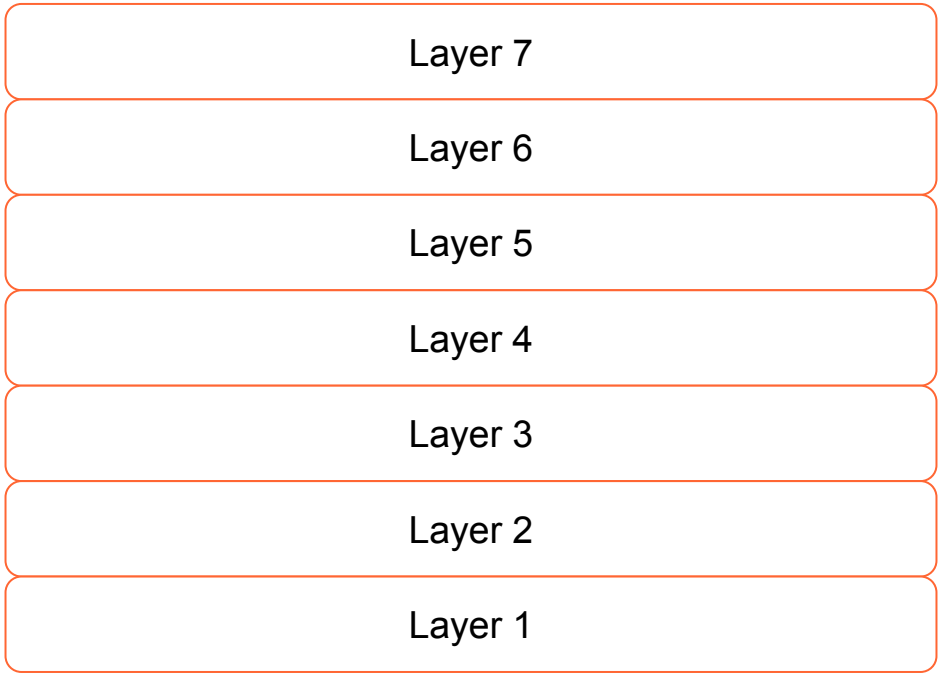
CLOUDFLARE

Layer 4¾ Fantastic quirks and where to find them

ANRW @ IETF 114, July 27 2022

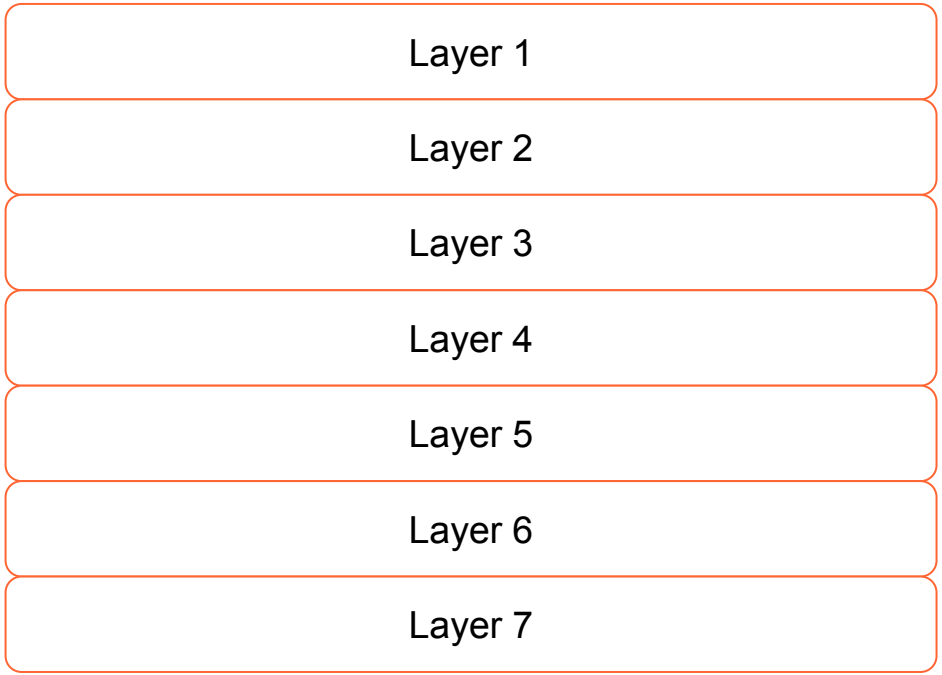
Lucas Pardue
Senior Engineer, Protocols Team, Cloudflare
Co-chair, IETF QUIC Working Group


CLOUDFLARE
9¾

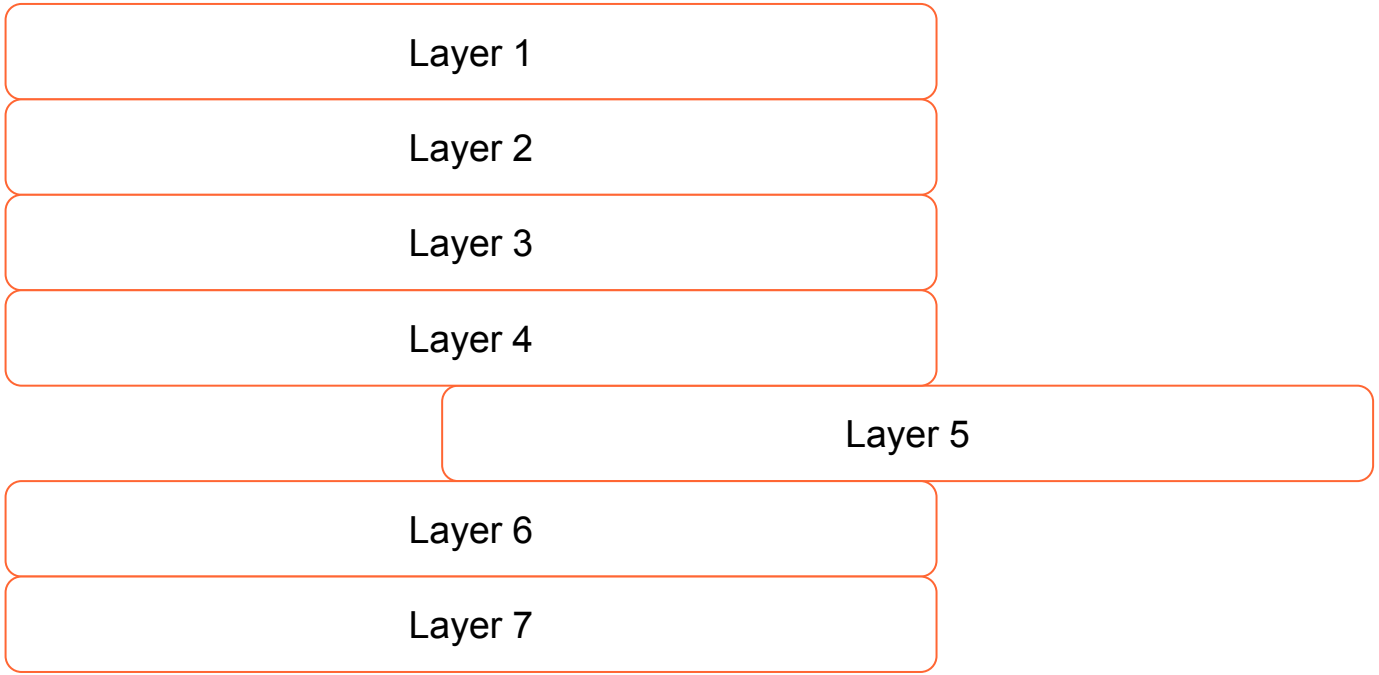

Layer 7
Layer 6
Layer 5
Layer 4
Layer 3
Layer 2
Layer 1

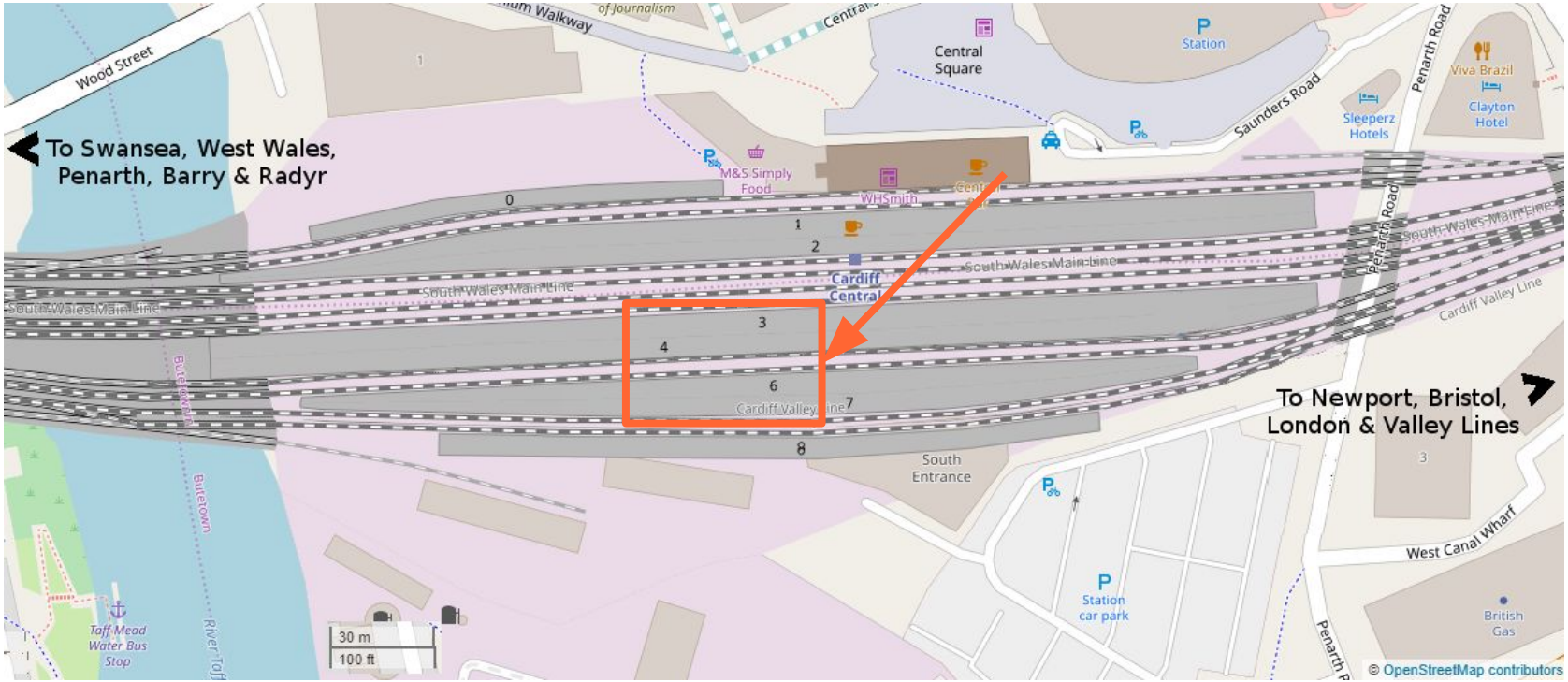
CLOUDFLARE

The Cake is a Lie


Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7

https://upload.wikimedia.org/wikipedia/commons/9/99/Cardiff_Central_plan.png
CC-BY-SA-2.0

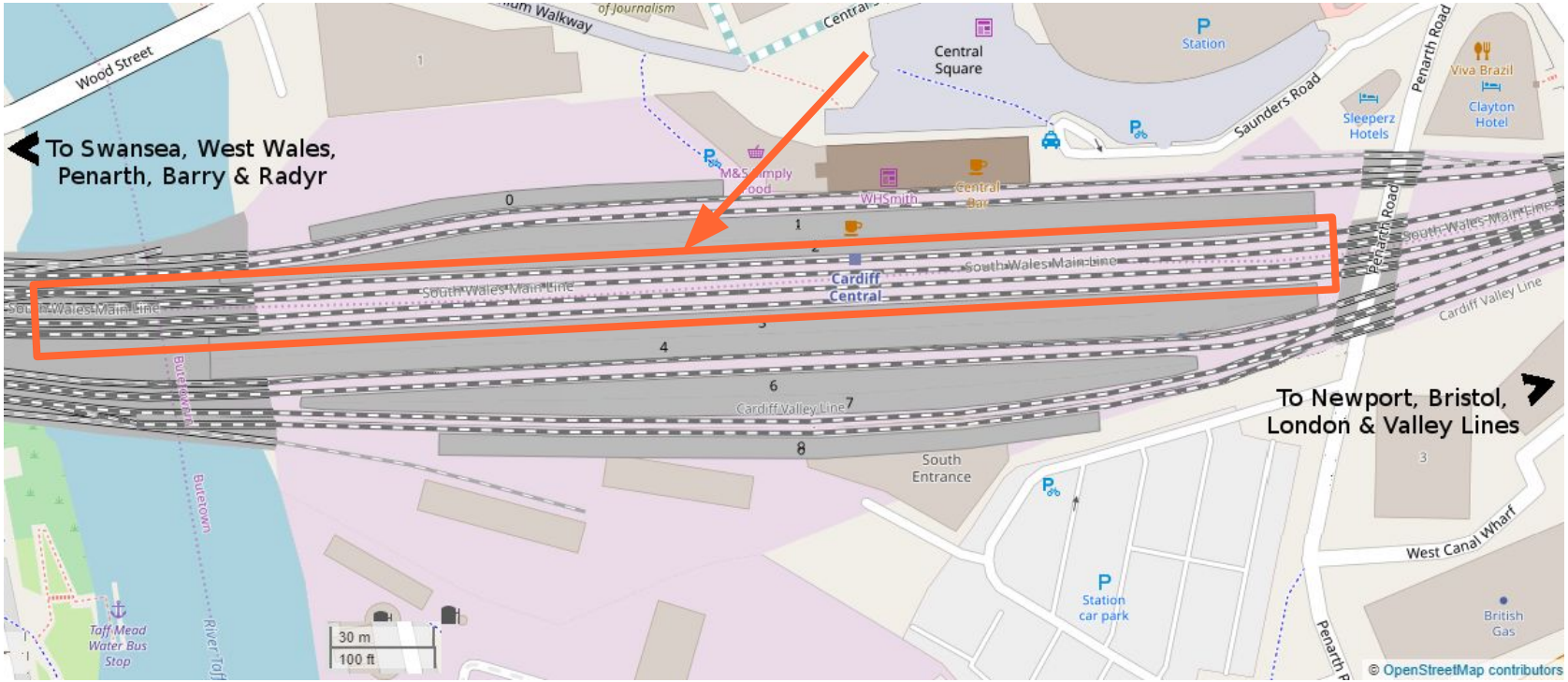

Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7



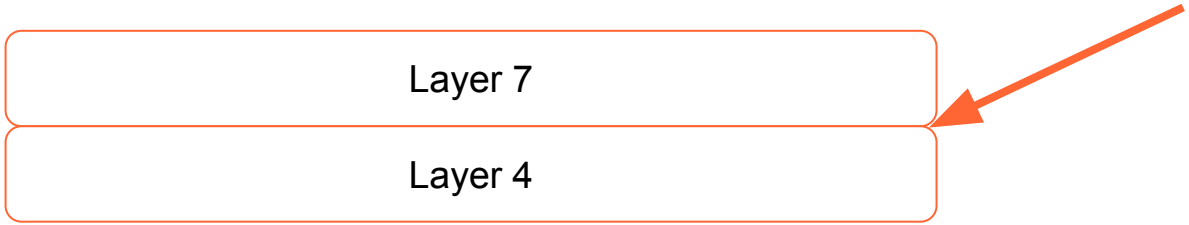
https://upload.wikimedia.org/wikipedia/commons/9/99/Cardiff_Central_plan.png
CC-BY-SA-2.0

https://commons.wikimedia.org/wiki/File:Cardiff_Central_platform_0_-_01.jpg
CC-BY-SA-2.0

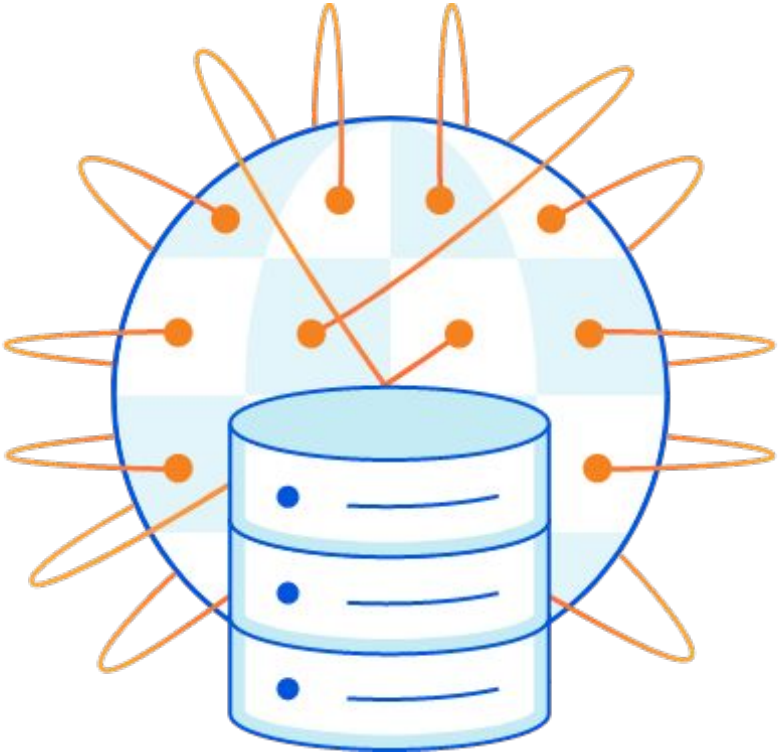
https://upload.wikimedia.org/wikipedia/commons/9/99/Cardiff_Central_plan.png
CC-BY-SA-2.0



https://upload.wikimedia.org/wikipedia/commons/9/99/Cardiff_Central_plan.png
CC-BY-SA-2.0


Layer 7
Layer 4

CLOUDFLARE



- QUIC
- HTTP
- HTTP/1.x
- HTTP/2
- HTTP/3
- TLS
- WebSocket
- gRPC

CLOUDFLARE

Spectrum

- Overview
- Protocols per plan
- About
- Get started
- How to
- Reference

Protocols per plan

On this table, you have information about which protocols are available per plan.

	Free	Pro	Business	Enterprise
TCP				✔
UDP				✔
Minecraft*		✔	✔	✔
SSH		✔	✔	✔
RDP			✔	✔

*Minecraft Java Edition is supported but Minecraft Bedrock Edition is not supported.

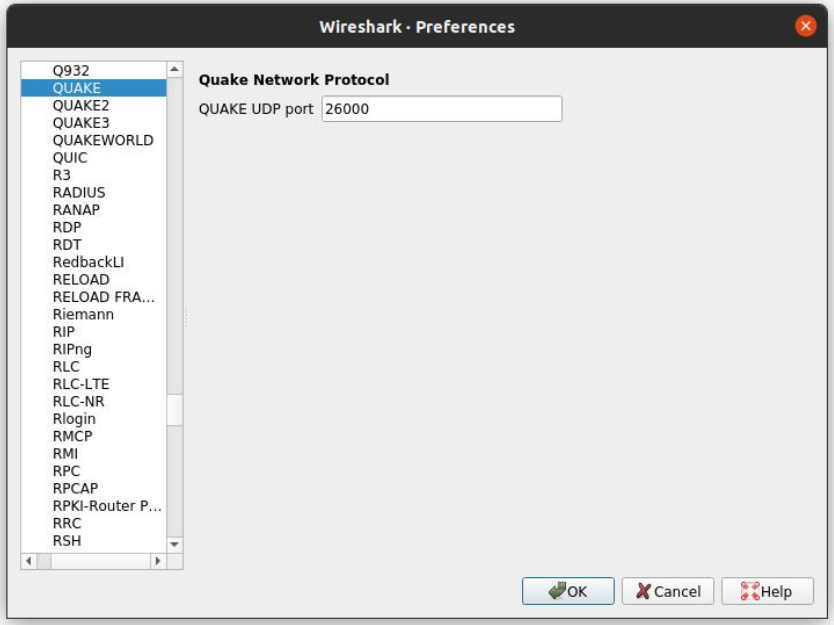
https://developers.cloudflare.com/spectrum/protocols-per-plan/
(2022/04/27)

CLOUDFLARE

- *"Minecraft Java Edition is supported but Minecraft Bedrock Edition is not supported"*

- https://minecraft.fandom.com/wiki/Protocol_version
- https://wiki.vg/Protocol_version_numbers
- https://wiki.vg/Bedrock_Protocol

- https://github.com/aresrpg/minecraft-dissector



CLOUDFLARE

HTTP means Hypertext Transport Protocol?

CLOUDFLARE

Mark Nottingham at IETF 99:

"Roy Fielding knows whenever you call it Hypertext Transport Protocol."

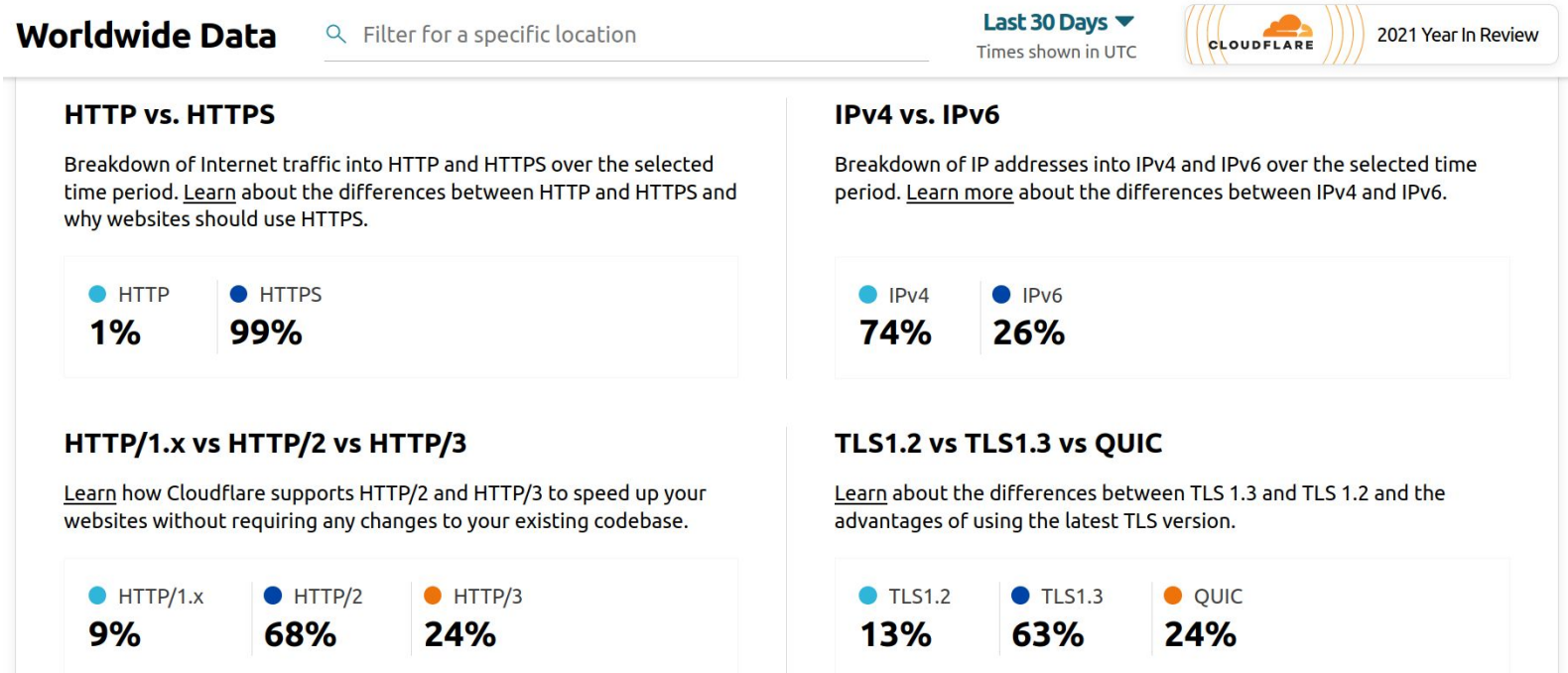
https://httpwg.org/wg-materials/ietf99/minutes.html

CLOUDFLARE

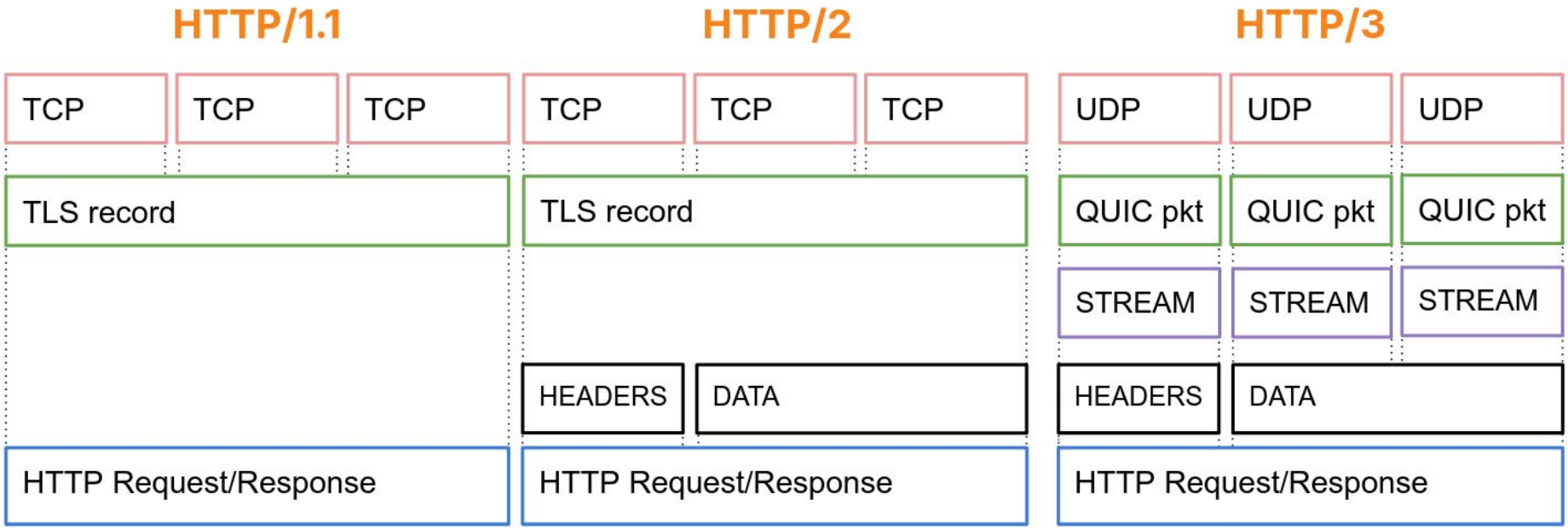
The view from our edge

CLOUDFLARE

- radar.cloudflare.com



https://radar.cloudflare.com
Last 30 days of traffic, excluding bots, on 2022/04/27



https://blog.cloudflare.com/a-primer-on-proxies/

CLOUDFLARE

Worldwide Data

Filter for a specific location

Last 30 Days
Times shown in UTC

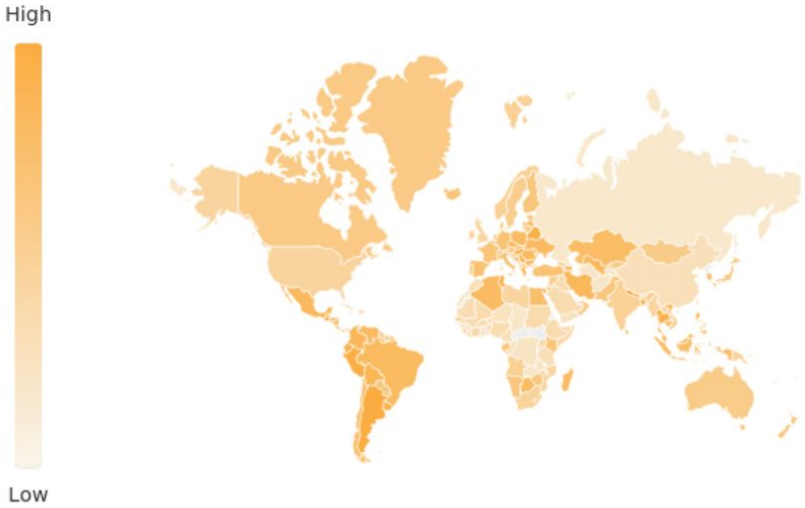
CLOUDFLARE 2021 Year In Review

Trends

Learn More

HTTP/3 (QUIC)

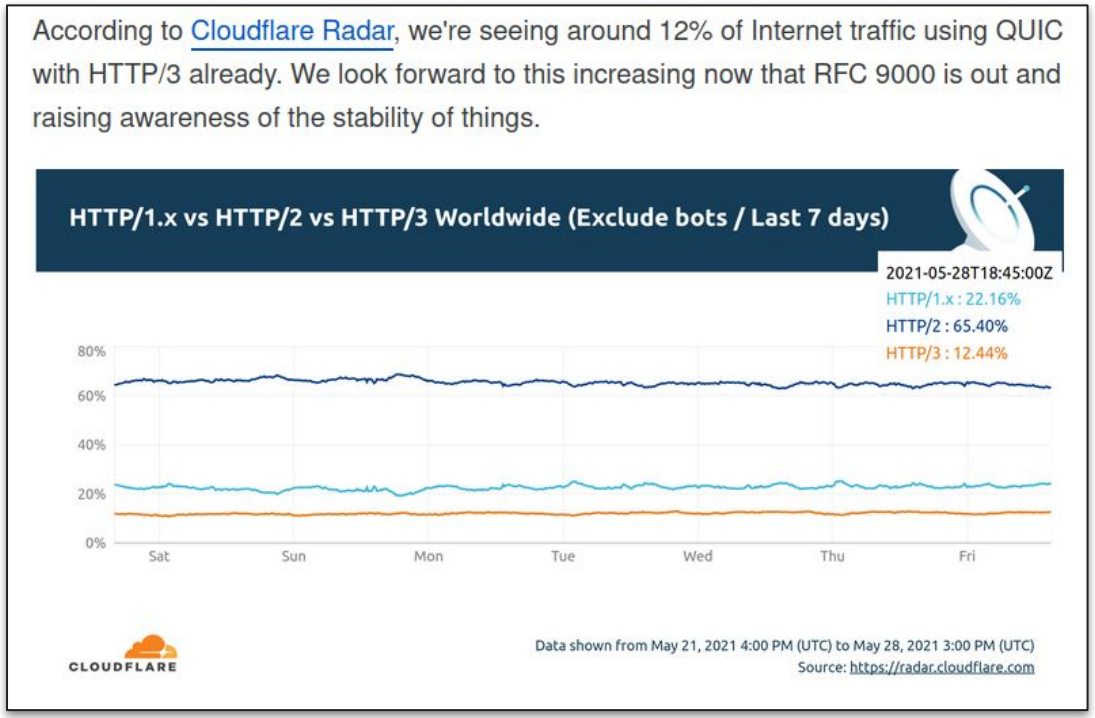
HTTP/3 (QUIC) traffic, in the selected time period, by country. Learn more about the advantages of HTTP/3 (QUIC).



	Country	Change	%
1.	Argentina	+1.60%	37.44%
2.	Peru	+1.86%	36.55%
3.	Nepal	+6.38%	36.24%
4.	North Macedonia	+1.20%	36.01%
5.	Belarus	+1.50%	35.76%
6.	Moldova	+0.54%	35.61%
7.	Serbia	+0.80%	35.50%
8.	Tunisia	-1.58%	35.42%
9.	Ecuador	+1.44%	35.19%
10.	Guatemala	+1.34%	34.79%

https://radar.cloudflare.com
Last 30 days of traffic, on 2022/04/27

- https://blog.cloudflare.com/quic-version-1-is-live-on-cloudflare/
  - Posted 2021/28/05



CLOUDFLARE

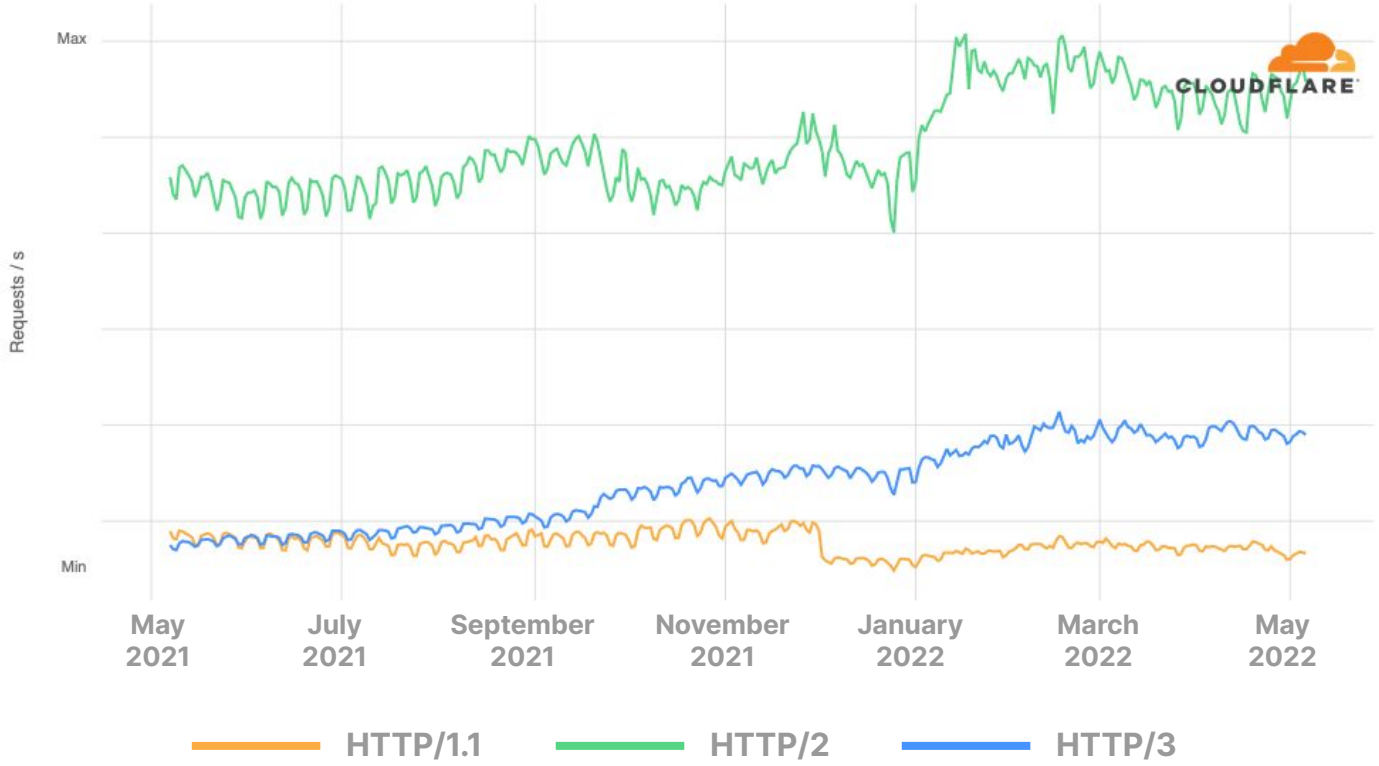
- https://blog.cloudflare.com/cloudflare-view-http3-usage/

HTTP RFCs have evolved: A Cloudflare view of HTTP usage trends

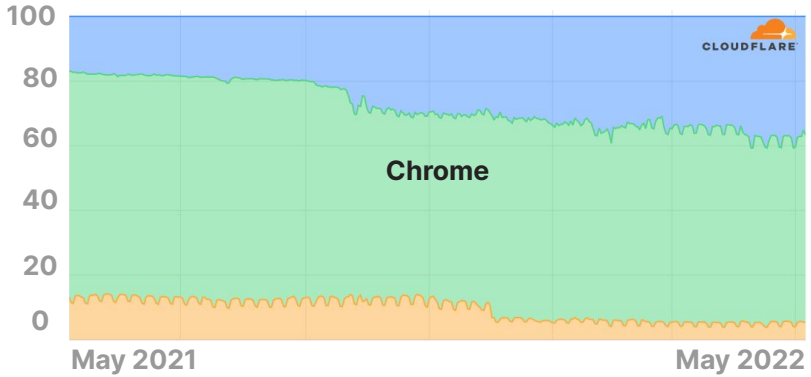
06/06/2022

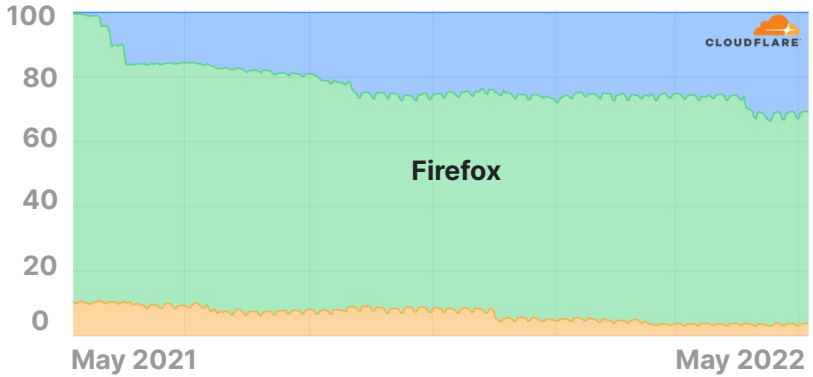
 Lucas Pardue  David Belson

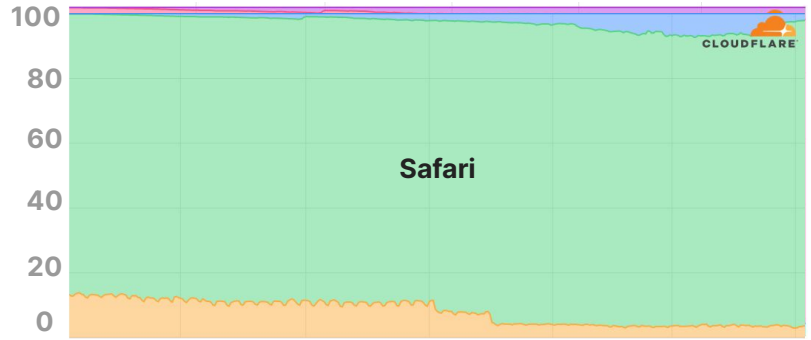
Long-term trends - Global HTTP requests by version, all browsers

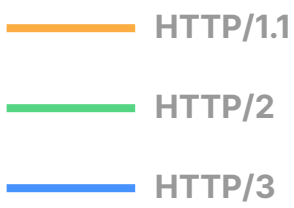


Long-term trends - Global HTTP requests by version, stacked %

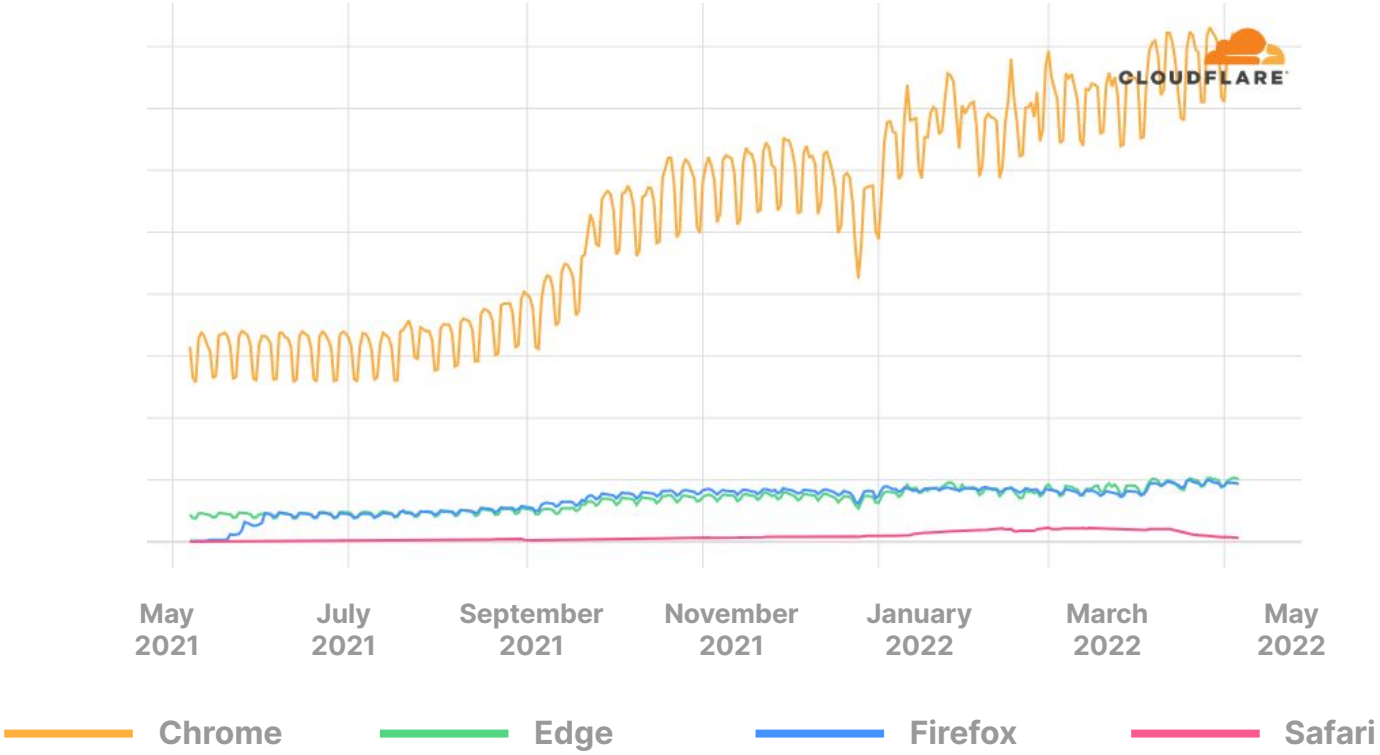




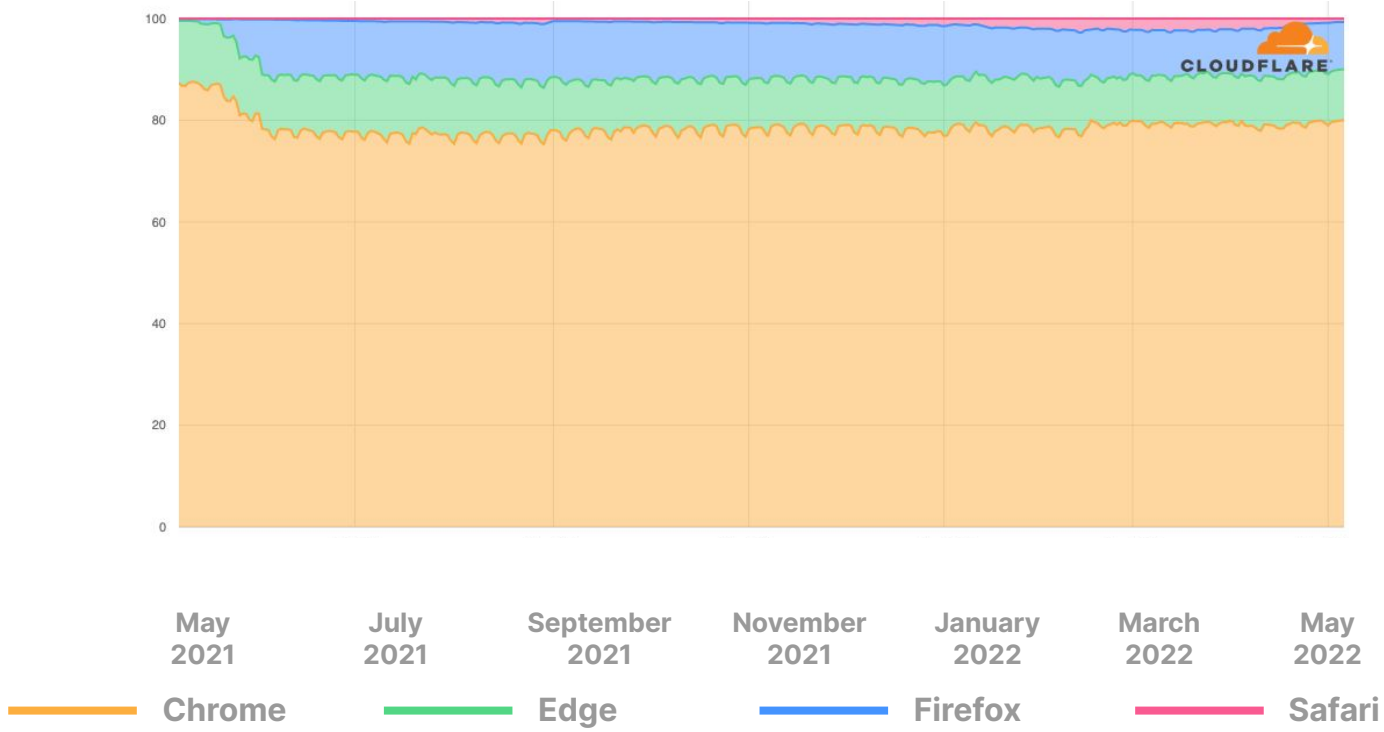




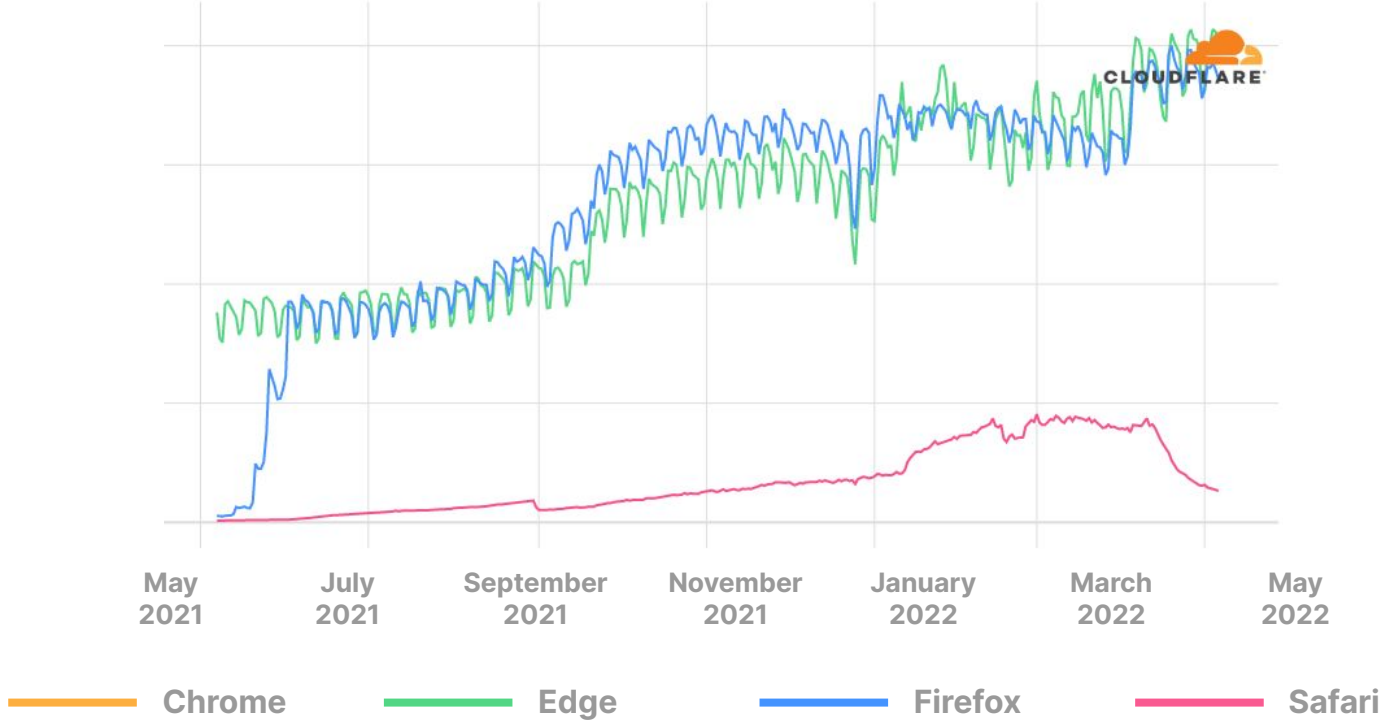
Long-term trends - Global HTTP/3 requests by browsers



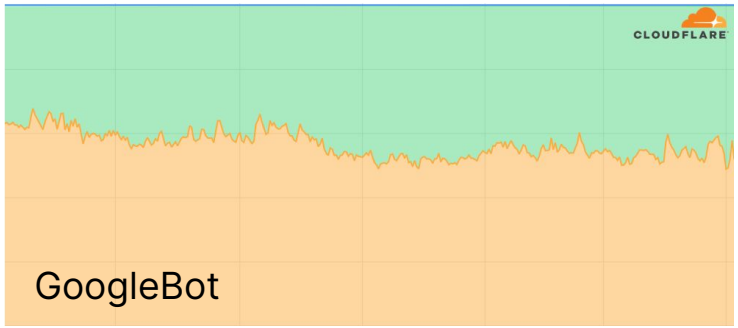
Long-term trends - Global HTTP/3 requests by browsers, stacked %



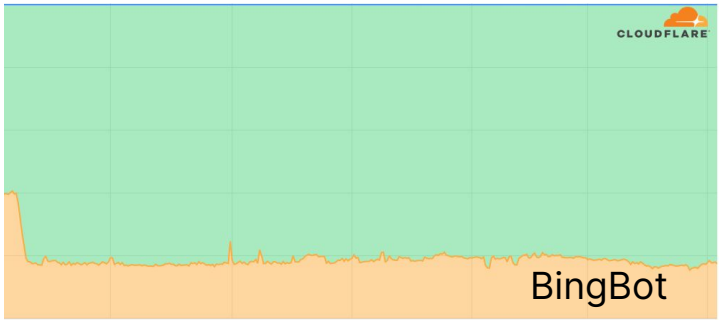
Long-term trends - Global HTTP/3 requests by browsers (w/o Chrome)



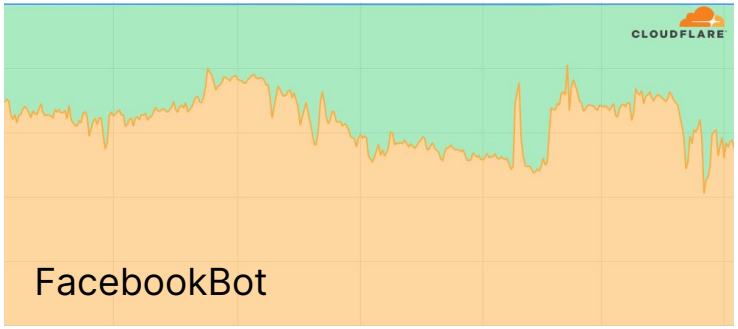
Long-term trends - Global HTTP requests by version, bots

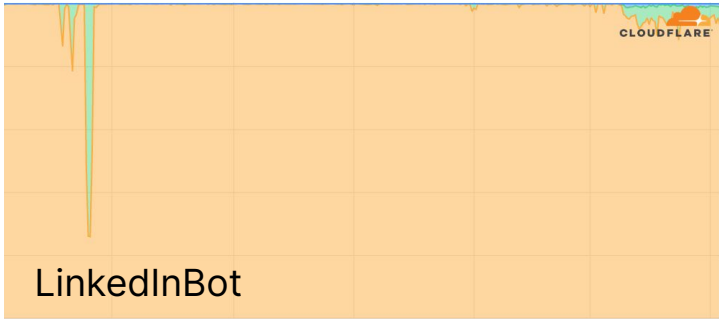


May 2021 May 2022



May 2021 May 2022





HTTP/1.1 HTTP/2 HTTP/3

Binary framing, the solution to all our HTTP problems?

CLOUDFLARE

- Text-based HTTP has pedigree in providing quirks
- Recent examples include "request smuggling"
  - https://portswigger.net/web-security/request-smuggling
- Despite the baggage, HTTP/1 is quite well exercised and understood by technically-minded folks
- Yet, HTTP/2 and HTTP/3 constitute the majority of Web traffic
- How do we characterize their quirks and performance?

CLOUDFLARE

- h2spec - https://github.com/summerwind/h2spec
- h3spec - https://github.com/kazu-yamamoto/h3spec
- h2load - https://nghttp2.org/documentation/h2load-howto.html
- netlog
- qlog?
- *<insert your favourite>*
- Useful tools, but in general our coverage over HTTP/2 or HTTP/3, as an Internet community, seems lacking

CLOUDFLARE

ERR_SPDY_PROTOCOL_ERROR

CLOUDFLARE

ERR_SPDY_PROTOCOL_ERROR

"Flush the SPDY pockets"

https://kinsta.com/knowledgebase/err_spdy_protocol_error/#method-4-flush-the-spdy-pockets

CLOUDFLARE

ERR_HTTP2_PROTOCOL_ERROR

CLOUDFLARE

ERR_HTTP2_PROTOCOL_ERROR

"All I had to do was turn my antivirus off and then on again haha"

https://support.google.com/chrome/thread/117505176/err-http2-protocol-error-please-help-all-browsers-including-chrome-will-not-work?hl=en

CLOUDFLARE

ERR_QUIC_PROTOCOL_ERROR

CLOUDFLARE

ERR_QUIC_PROTOCOL_ERROR

"If you're still stuck after trying all these methods, contact Google customer support for help."

https://www.lifewire.com/how-to-fix-chrome-err-quic-protocol-error-4686703

CLOUDFLARE

Issue 1121658: QUIC & HTTP/3 Network Error Logging Granularity Parity

Reported by lucas...@gmail.com on Tue, Aug 25, 2020, 6:38 PM GMT+1

UserAgent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:79.0) Gecko/20100101 Firefox/79.0

Steps to reproduce the problem:
1. Open the source code at https://chromium.googlesource.com/chromium/src/+/refs/tags/87.0.4244.2/net/network_error_logging/network_error_logging_service.cc#97
2. See there is one single entry for ERR_QUIC_PROTOCOL_ERROR that maps to h3.protocol.error
3. Maybe try and generate some QUIC or HTTP/3 errors and be flumuxed in your debugging using Network Error Logging because the level of detail is not sufficient

What is the expected behavior?
It would be nice for {QUIC & HTTP/3} NEL error types to have parity with {TCP, TLS & HTTP/2}. TLS errors are probably common, so this is really about surveying the TCP and HTTP/2 error codes and seeing what maps and devs would find useful.
The list is

Comment 4 by b...@chromium.org on Wed, Aug 26, 2020, 7:29 PM GMT+1 Project Member

If I understand correctly, this is a fairly large project. It's not that Network Error Logging errors are not granular enough, but in fact the net errors are not granular enough. One would need to add a couple dozen new QUIC error codes to net/base/net_error_list.h (and a couple dozen more HTTP/2 error codes would also be very helpful), and emit them in the right places of the network stack.

This is definitely a worthwhile task, because these error codes are user visible, and would therefore give our users a lot more information.

I'll see what I can do in the near future.

Comment 5 by lucas...@gmail.com on Wed, Aug 26, 2020, 8:00 PM GMT+1

Improving HTTP/2 granularity would be greatly appreciated too.

https://bugs.chromium.org/p/chromium/issues/detail?id=1121658

CLOUDFLARE

"Lucas's Malformed"

- https://quicwg.org/base-drafts/draft-ietf-quic-http.html#section-4.1.3
  - ...
    the absence of mandatory pseudo-header fields,
    ...
- Every request needs a method:
  - *Malformed requests or responses that are detected **MUST** be treated as a stream error (Section 8) of type H3_MESSAGE_ERROR.* (0×010E)
  - *For malformed requests, a server **MAY** send an HTTP response indicating the error prior to closing or resetting the stream.*

CLOUDFLARE

"Lucas's Malformed" - HTTP/3 server reaction to missing :method

Server host	Result
lucaspardue.com	400 Bad Request
google.com	400 Bad Request
facebook.com	400 Bad Request
ietf.akaquic.com	400 Bad Request
test.privateoctopus.com:4433	405 Method Not Allowed
mew.org	RESET_STREAM, 0×0102
msquic.net	RESET_STREAM, 0×0102
interop.seemann.io	RESET_STREAM, 0×0101
quic.aiortc.org	App CONNECTION_CLOSE, 0×010E
nghttp2.org	App CONNECTION_CLOSE, 0×010E

https://blog.cloudflare.com/on-the-recent-http-2-dos-attacks/

https://github.com/Netflix/security-bulletins/blob/master/advisories/third-party/2019-002.md

CLOUDFLARE

- CVE-2019-9512
  - *Some HTTP/2 implementations are vulnerable to ping floods, potentially leading to a denial of service. The attacker sends continual pings to an HTTP/2 peer, causing the peer to build an internal queue of responses. Depending on how efficiently this data is queued, this can consume excess CPU, memory, or both.*

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9512

CLOUDFLARE

- gRPC
  - *The idea is simple and powerful: every time a receiver gets a data frame it sends out a BDP ping (a ping with unique data only used by BDP estimator). After this, the receiver starts counting the number of bytes it receives (including the ones that triggered the BDP ping) until it receives the ack for that ping.*

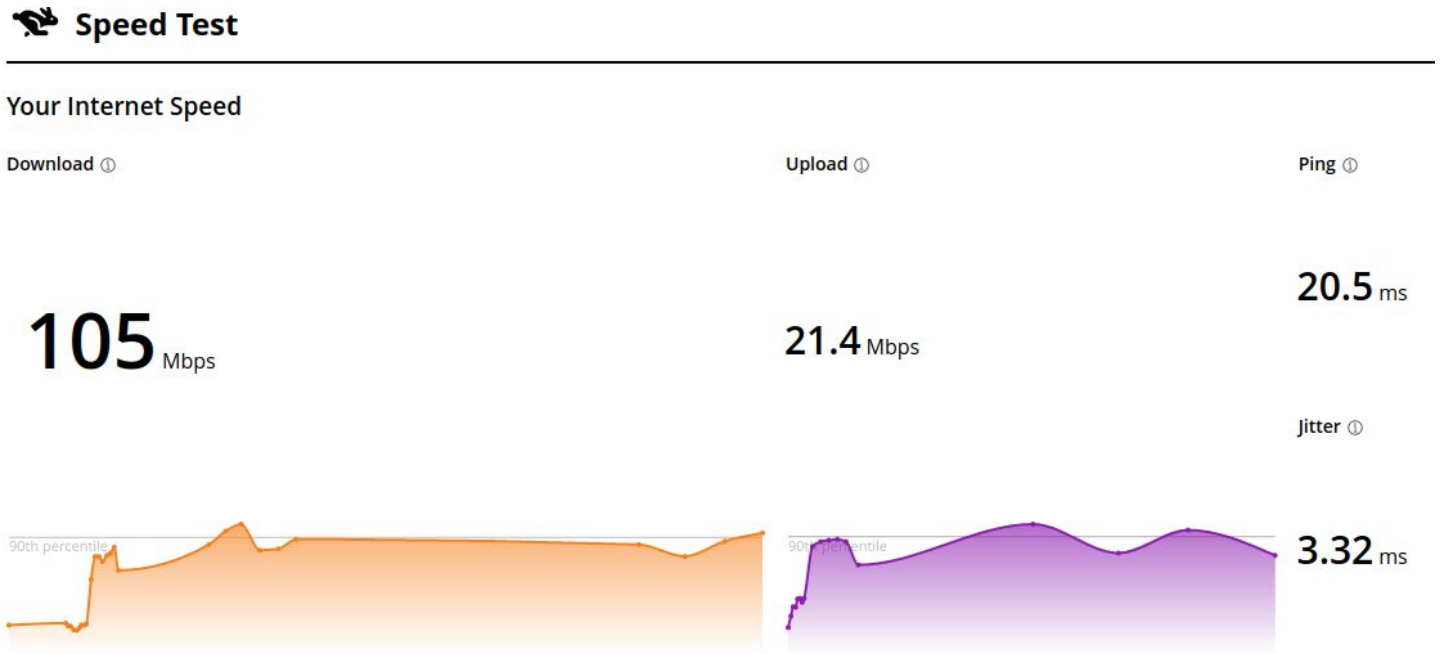
https://grpc.io/blog/grpc-go-perf-improvements/#bdp-estimation-and-dynamic-flow-control-window

CLOUDFLARE

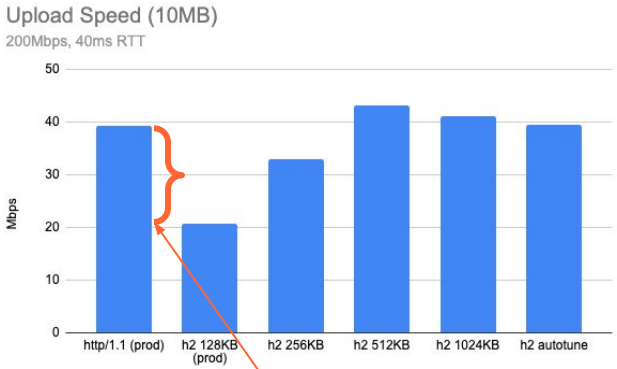
- Rust Hyper
  - Problem
    - HTTP/2 Adaptive Window sometimes triggers ENHANCE_YOUR_CALM
  - Fix
    - *This introduces a delay to sending a ping to calculate the BDP that becomes shorter as the BDP is changing, to improve throughput quickly, but then also becomes longer as the BDP stabilizes, to reduce the amount of pings sent.*

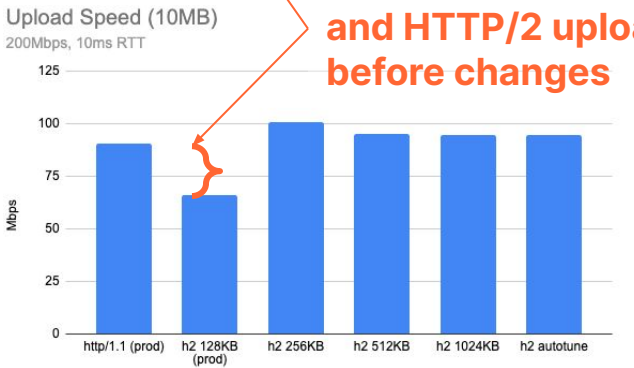
https://github.com/hyperium/hyper/issues/2526
https://github.com/hyperium/hyper/pull/2550

CLOUDFLARE

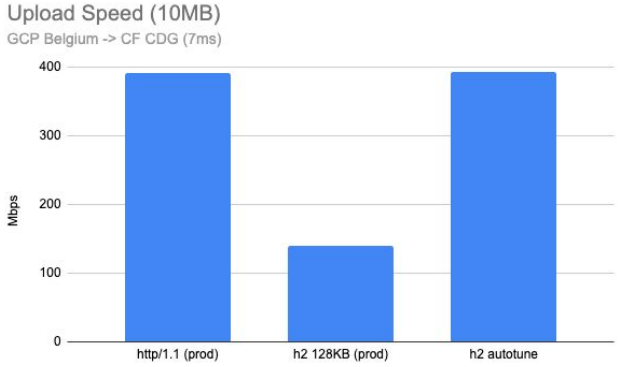


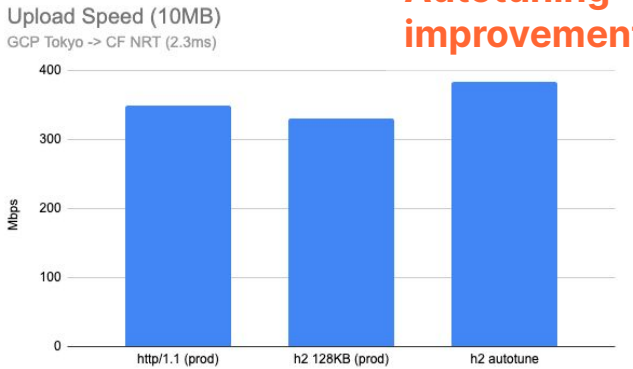
https://speed.cloudflare.com





Difference between HTTP/1.1 and HTTP/2 upload throughput before changes





Autotuning improvement

https://blog.cloudflare.com/delivering-http-2-upload-speed-improvements/

CLOUDFLARE

What do we measure? Layer 4, layer 7, the whole thing?

- Layers below HTTP will affect its performance
- But they aren't indicative of upper layer success
  - See our IAB Measuring Network Quality submission [1]
- Latency and jitter are fundamental problems
  - See our FCC submission about Broadband Nutrition [2]
- draft-ietf-ippm-responsiveness
  - Responsiveness under Working Conditions
  - Testing the "whole stack", including HTTP/2

https://www.iab.org/activities/workshops/network-quality/

[1] https://www.iab.org/wp-content/IAB-uploads/2021/09/Lower-layer-performance-is-not-indicative-of-upper-layer-success-20210906-00-1.pdf

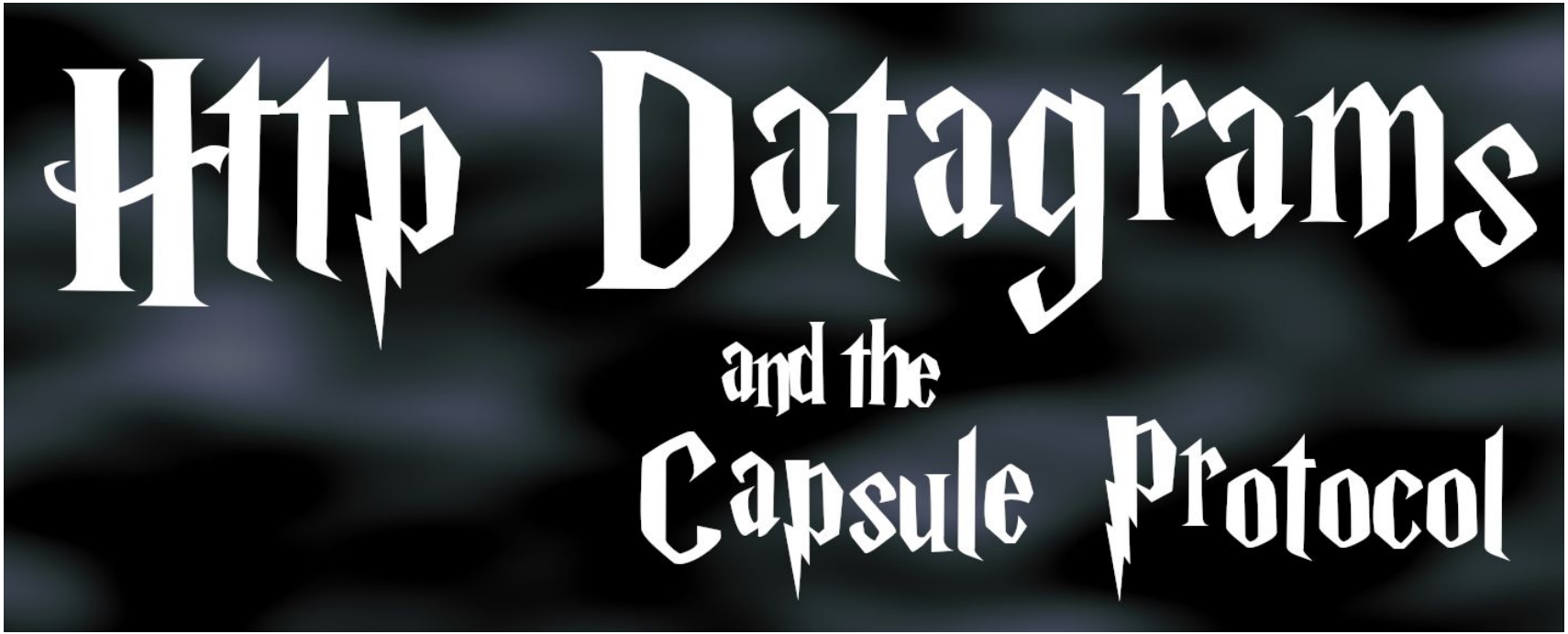
[2] https://blog.cloudflare.com/breaking-down-broadband-nutrition-labels/

CLOUDFLARE

Professor Snape in The Order of the Phoenix:

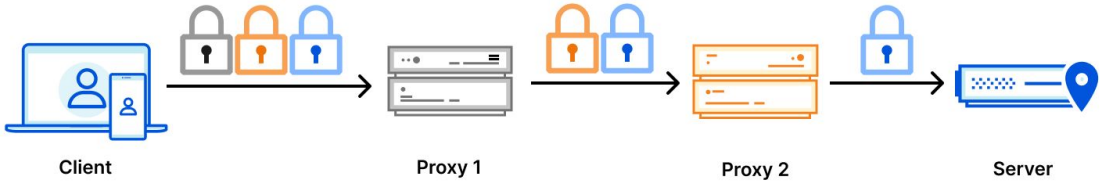
“Well, it may have escaped your notice, but life isn’t fair.”

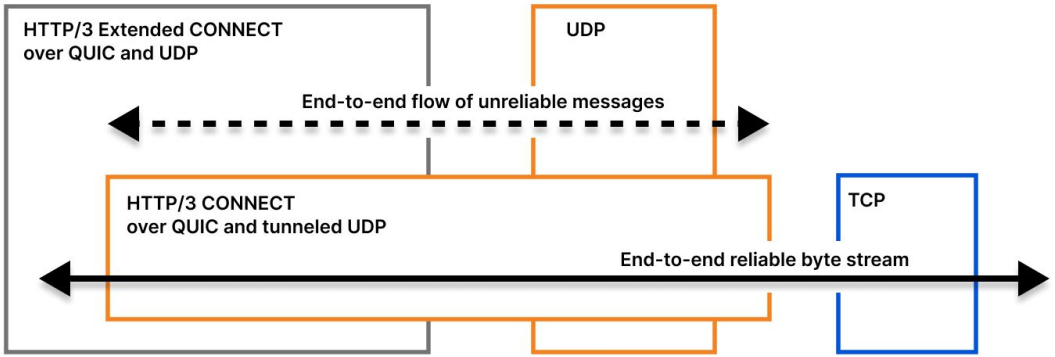
CLOUDFLARE



CLOUDFLARE

- UDP 443 All the things
  - QUIC DATAGRAM frames
  - HTTP Datagrams
  - Capsule Protocol
  - Proxying UDP over HTTP aka MASQUE
  - WebTransport
  - Media over QUIC (MoQ)
  - ...

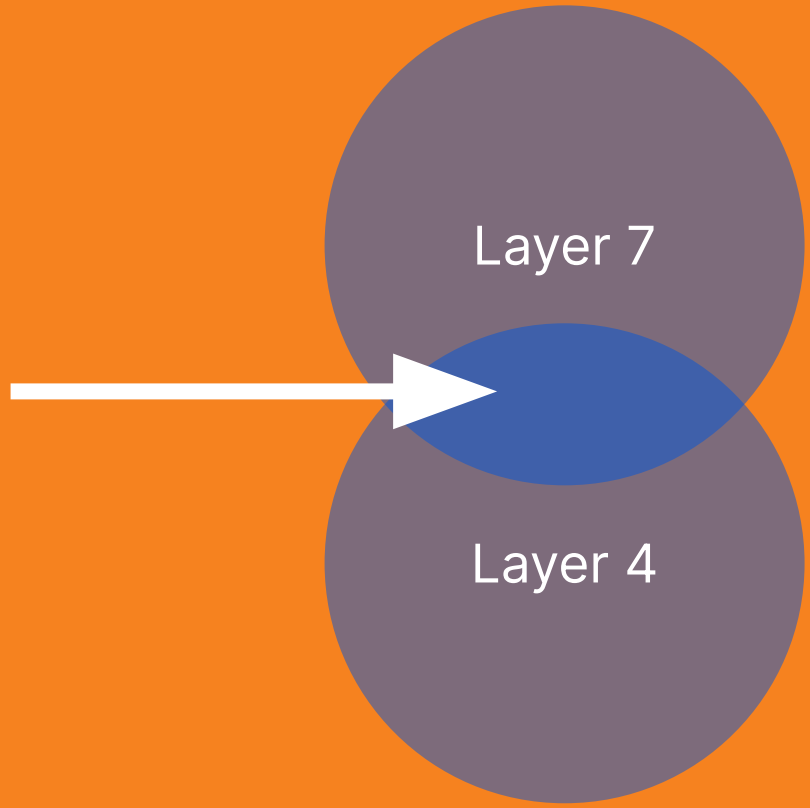




https://blog.cloudflare.com/unlocking-quic-proxying-potential/

CLOUDFLARE

4¾



CLOUDFLARE

Thank you

Lucas Pardue
Senior Engineer, Protocols Team, Cloudflare
Co-chair, IETF QUIC Working Group
@SimmerVigor