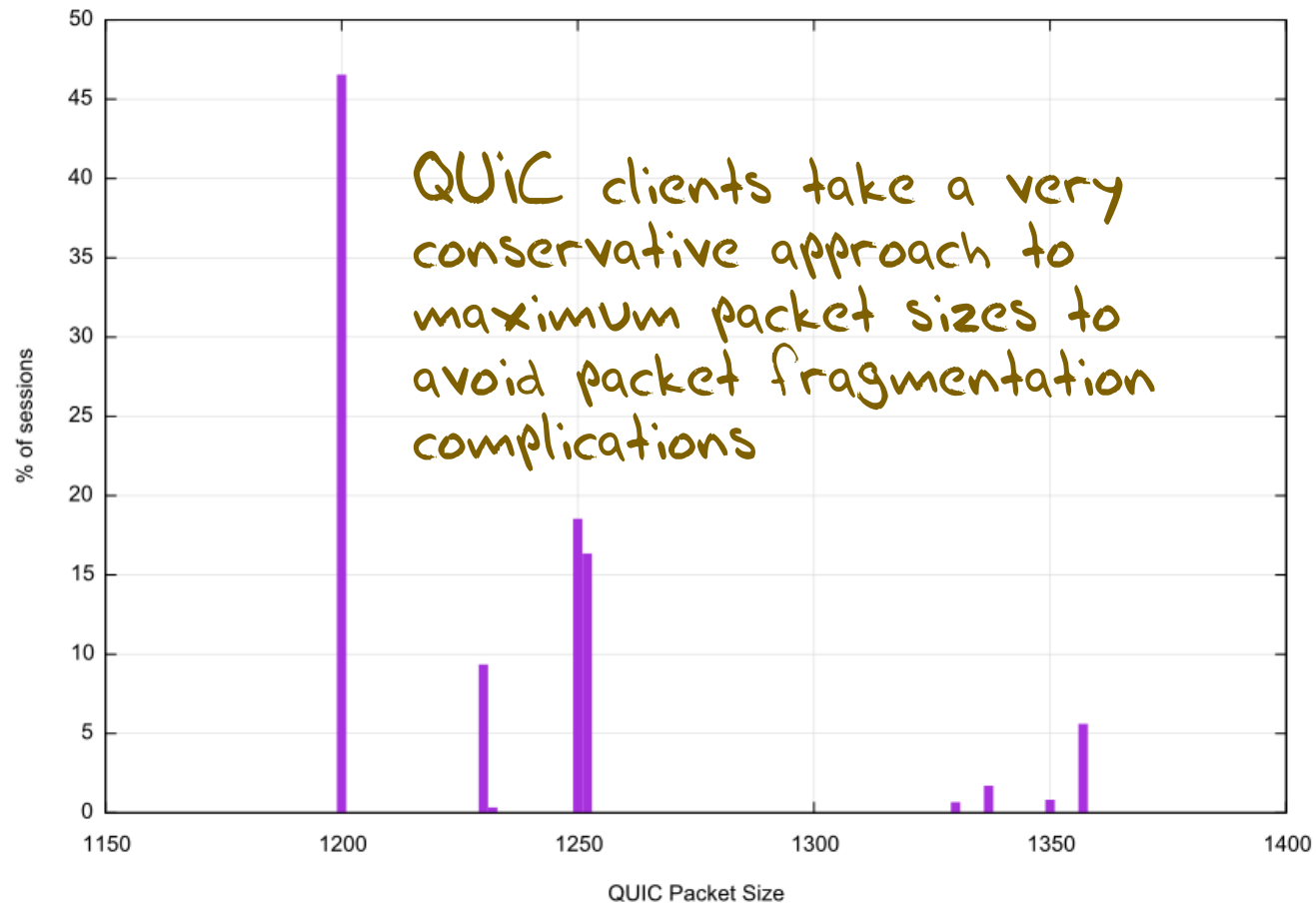
* Based on reported browser string

1. Who does QUIC and why?

Apple Safari clients use a **DNS HTTPS** query and *some* of these clients then follow up with a fetch over QUIC. The observed DNS HTTPS query to QUIC fetch conversion rate was relatively small.

Chrome clients use the **Alt-Svc** field as a QUIC trigger for *some* clients. Again, the observed QUIC conversion rate was observed to be relatively small

2. QUIC Packet Size distribution



Maximum Packet Sizes used in QUIC sessions:

1,200 octets – 46.6%

1,250 octets – 18.5%

1,252 octets – 16.4%

3. QUIC Connection Loss

In this measurement framework we cannot measure client -> server loss, but we can measure server-> client loss by looking for incomplete QUIC initial connections that do not complete

(this form of connection loss could be due to the client filtering incoming UDP port 443 packets)

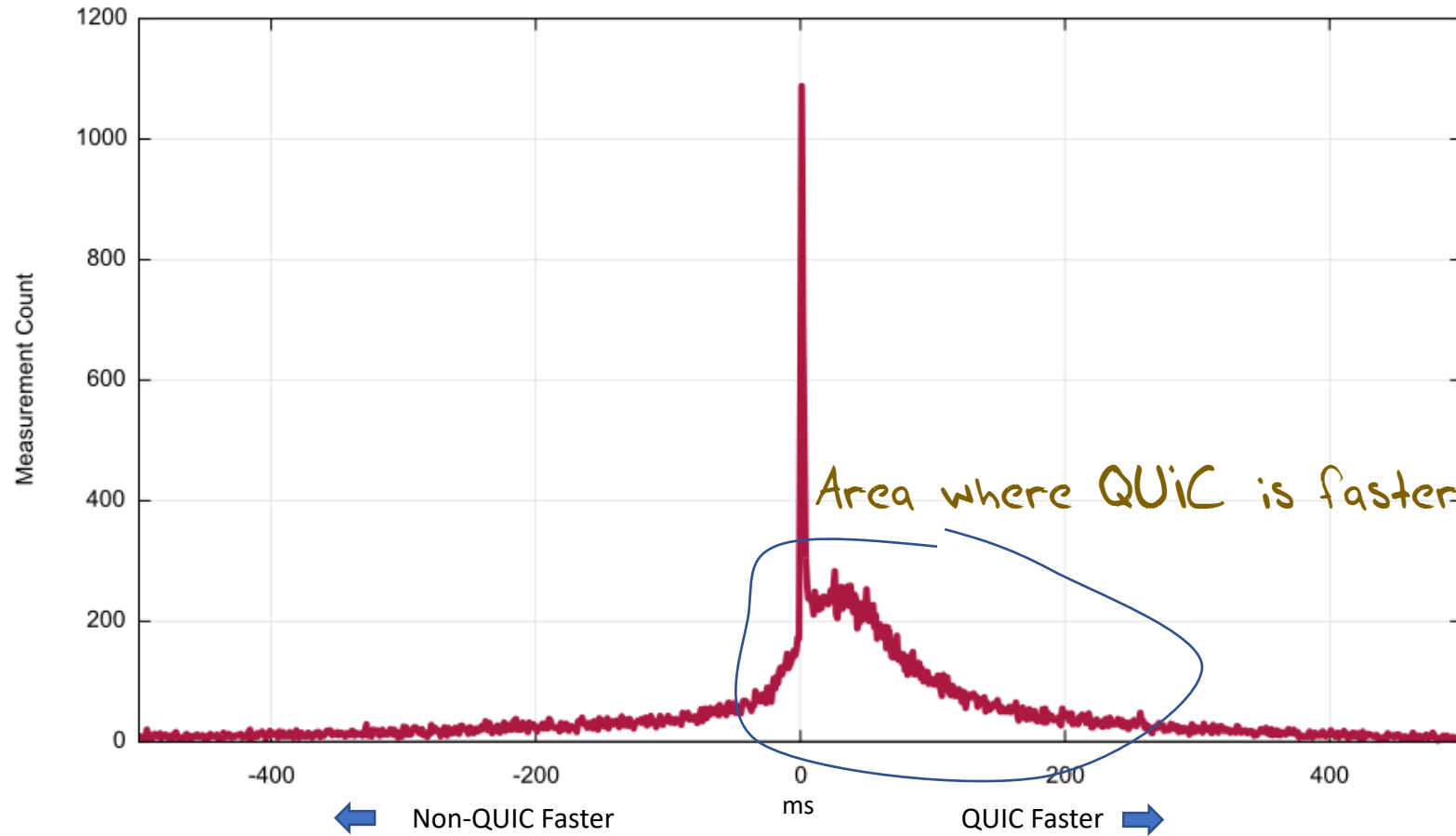
Initial QUIC Connections:	19,211,357
Failed Connections:	46,645
Failure Rate:	0.24%

4. Is QUIC Faster?

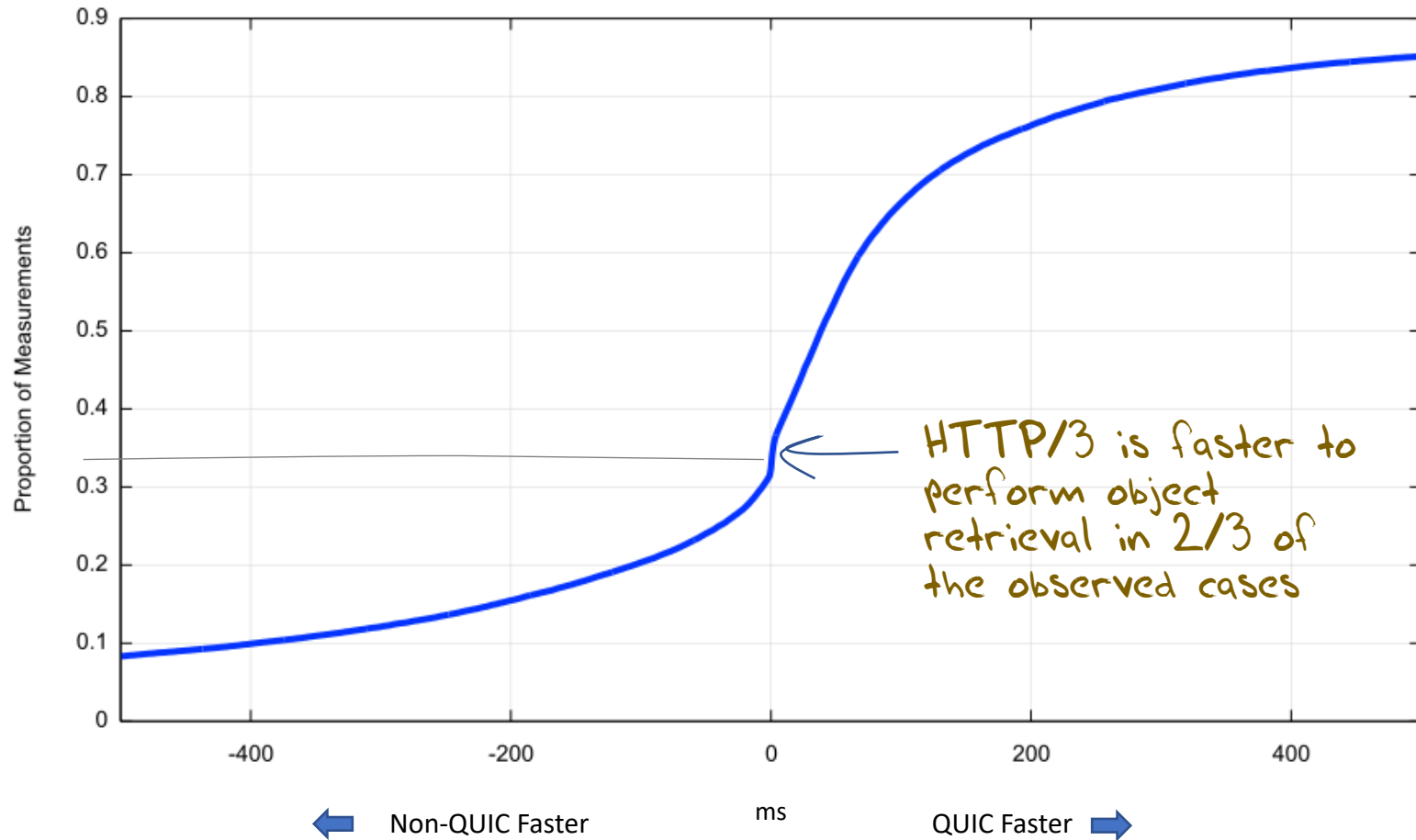
Let's compare the user-measured time to load an object using HTTP/2 and the same user's measured time to load the same object using HTTP/3

- There are a number of variables in the user time measurement, including varying time penalties relating to the internal task scheduling within the browser, but these individual factors should be cancelled out over a large enough sample set

4. TCP/TLS vs QUIC speed difference



4. Cumulative Distribution



Some Answers:

1. Which clients are performing QUIC?

The recent change appears to relate predominately to iOS 15.x clients (iPhones and iPads) using HTTPS queries and selectively performing an object retrieval over HTTP/3 at a rate of approximately 1 in 5

2. What are the QUIC MSS values?

Most QUIC clients limit their total IP packet size to a max of either 1,250 or 1,252 octets. Largest observed packet was 1,357 octets

3. What is the QUIC connection failure rate?

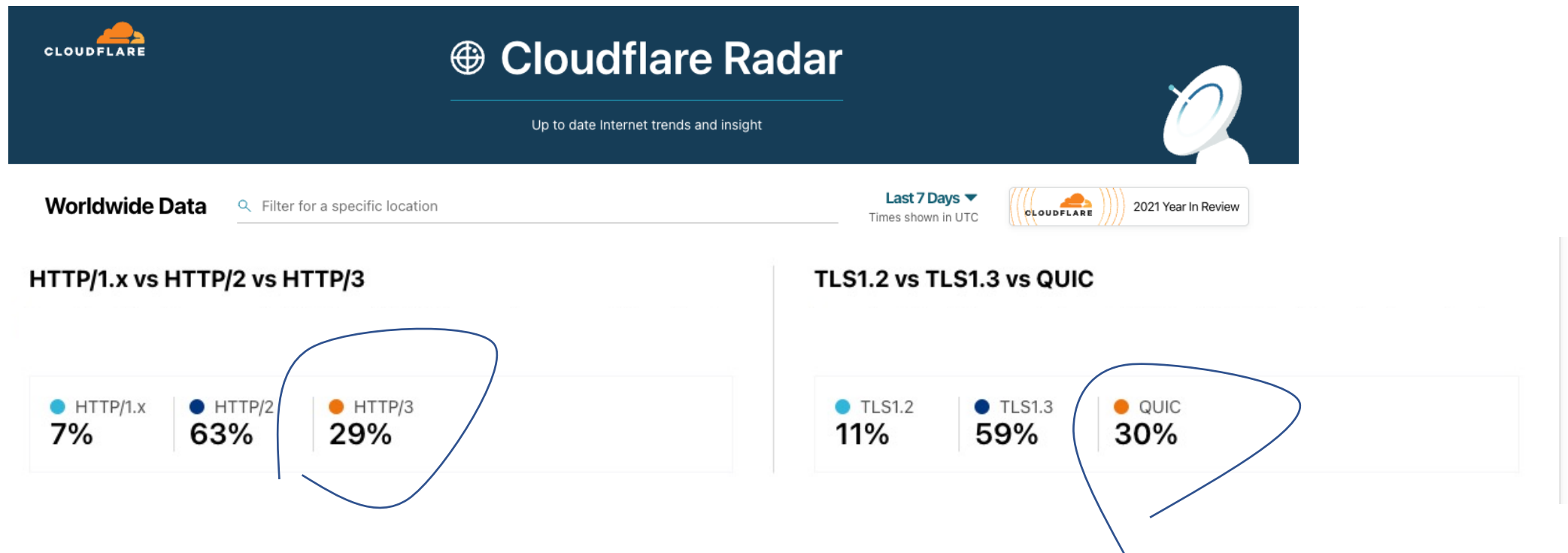
Extremely small at 0.24%. This falls within the bounds of experimental error in this experiment's framework.

4. Is QUIC faster than HTTP/2 + TLS?

Yes, more than 2/3 of the time QUIC will complete in less elapsed time than the equivalent HTTP/2 retrieval

Observations

These QUIC use numbers are **far lower** than other published measures



Why so low?

- QUIC is enabled by default in Chrome, so we might expect to see 90% of all connections using QUIC on the second and subsequent fetches.
- Cloudflare's numbers are based on fetches from their stores, and its unclear which are "first" fetches and which are "repeat" fetches from their summary data.
- The APNIC numbers are based on scripted behaviours of individual clients enrolled via online ads, and the Chrome behaviour is triggered by a scripted repeat fetch following a 2 second timer.

More Questions

- Why is the HTTPS DNS query rate 3x to 4x higher than the Safari QUIC fetch rate?
 - Surely a HTTP/3 aware browser would try using QUIC if the DNS response indicated that the service was HTTP/3 capable
- Is a 2 second scripted wait for Chrome too fast in practice?
 - How long does it take for content directives to be stashed in the browser to direct future actions?
- Does Chrome have the same defaults everywhere? Or are there localized settings in India and China (for example) that disable QUIC by default?
- Does Safari have the same defaults everywhere?

Thanks!

**Ongoing HTTP/3 Measurement Report at APNIC Labs:
<https://stats.labs.apnic.net/QUIC>**