A QUICker Internet?

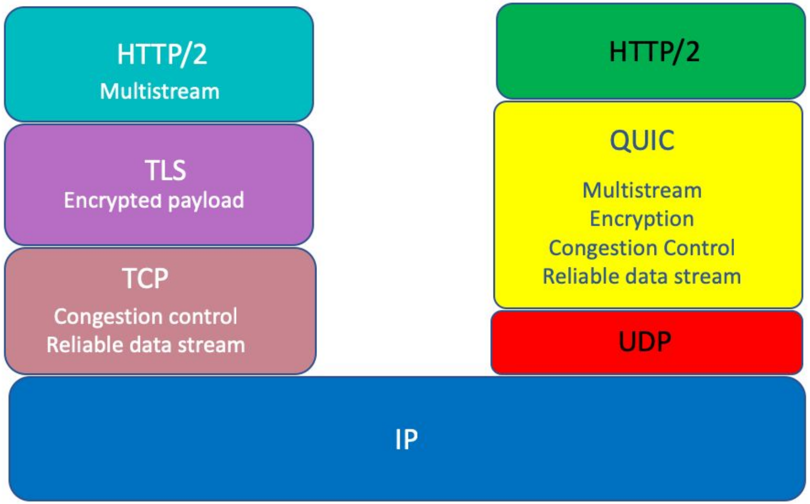
On pitfalls, attacks, and discovering hypergiant infrastructures with QUIC

Jonas Mücke <jonas.muecke@fu-berlin.de>

What is QUIC?

- A new transport protocol
    - supports multiple streams
- UDP based
    - implements reliable data streams and congestion control
- Encryption built-in
    - even metadata is protected

2020: 75 % of Facebook's Internet Traffic is QUIC.



https://blog.apnic.net/2019/03/04/a-quick-look-at-quic/

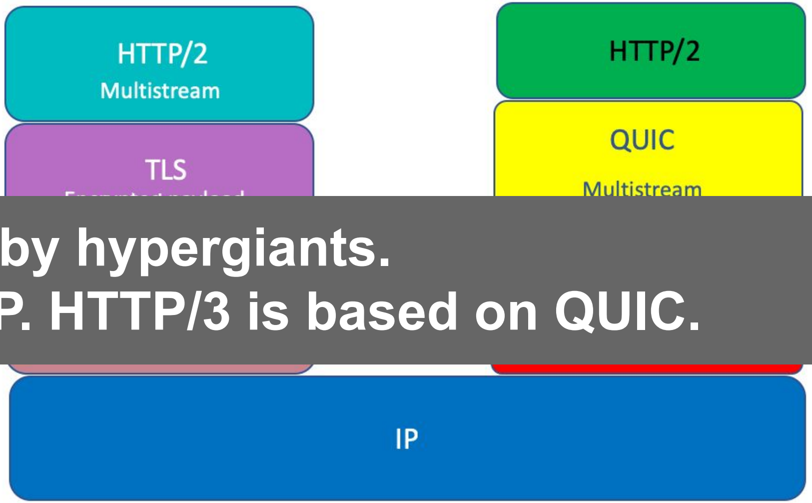
What is QUIC?

- A new transport protocol
  - supports multiple streams
- UDP based

QUIC is widely used by hypergiants.
It has beneficial features for HTTP. HTTP/3 is based on QUIC.

- Encryption baked in
  - even metadata is protected

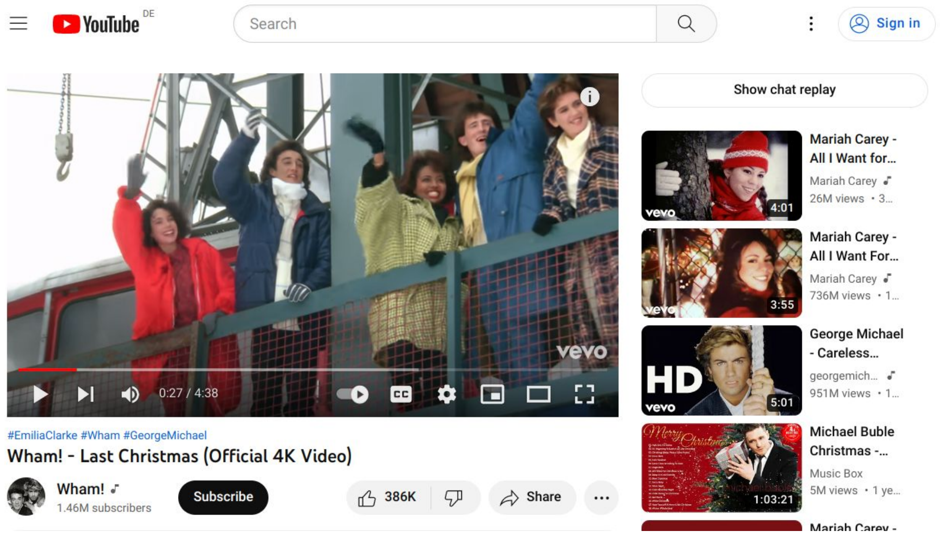
2020: 75 % of Facebook's Internet Traffic is QUIC.



https://blog.apnic.net/2019/03/04/a-quick-look-at-quic/

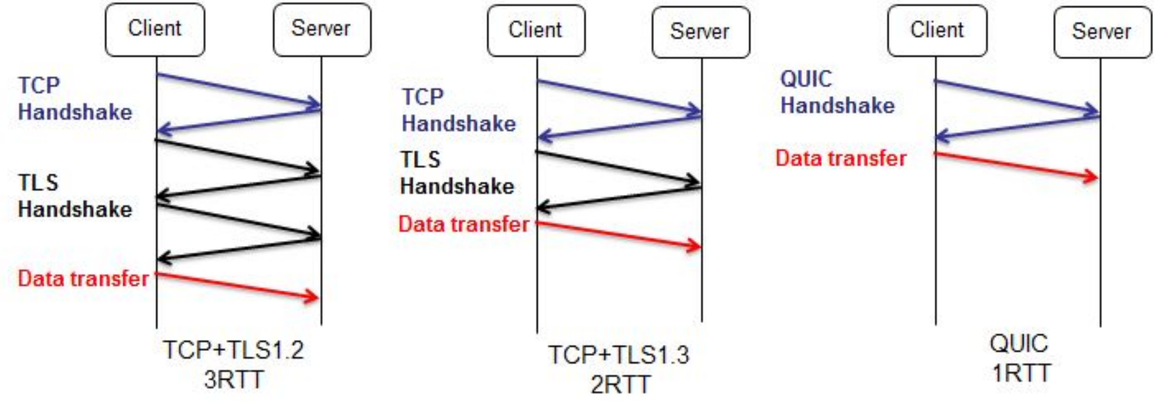
You access www.youtube.com to stream a christmas song

You want encryption, because you don't want your colleagues to know that you want to stream Last Christmas.



https://www.youtube.com/watch?v=bwNV7TAWN3M

Why is QUIC faster?



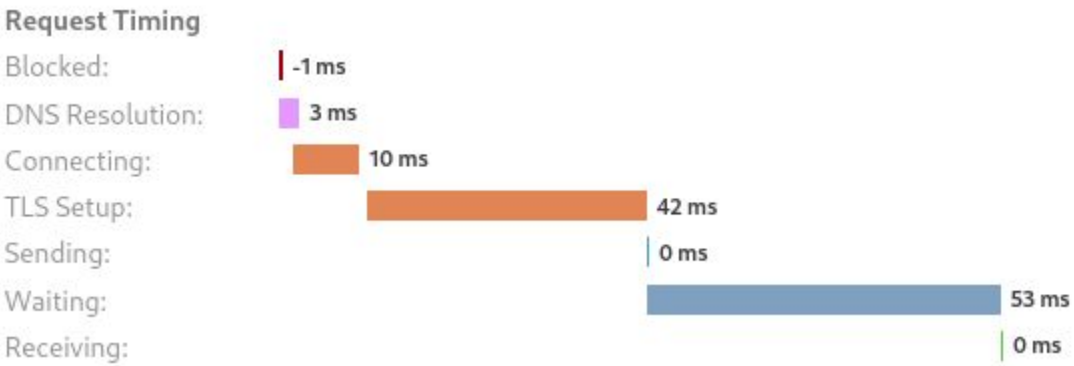
The TLS handshake is embedded into the QUIC handshake.

QUIC-Handshake = TCP-Handshake and TLS-Handshake

Reduction of 1-2 RTTs.

Cui et al., Innovating Transport with QUIC: Design Approaches and Research Challenges. IEEE Internet Computing, 21(2):72–76

How does it look in your browser?



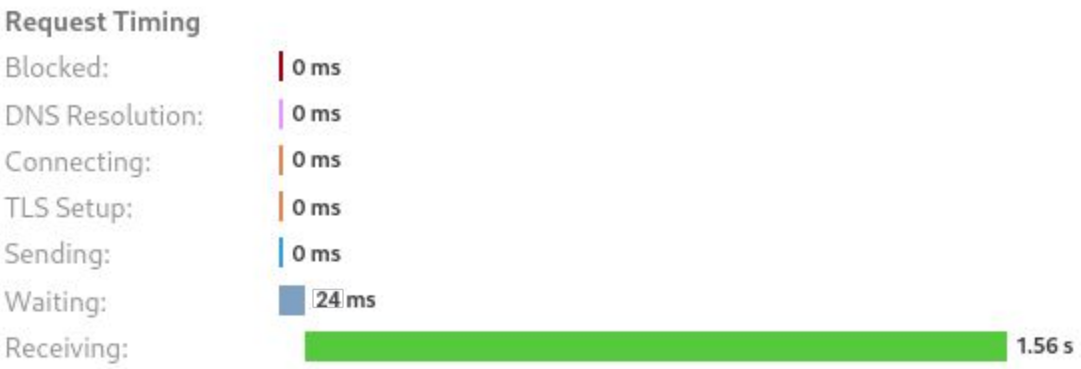
www.youtube.com with Firefox and TCP + TLS (HTTP/2)

TCP Handshake: 10ms

TLS Handshake: 42ms

Total: 52ms

How does it look in your browser?



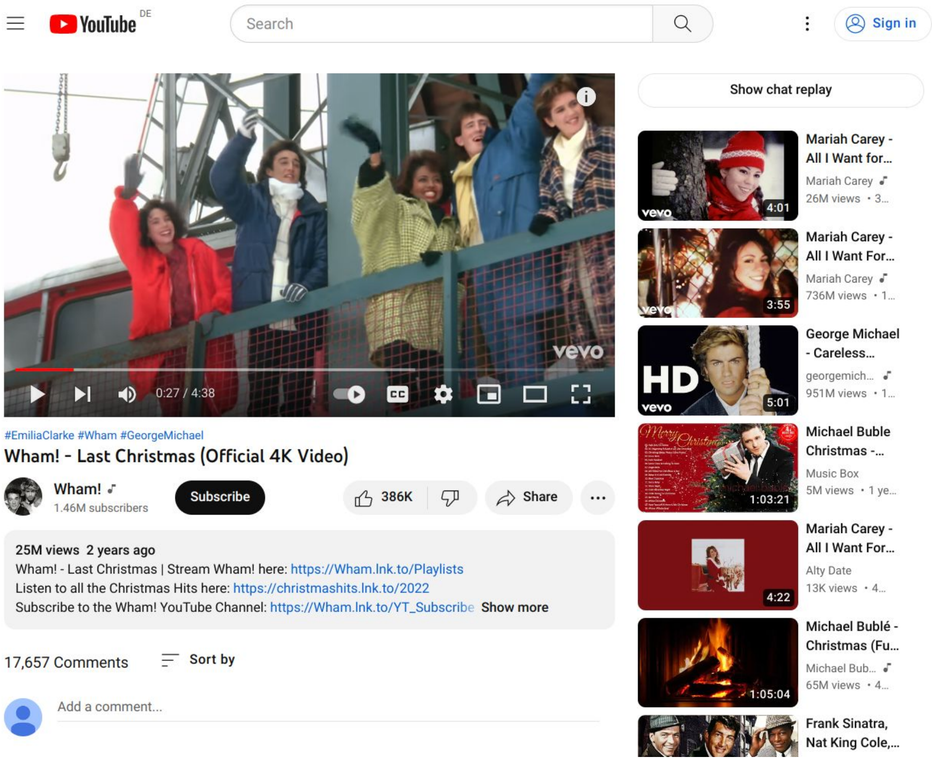
www.youtube.com with Firefox and QUIC (HTTP/3, best case)

QUIC Handshake: 24ms

Total: 24ms (~54% reduction)

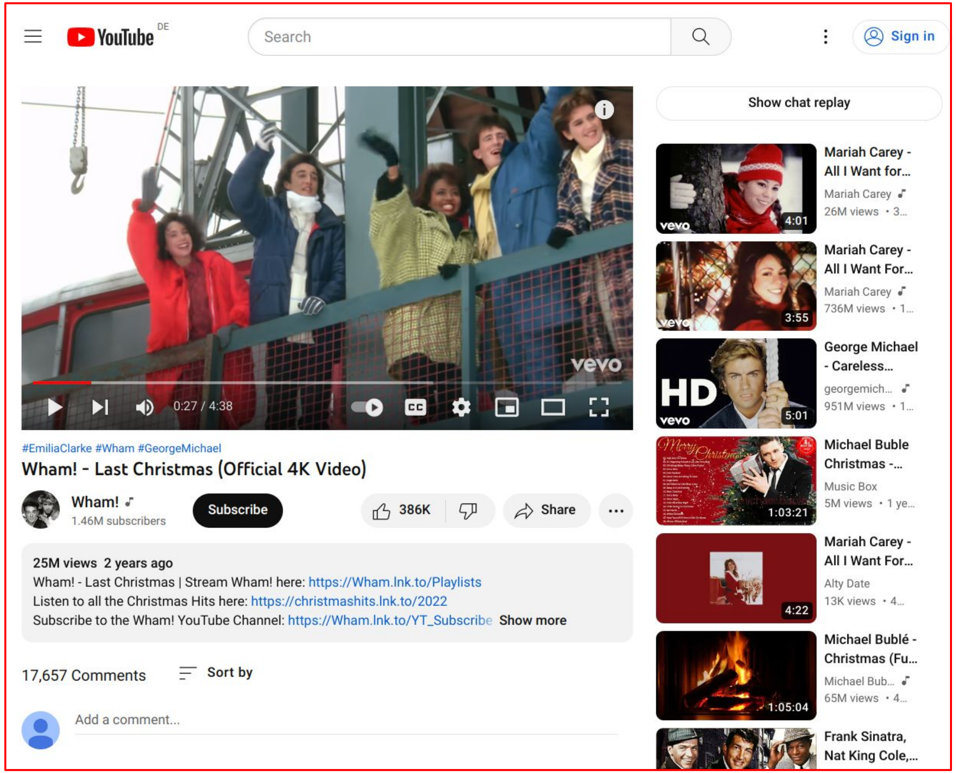
Loading content

Most websites consist of more information than html. Styling information(css), images and support for interactivity (javascript) is required.



Loading content

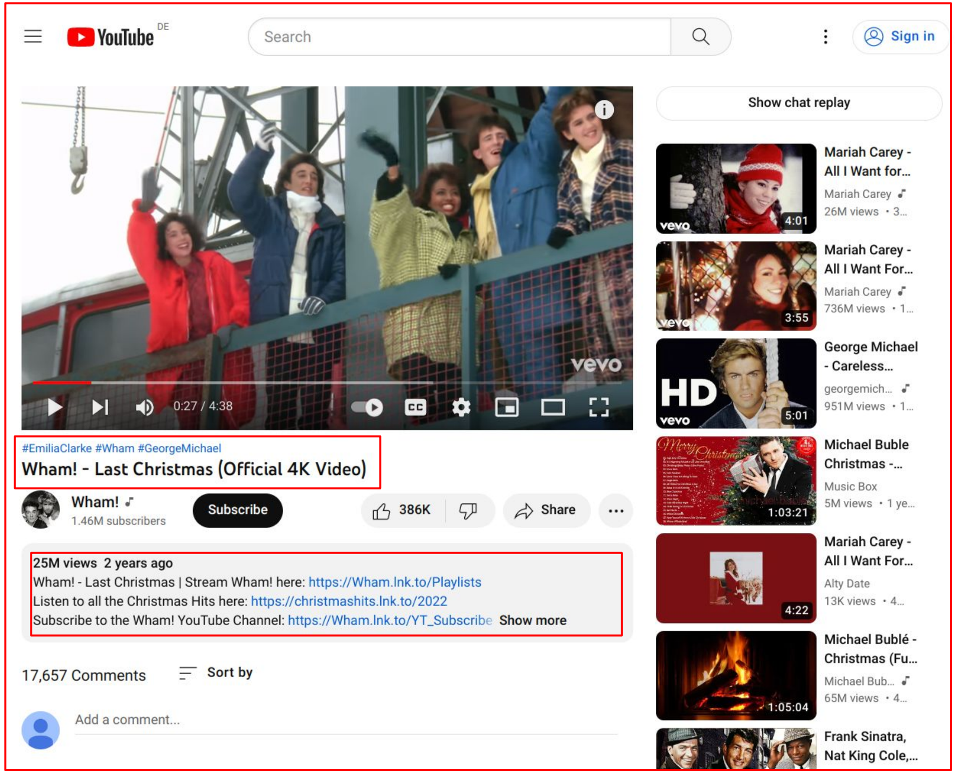
Most websites consist of more information than html. Styling information(css), images and support for interactivity (javascript) is required.



html

Loading content

Most websites consist of more information than html. Styling information(css), images and support for interactivity (javascript) is required.

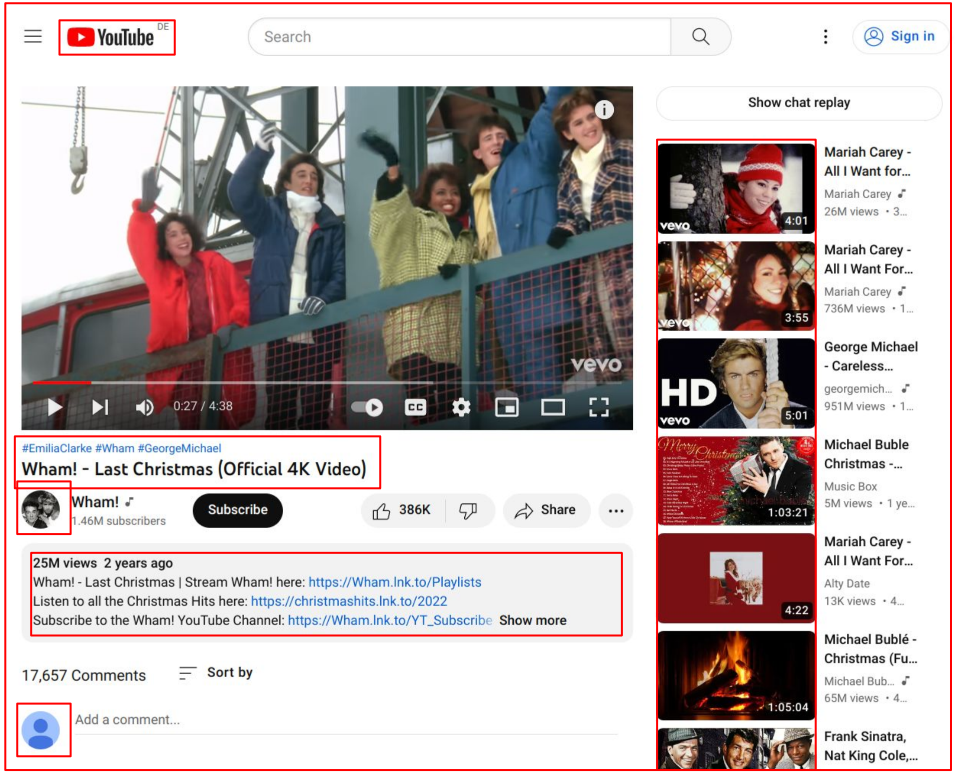



html


css

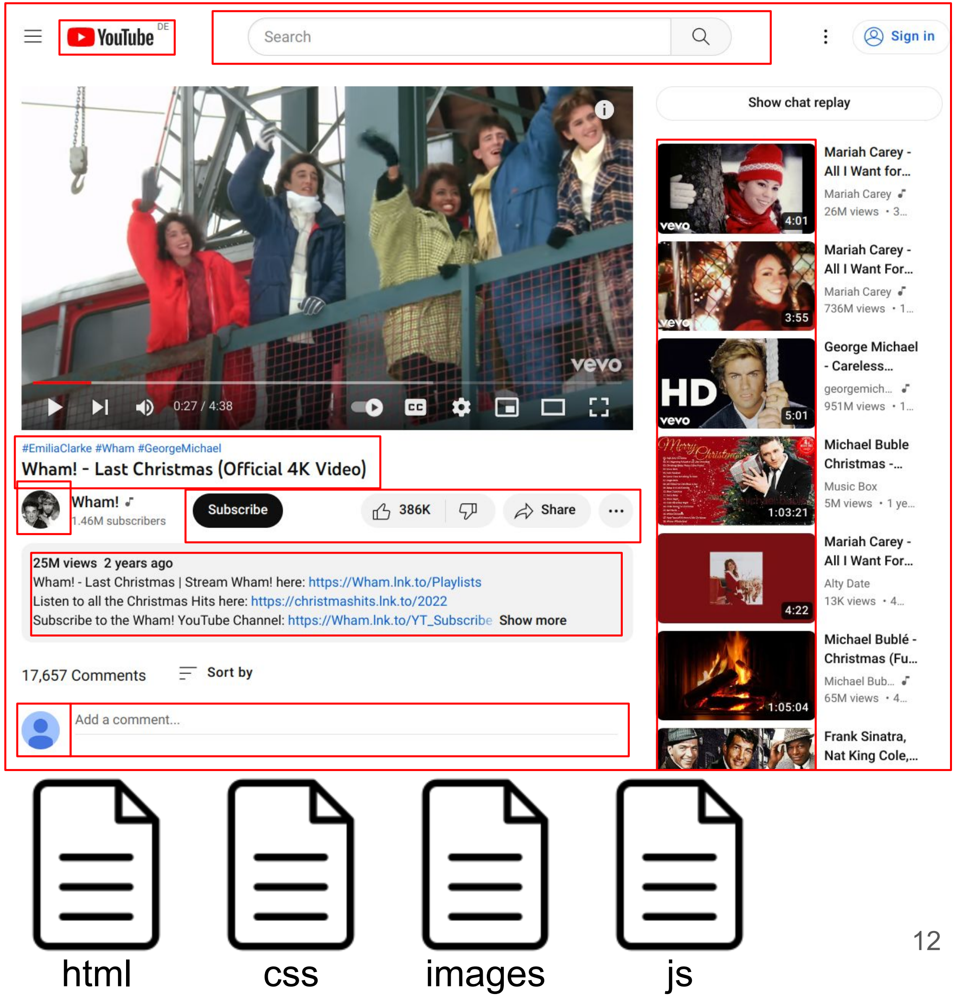
Loading content

Most websites consist of more information than html. Styling information(css), images and support for interactivity (javascript) is required.



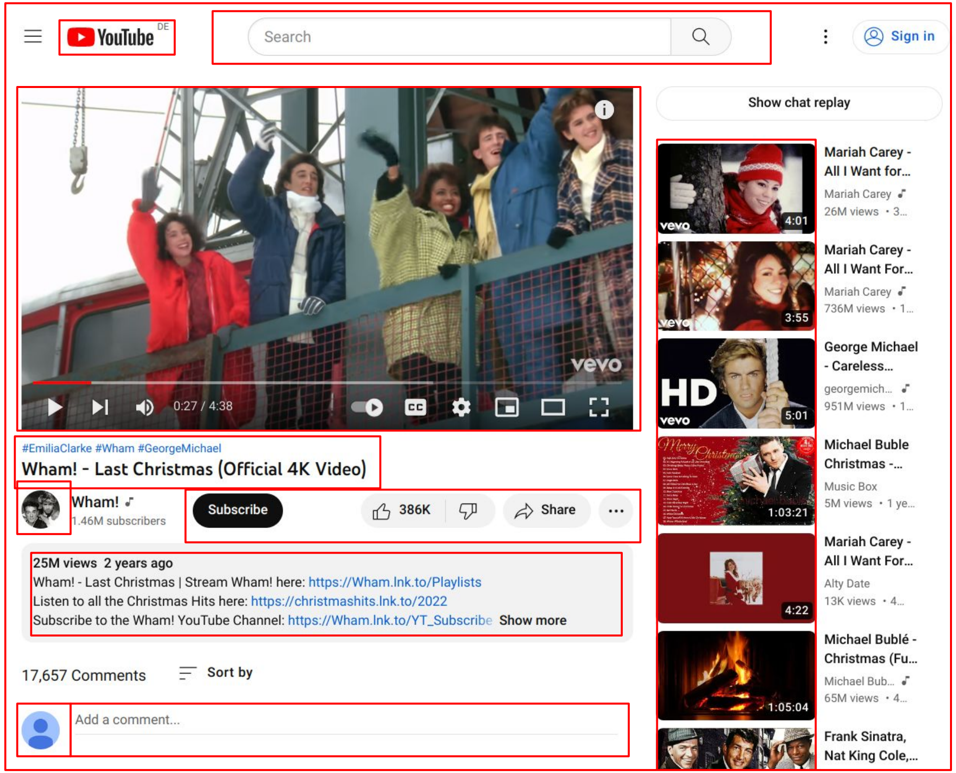
Loading content

Most websites consist of more information than html. Styling information(css), images and support for interactivity (javascript) is required.



Loading content

Most websites consist of more information than html. Styling information(css), images and support for interactivity (javascript) is required.



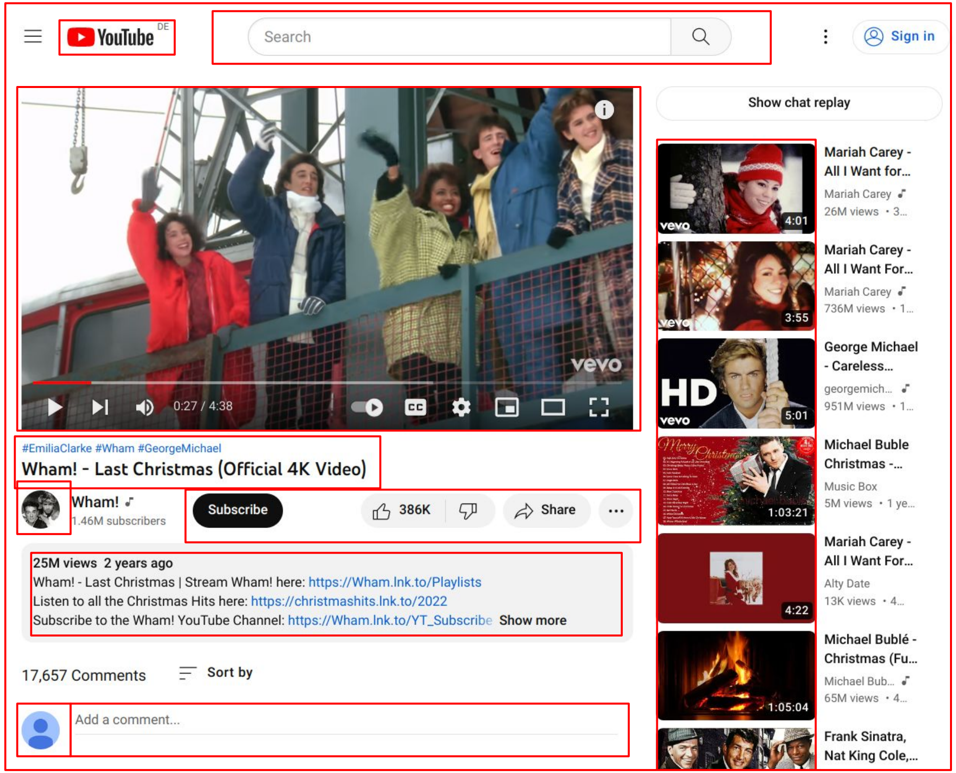
Loading content

Most websites consist of more information than html. Styling information(css), images and support for interactivity (javascript) is required.

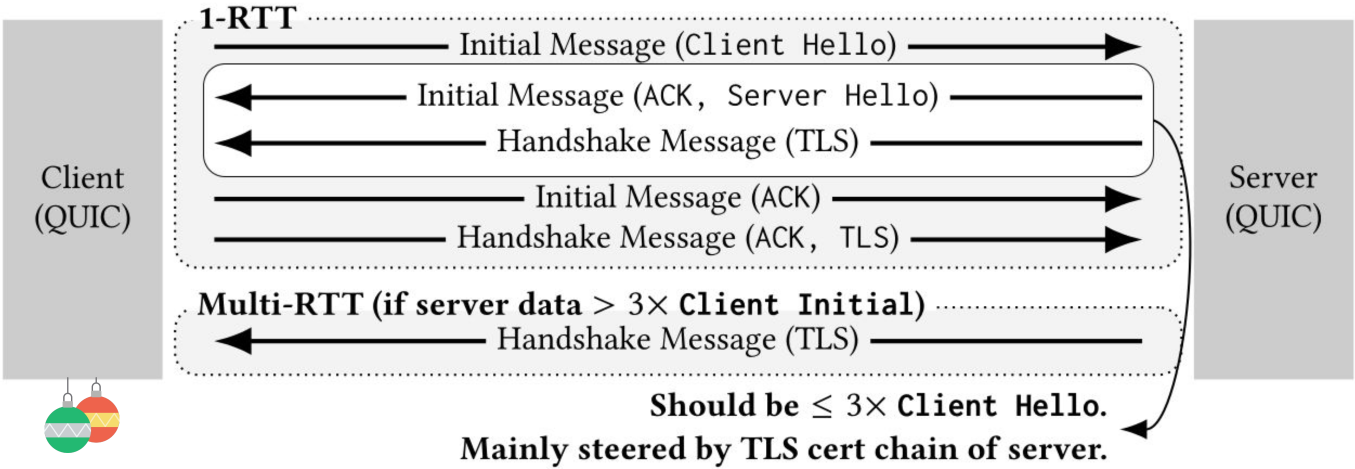
QUIC can fetch the different files independently.

Packet loss only affects a single stream/file.

In TCP all streams are blocked if a single packet is lost, because it only supports a single reliable data-stream (head-of-line blocking).



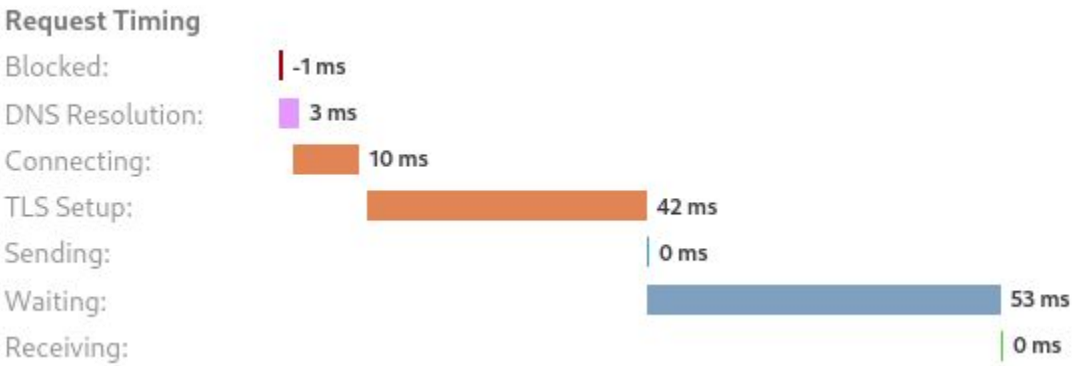
How does the Handshake look like?



Before the client IP address is verified, the server is allowed to send up to 3X the size of the UDP payload it received.

The TLS certificate is included in the handshake message from the server.

How does it look in your browser?



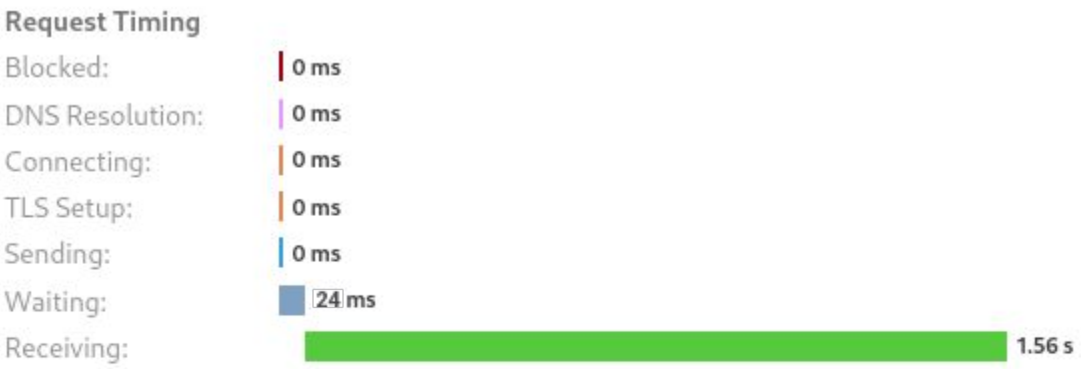
www.youtube.com with Firefox and TCP + TLS (HTTP/2)

TCP Handshake: 10 ms

TLS Handshake: 42 ms

Total: 52 ms

How does it look in your browser?

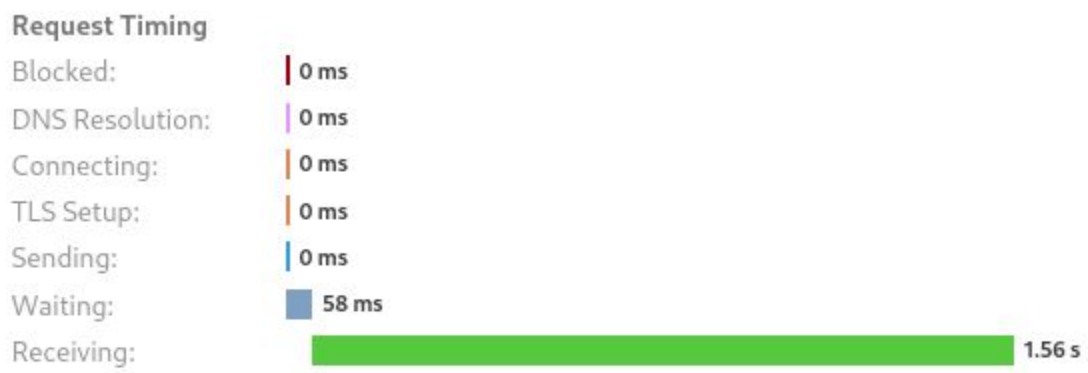


www.youtube.com with Firefox and QUIC (HTTP/3, best case)

QUIC Handshake: 24 ms

Total: 24 ms (~54% reduction)

The actual QUIC-handshake

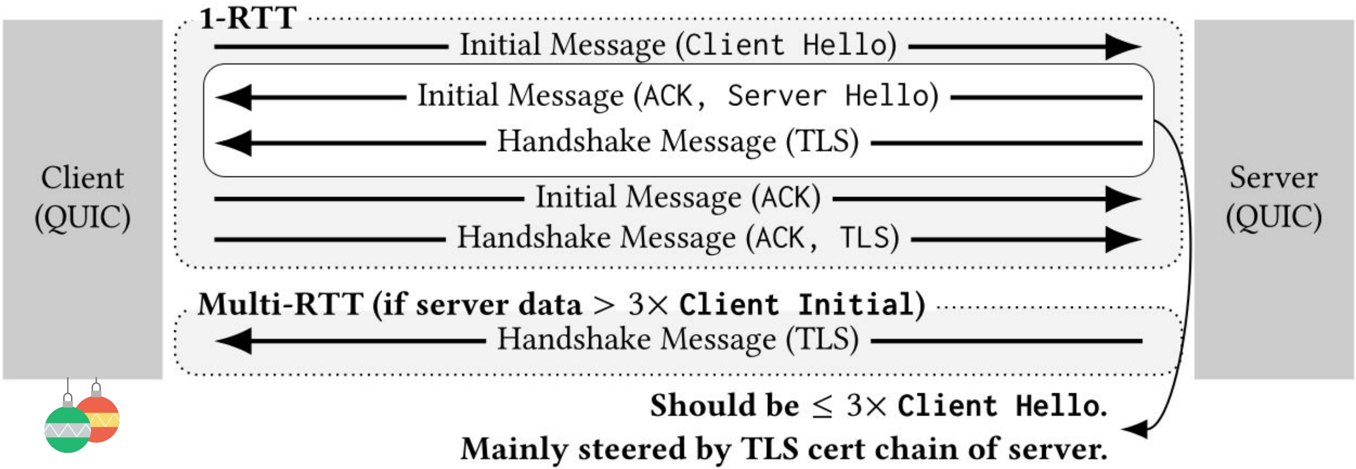


www.youtube.com with Firefox and QUIC (HTTP/3; reality)

QUIC Handshake: 58 ms

Total: 58 ms (~11% increase)

How does the Handshake look like?



Before the client IP address is verified, the server is allowed to send up to 3X the size of the UDP payload it received.

The TLS certificate is included in the handshake message from the server.

Is this a general problem or just a single bad example?

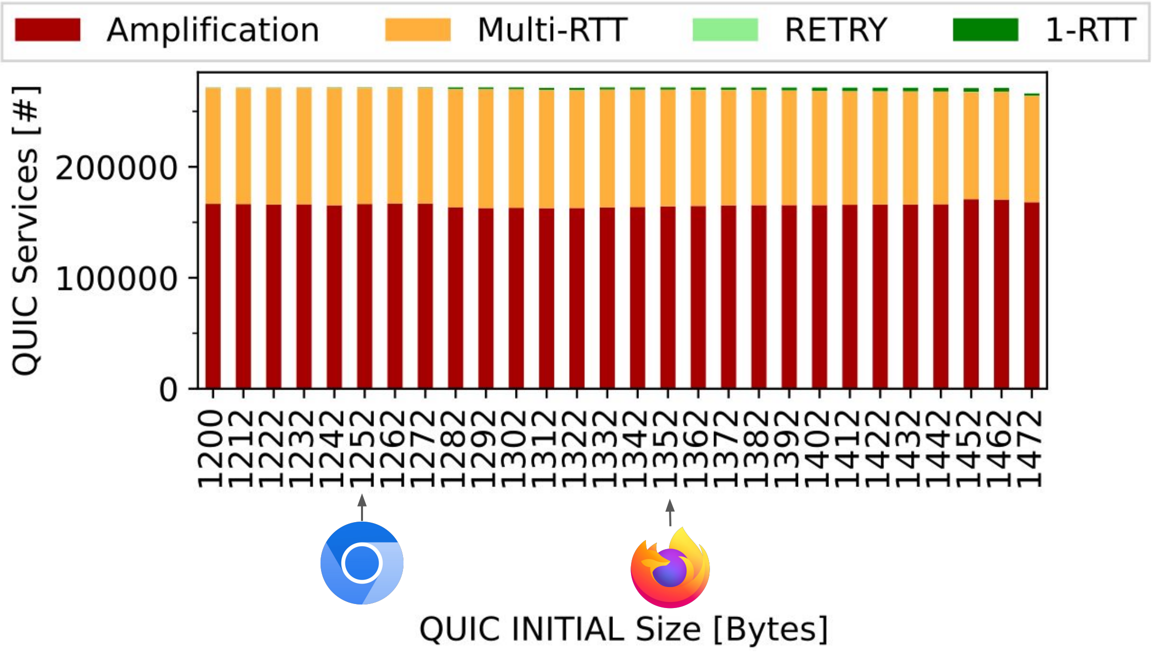
Measurement study on the Tranco Top 1M list.

We connect via HTTP(S) and QUIC to all domains and collect TLS certificates.

Results:

- 272k QUIC supporting domains.

How often do we encounter those cases



Amplification

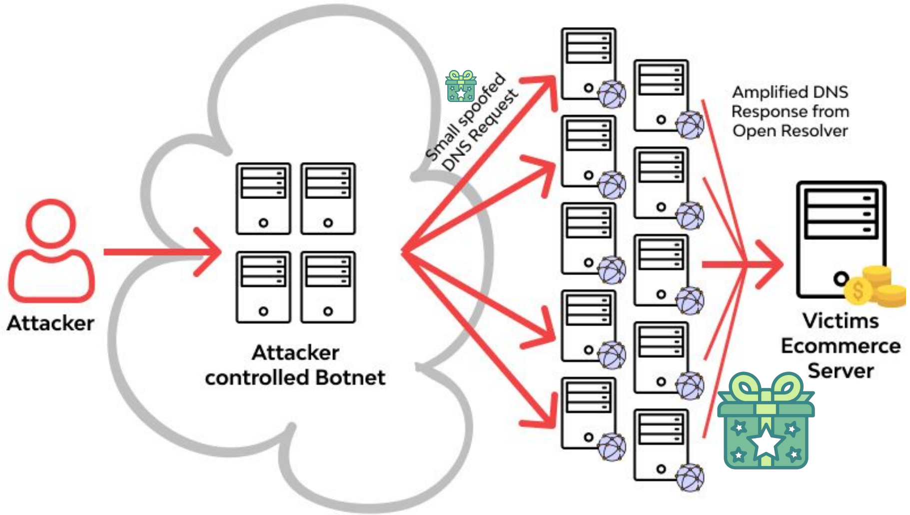
UDP operates connection-less.

No handshake = no verification of source IP address = potential for amplification attacks.

Typical amplification factor of DNS: 28X to 54X

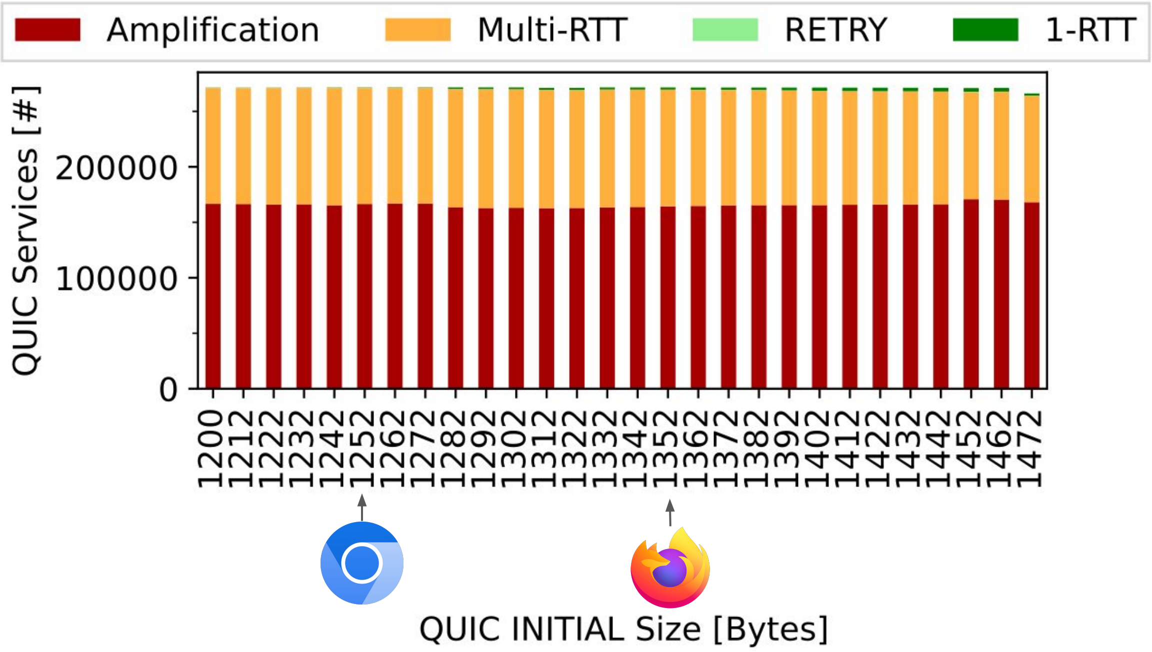
QUIC servers are allowed to send 3X the UDP payload size received from a client.

QUIC should be unattractive for amplification attacks.



https://www.wallarm.com/what/dns-amplification-attacks-explained
https://www.cisa.gov/uscert/ncas/alerts/TA14-017A

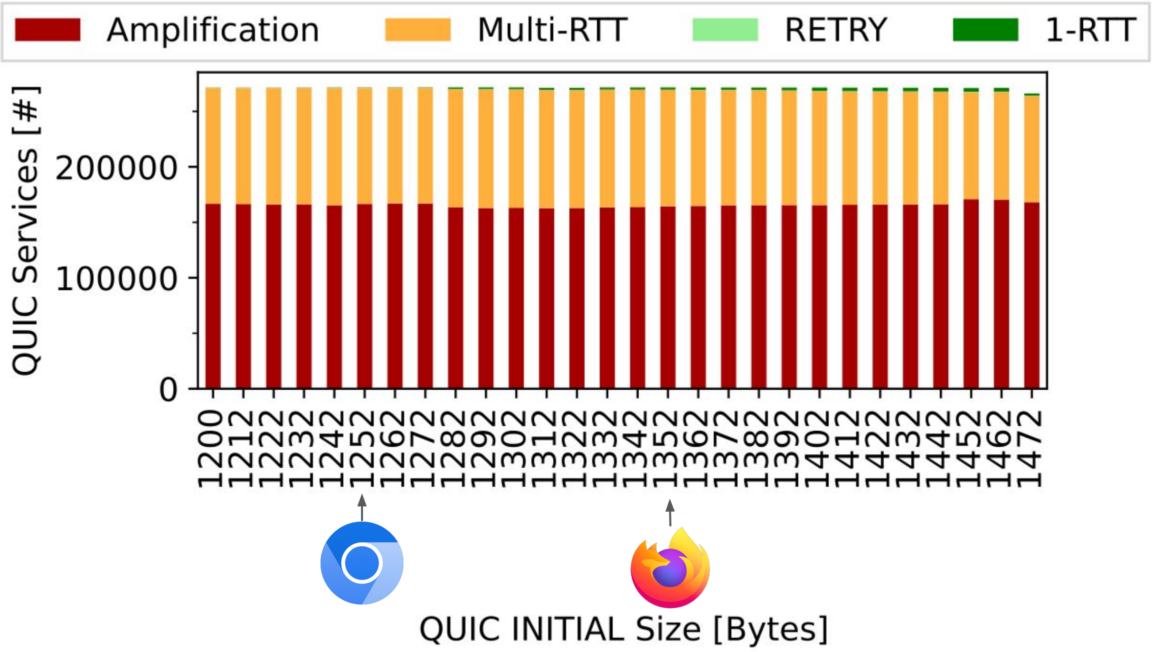
How often do we encounter those cases



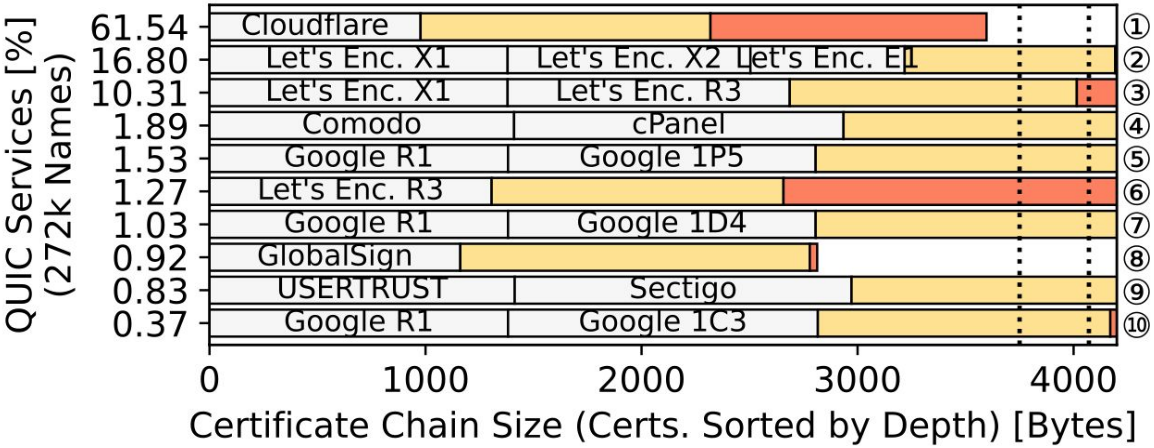
The other two cases: RETRY and 1-RTT

- Multi-RTT (unnecessary): Handshakes that do not use Retry but require multiple RTTs because of large certificates.
- Amplification (not RFC-compliant): Handshakes that complete within 1-RTT but exceed the anti-amplification limit.
- **RETRY** (less efficient): Handshakes that require multiple RTTs because the Retry option is used.
- **1-RTT** (optimal): Handshakes that complete within 1-RTT and comply with the anti-amplification limit.

How often do we encounter those cases



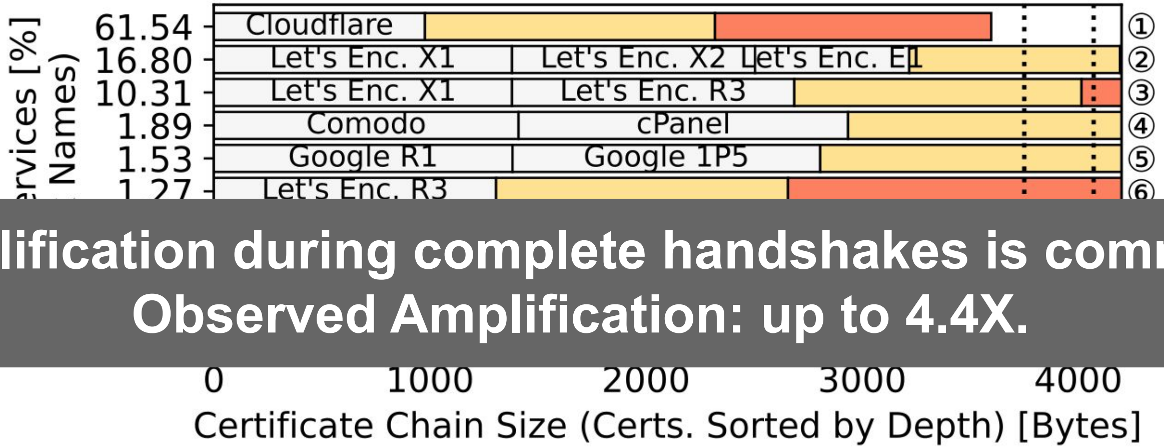
What is the main reason?



Large certificate chains are the main reason.

Effective TLS setup influences the performance of the transport protocol.

What is the main reason?

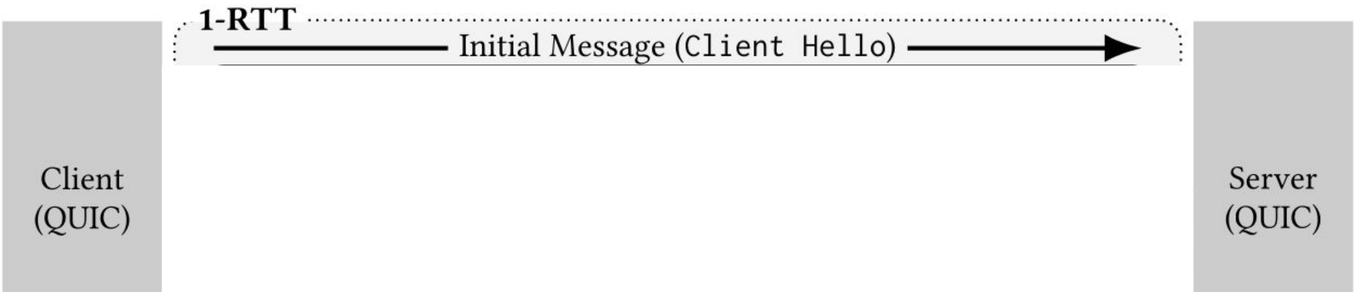


Amplification during complete handshakes is common.
Observed Amplification: up to 4.4X.

Large certificate chains are the main reason.

Effective TLS setup influences the performance of the transport protocol.

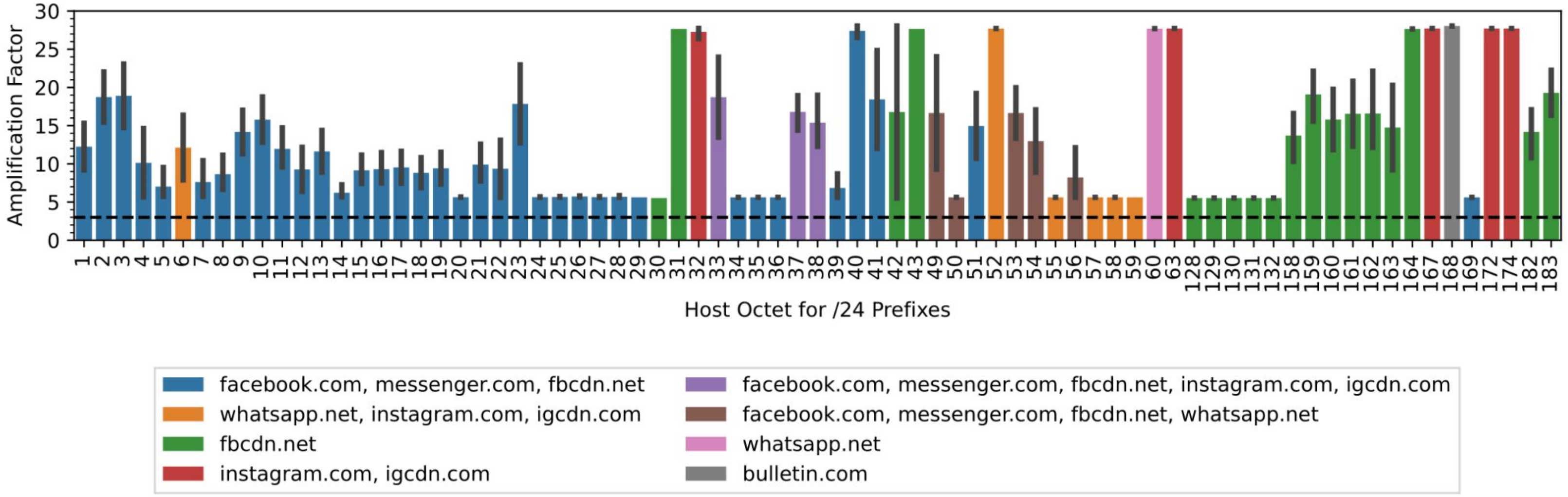
What about incomplete handshakes?



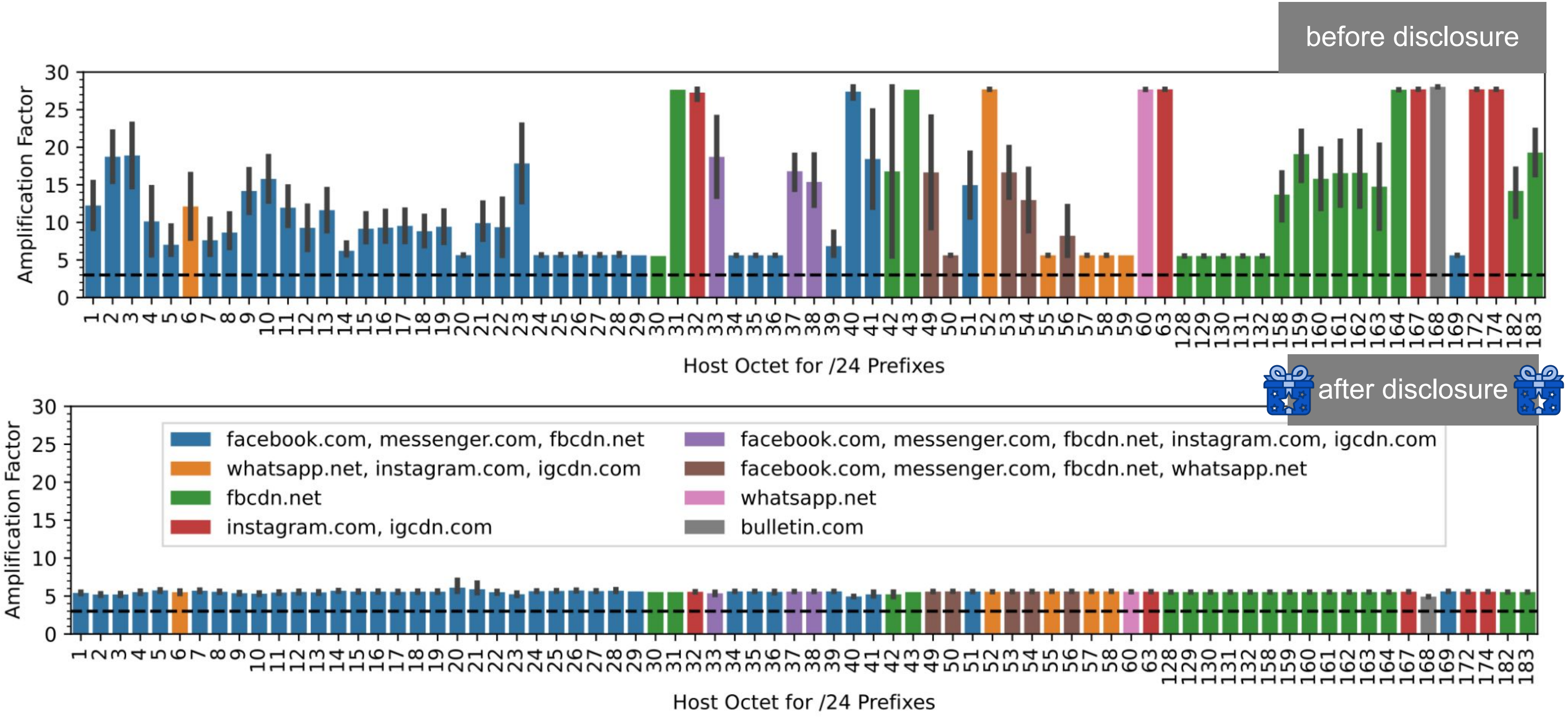
Send a client Initial packet to a server and collect response traffic, but do not send any other packets.

We scanned all Facebook IPv4 QUIC servers.

Amplification in incomplete Handshakes



Amplification in incomplete Handshakes



Can we improve the situation?

The 3X anti-amplification limit and certificates impact the performance of the QUIC handshake.

Mitigations:

- Reduction of certificate size using other signing algorithms (ECDSA vs. RSA)
- Enabling certificate compression. Not all TLS libraries support it yet.
- Packet coalescence should be enabled.
- Resend packets and padding must be included in anti-amplification checks.

On lossy links only one resend is possible within the 3X limit.

Test your websites on understanding-quic.net!

QUIC Classification

Welcome to the QUIC classification project. This is a project of Freie Universität Berlin and HAW Hamburg.

We classify QUIC Handshakes in a user friendly way.

www.youtube.com | Analyze

Show advanced options

We might collect the server name you want to analyze and the measurement results.

Client Initial 1250 Bytes (Chromium default)

Multi-RTT Handshake	24.410ms RTT	3.3x send/receive ratio
Multi-RTT Handshake (inefficient)	Initial complete: 25.136ms Handshake complete: 58.560ms	Data sent: 2500B (3 Pkts.) Data received: 8326B (7 Pkts.)

Client Initial 1350 Bytes (Firefox default)

Multi-RTT Handshake	24.081ms RTT	3.1x send/receive ratio
Multi-RTT Handshake (inefficient)	Initial complete: 24.885ms Handshake complete: 58.562ms	Data sent: 2700B (3 Pkts.) Data received: 8426B (7 Pkts.)

TLS Certificate Compression

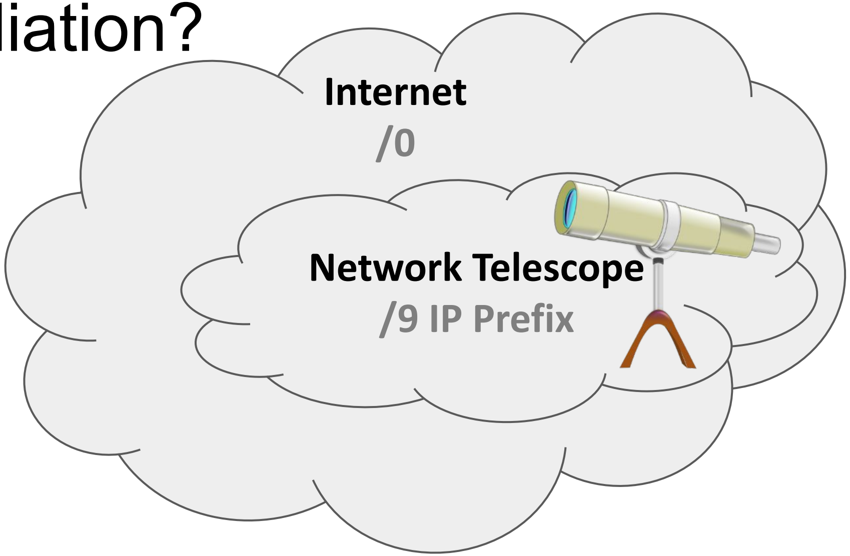
zlib	zstd	39.4 %
Compression algorithm not supported.	Compression algorithm not supported.	reduction

understanding-quic.net

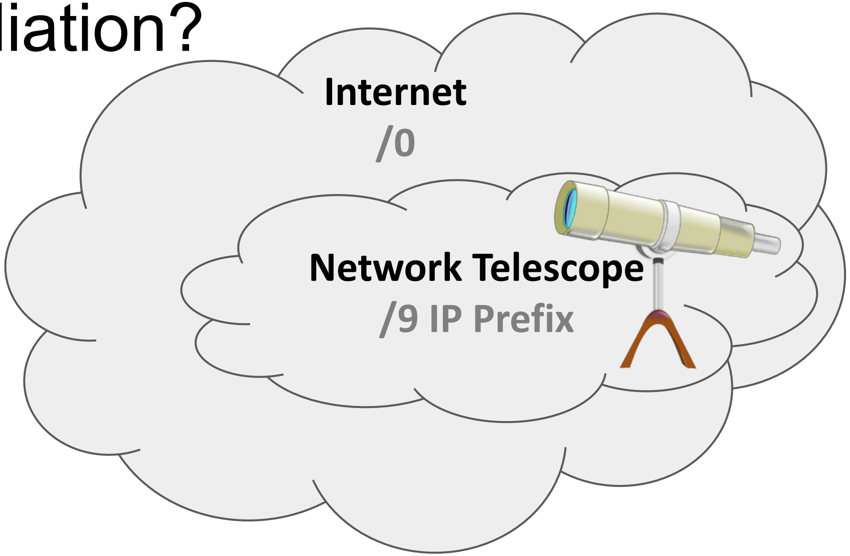
Active measurements are great. But what can we do with passive measurements?

We use Internet Background Radiation

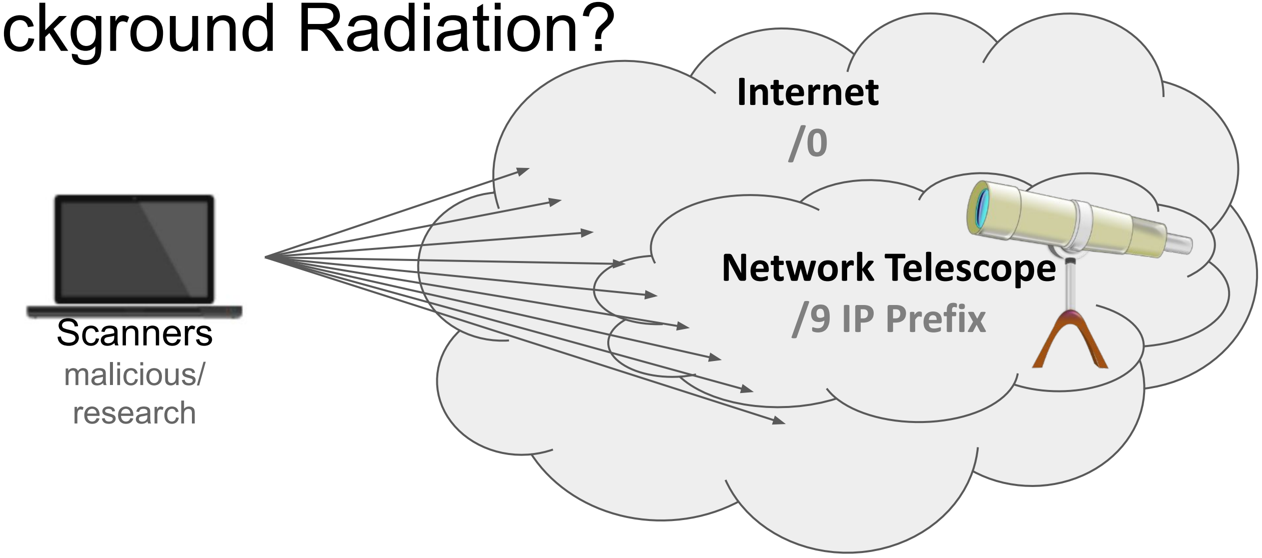
What is Internet Background Radiation?



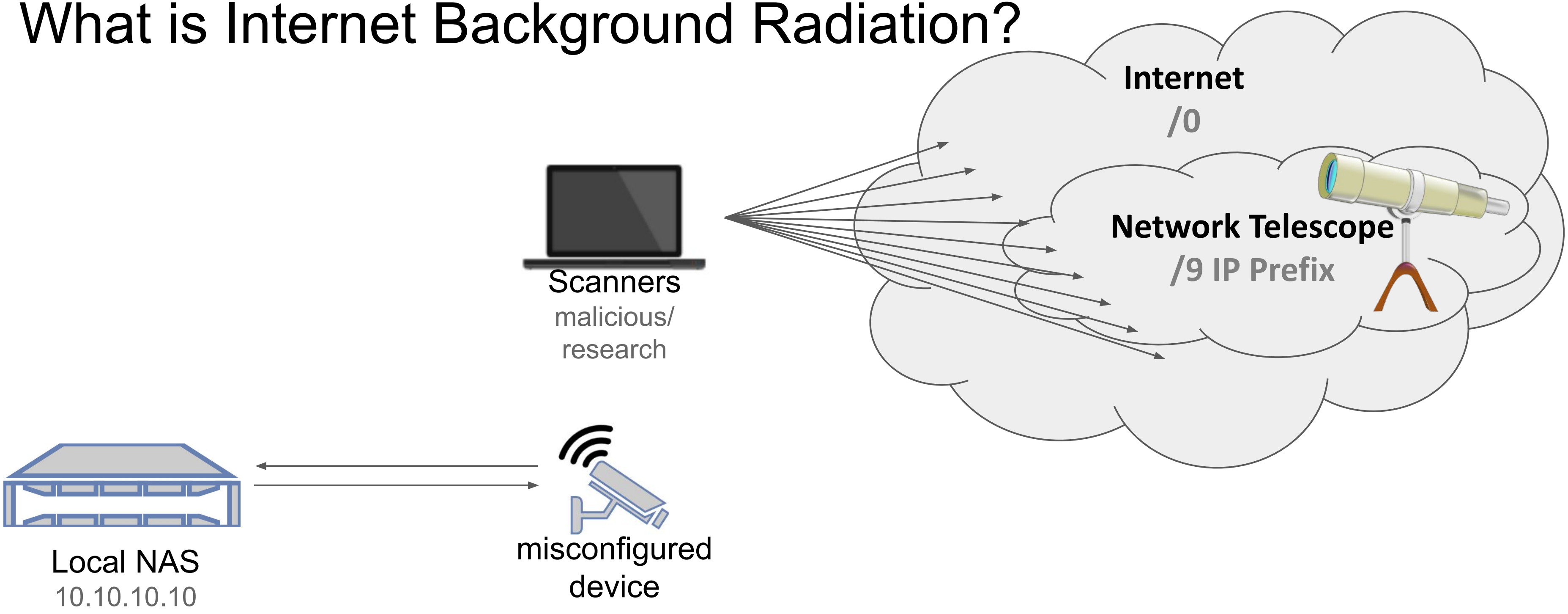
What is Internet Background Radiation?



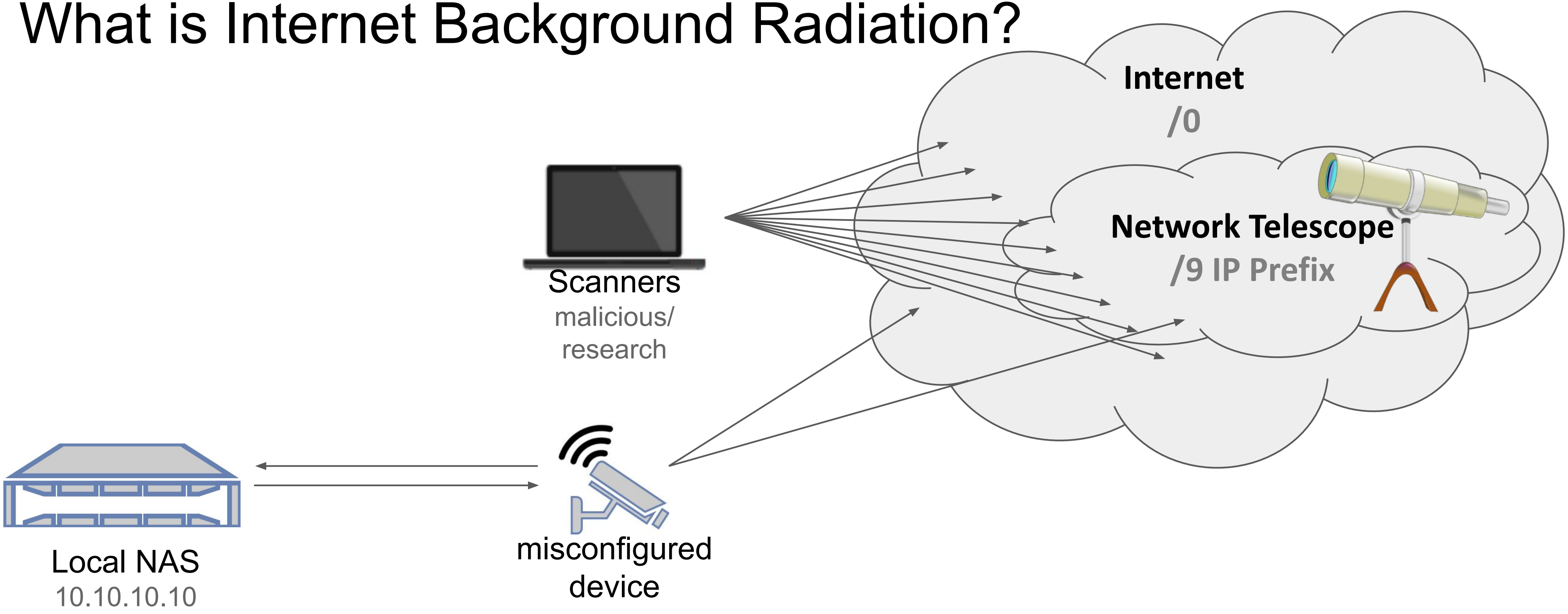
What is Internet Background Radiation?



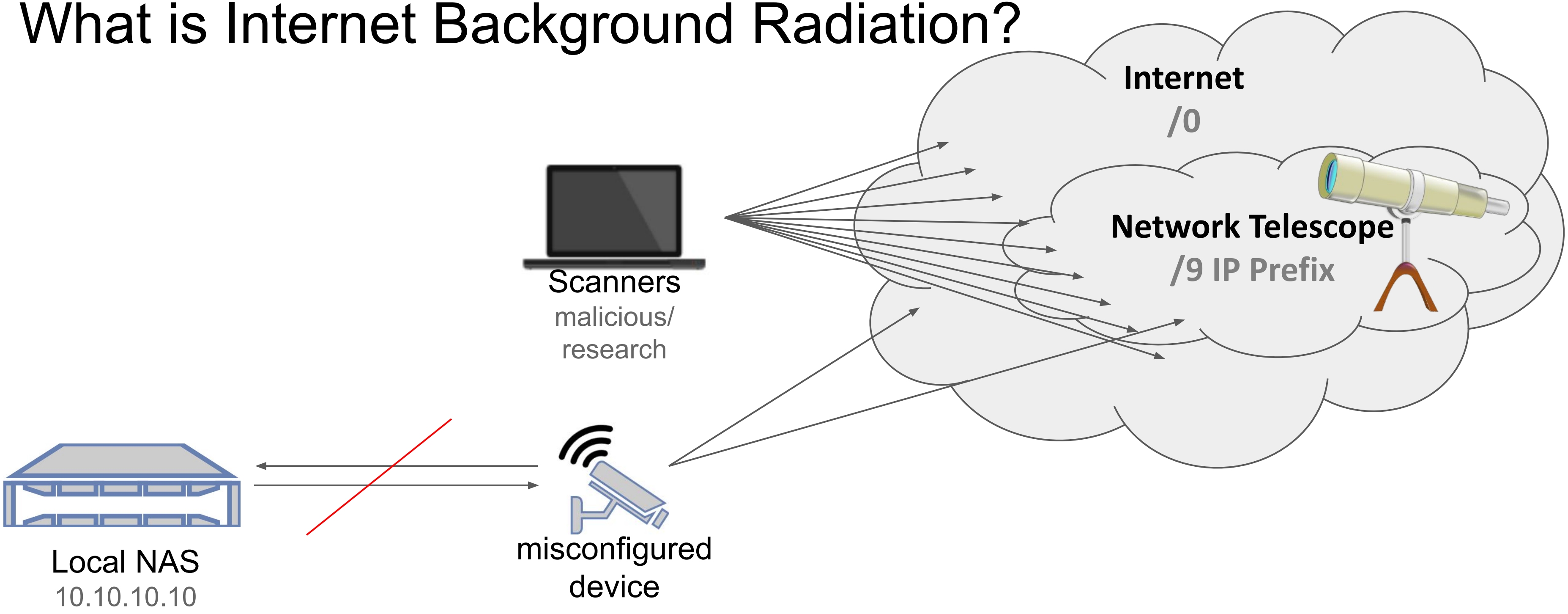
What is Internet Background Radiation?



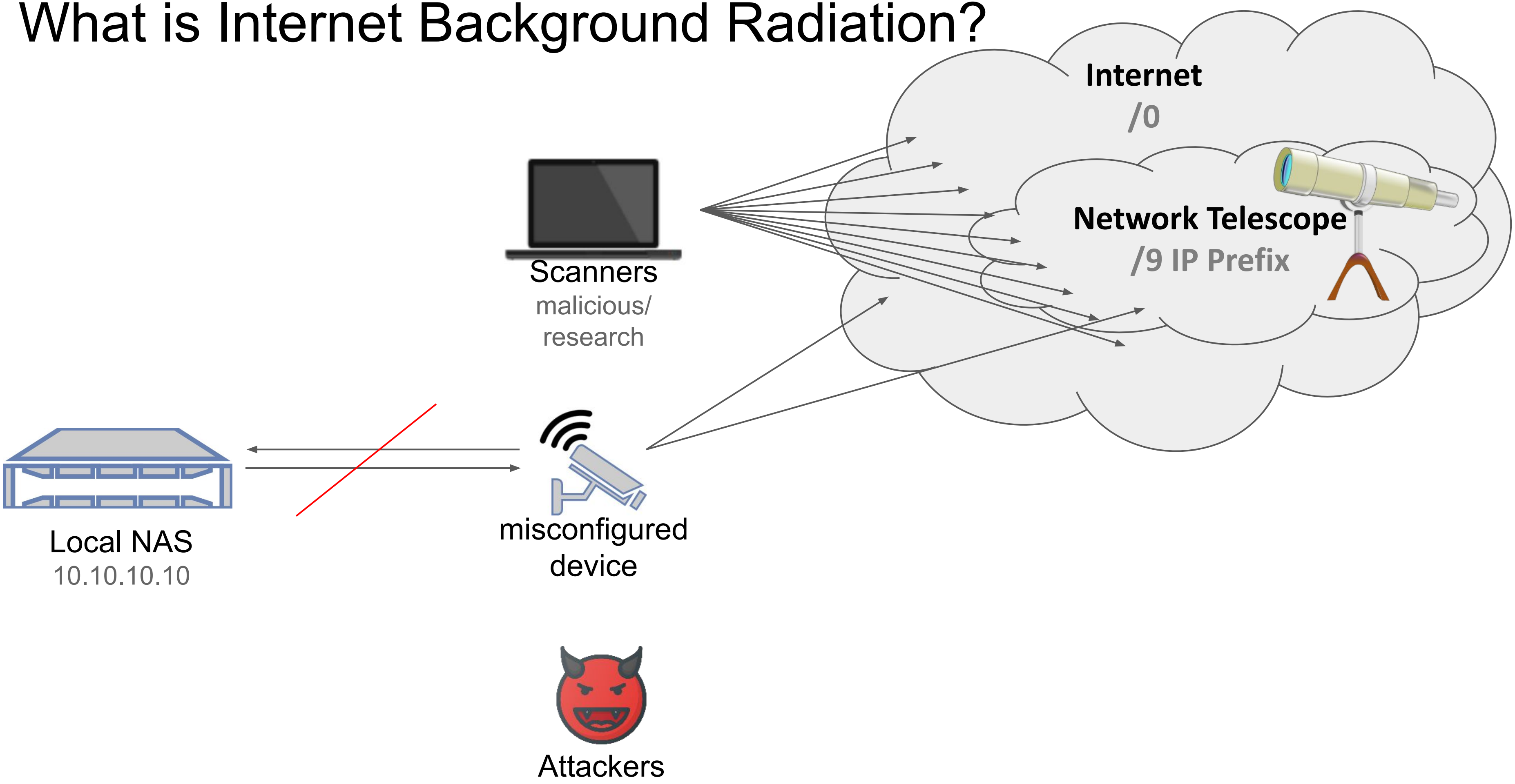
What is Internet Background Radiation?



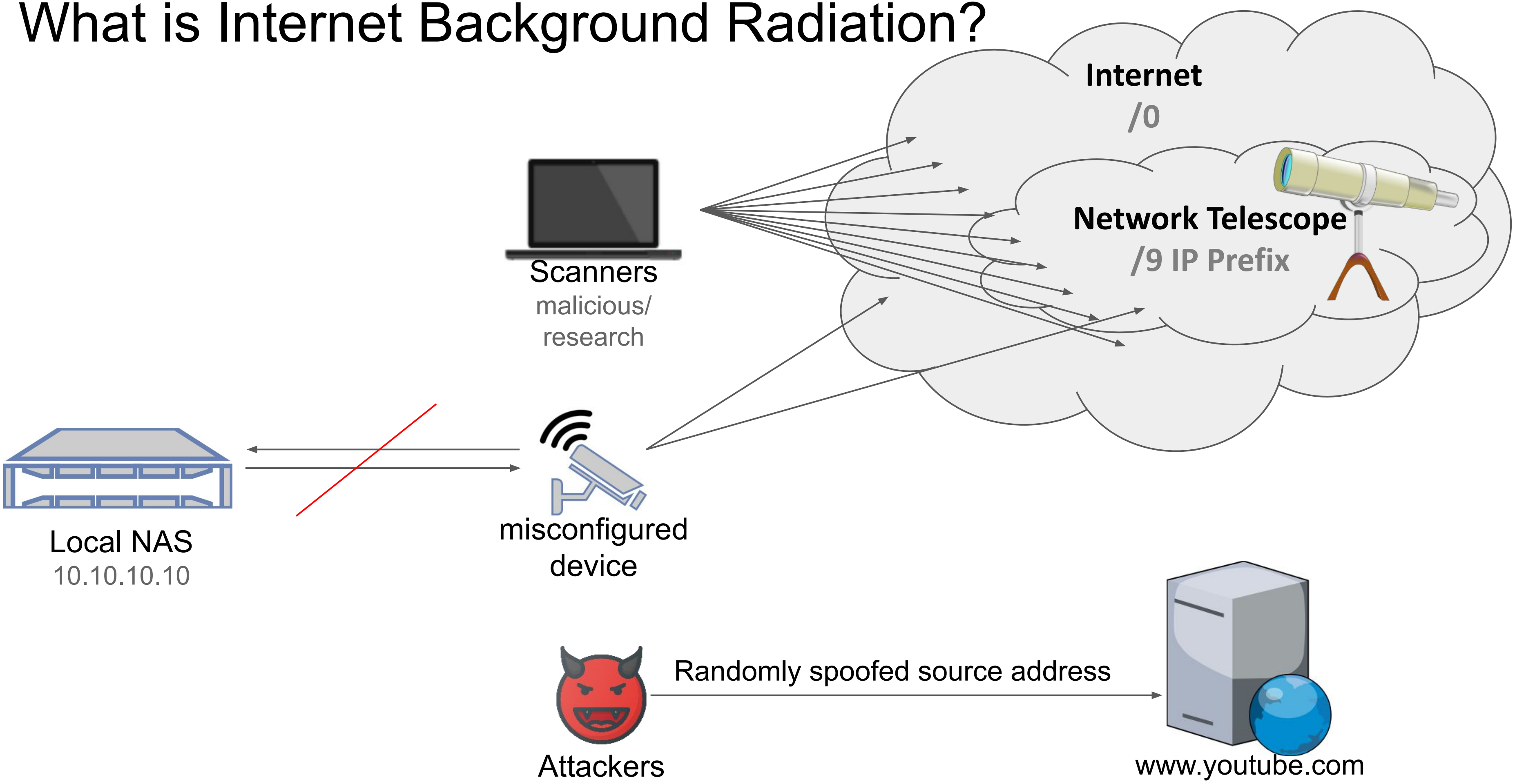
What is Internet Background Radiation?



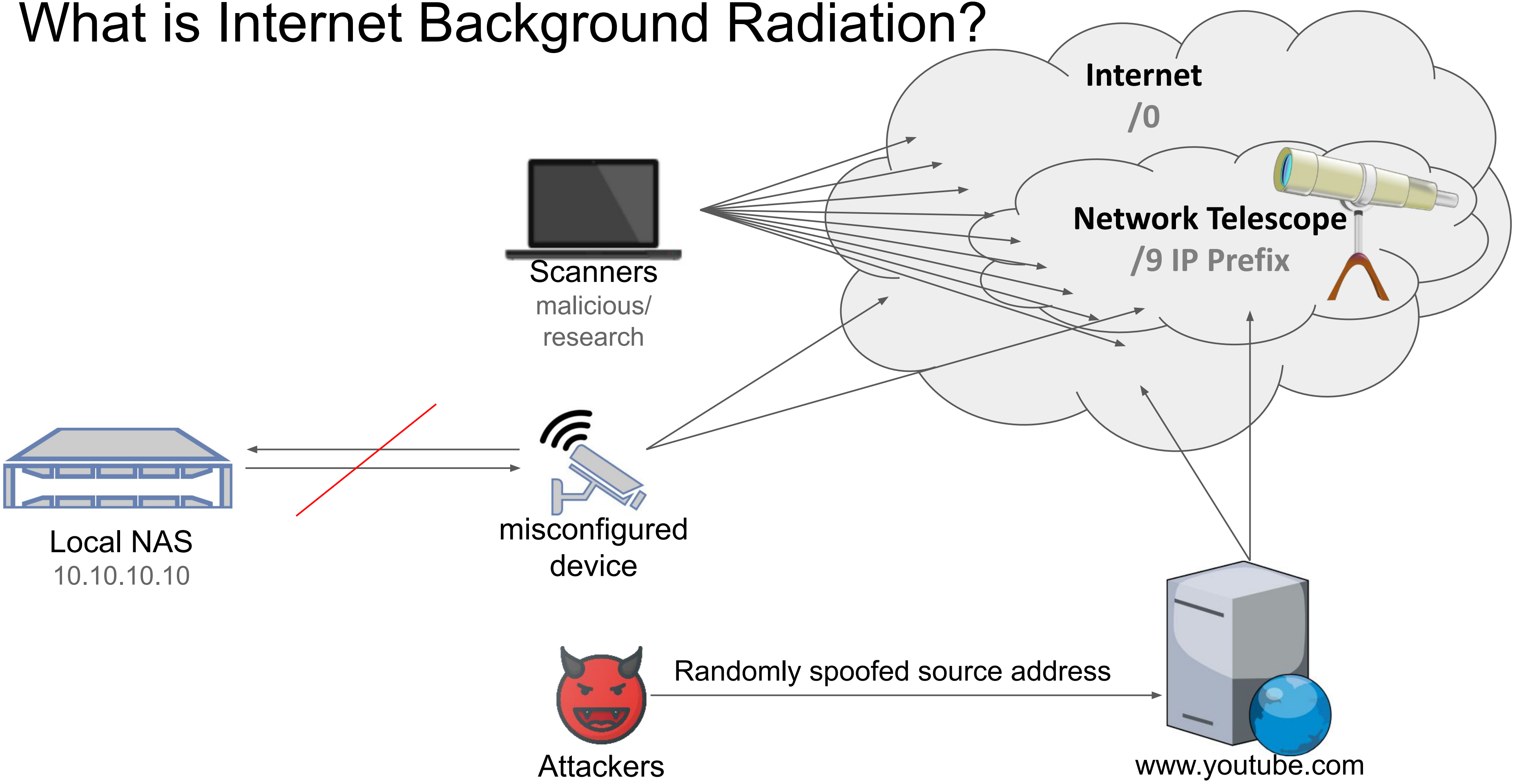
What is Internet Background Radiation?



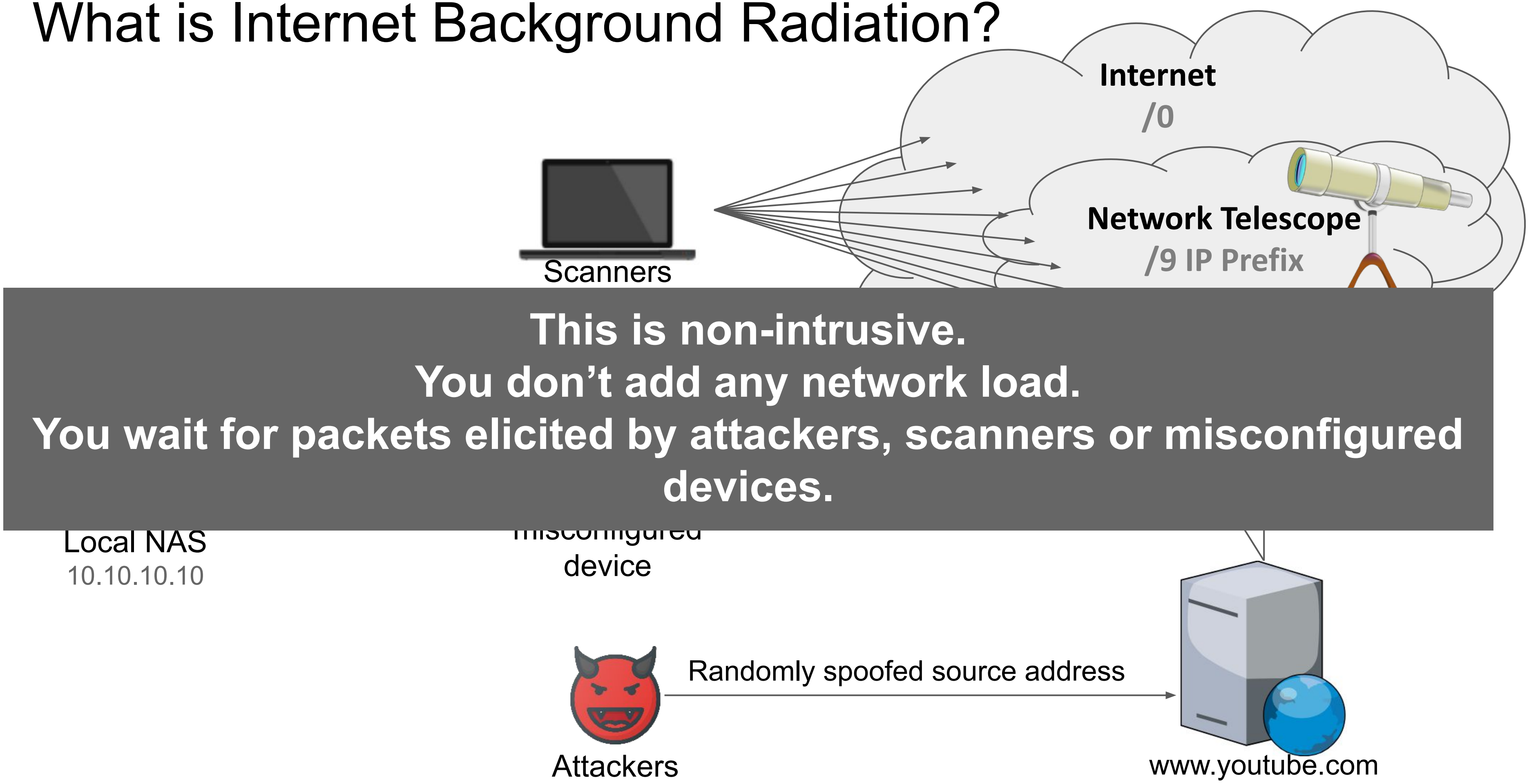
What is Internet Background Radiation?



What is Internet Background Radiation?



What is Internet Background Radiation?

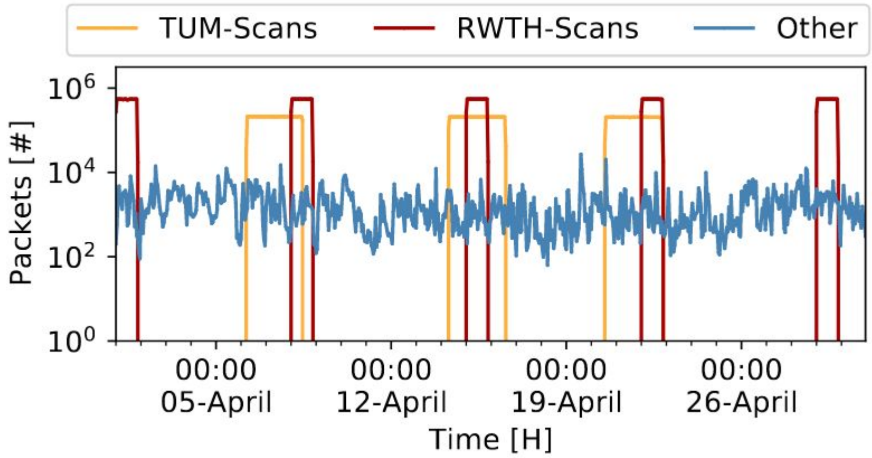


This is non-intrusive.
You don't add any network load.
You wait for packets elicited by attackers, scanners or misconfigured devices.

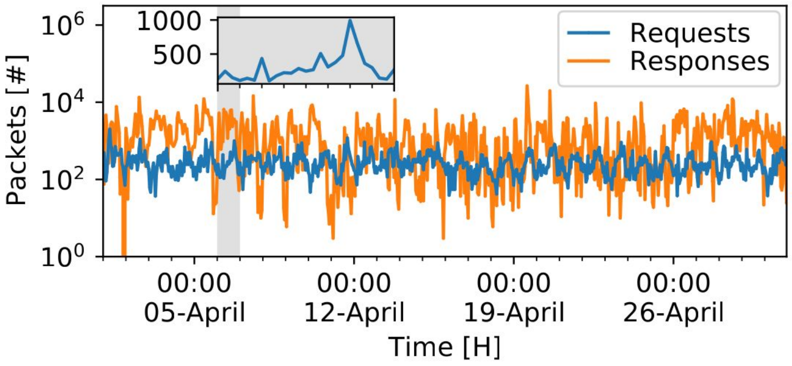
We observe scanners.

Scanners look for QUIC servers:

- Connection attempts to port 443 (requests)



We can group responses into attack sessions.

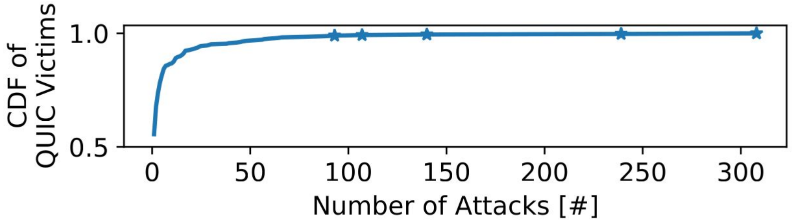


We group responses into attack sessions with the following thresholds:

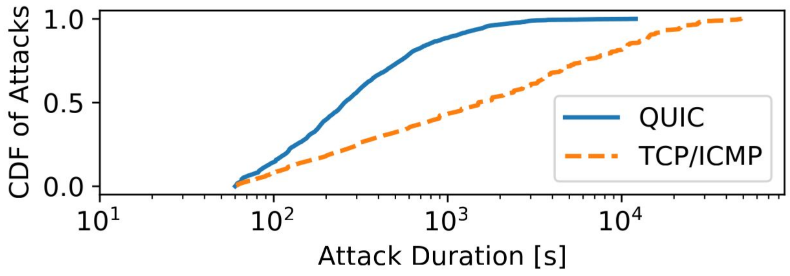
- more than 25 packets
- longer than 60s
- maximum packet rate > 0.5pps

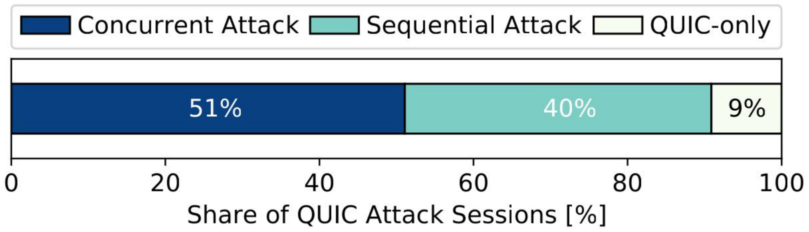
2905 attacks (394 IP addresses)

More than half are attacked only a single time.



We observe QUIC attacks in parallel with TCP/ICMP attacks.





QUIC floods are shorter than TCP/ICMP attacks.

Most of the time a server is attacked using multiple protocols.

Hypergiants are the main targets (Google, Facebook, …).

Responsiveness of webservers is impacted by requests.

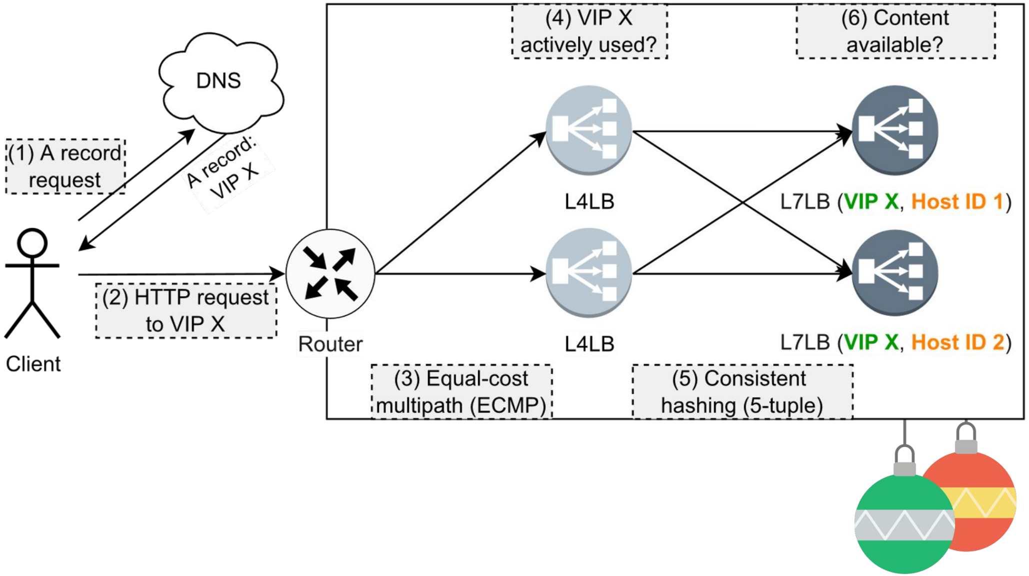
Attack	NGINX Config		Results			
Volume [pps]	QUIC Retry	Workers [#]	Client [# Req]	Server [# Resp]	Service Available	Extra RTT
10	✗	4	3,001	12,004	100%	✗
100	✗	4	30,001	81,680	68%	✗
1,000	✗	4	300,001	81,680	7%	✗
1,000	✗	auto=128	300,001	1,200,004	100%	✗
10,000	✗	auto=128	500,000	522,752	26%	✗
100,000	✗	auto=128	498,991	322,158	26%	✗
1,000	✓	4	300,001	300,001	100%	✓
10,000	✓	4	500,000	500,000	100%	✓
100,000	✓	4	500,000	500,000	100%	✓

Observed in backscatter

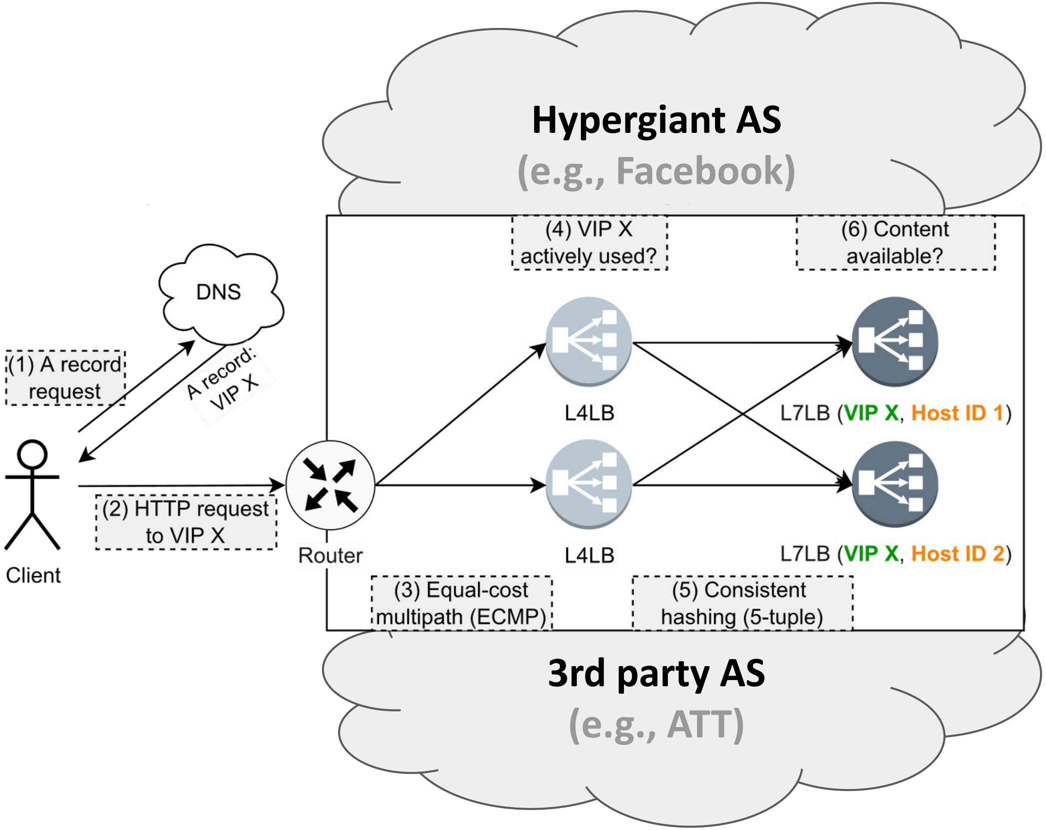
We just analyzed attacks. But can we also use backscatter to learn more about hypergiants?

We create fingerprints and use encoded information.

Attack targets are mainly hypergiants.



Attack targets are mainly hypergiants.



On-net deployment

Off-net deployment

Merging multiple QUIC packets into a single UDP datagram

	Packets from source network [%]			
QUIC packet type	Cloudflare	Facebook	Google	Remaining
Initial	56.029	47.695	23.239	46.960
Handshake	40.682	52.305	23.742	43.767
0-RTT	0.000	0.000	0.289	0.187
Retry	0.000	0.000	0.000	0.003
Coalescing packets				
Initial, Handshake	3.289	0.000	52.730	9.081
Handshake, Initial	0.000	0.000	0.000	0.001

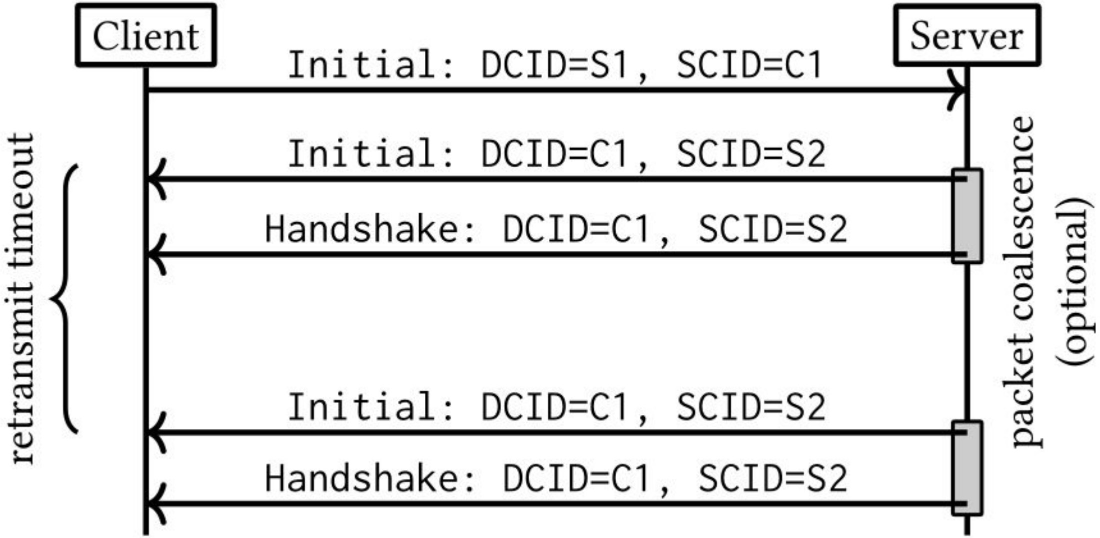
Merging multiple QUIC packets into a single UDP datagram

	Packets from source network [%]			
QUIC packet type	Cloudflare	Facebook	Google	Remaining

Cloudflare and Google enable packet coalescing.
Facebook does not.

Initial, Handshake	3.289	0.000	52.730	9.081
Handshake, Initial	0.000	0.000	0.000	0.001

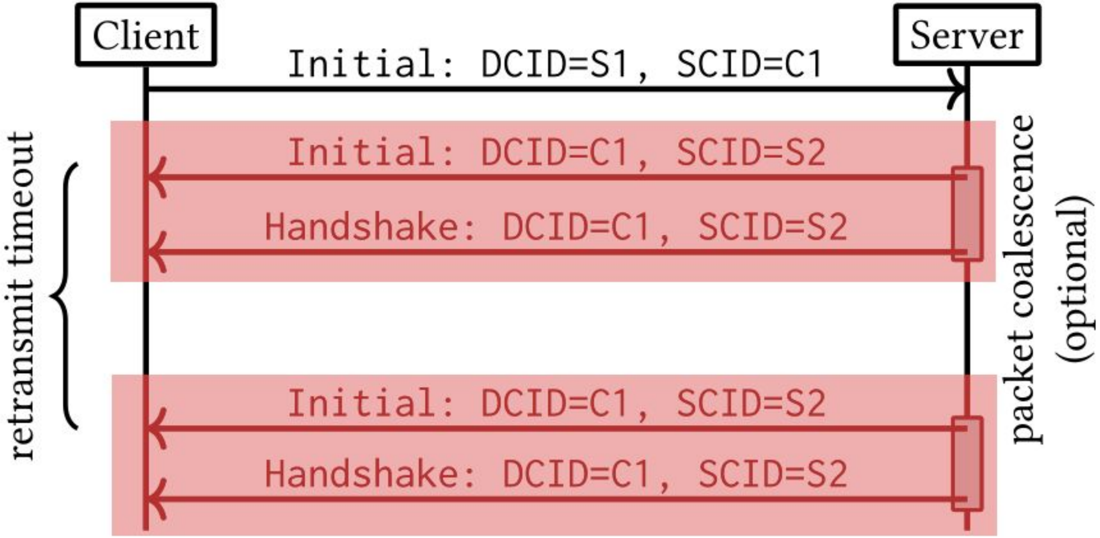
Incomplete handshakes cause resends



QUIC connections are identified by **connection IDs** and not ports.

Attackers can only perform incomplete handshakes, since information from the server response is required to complete the handshake.

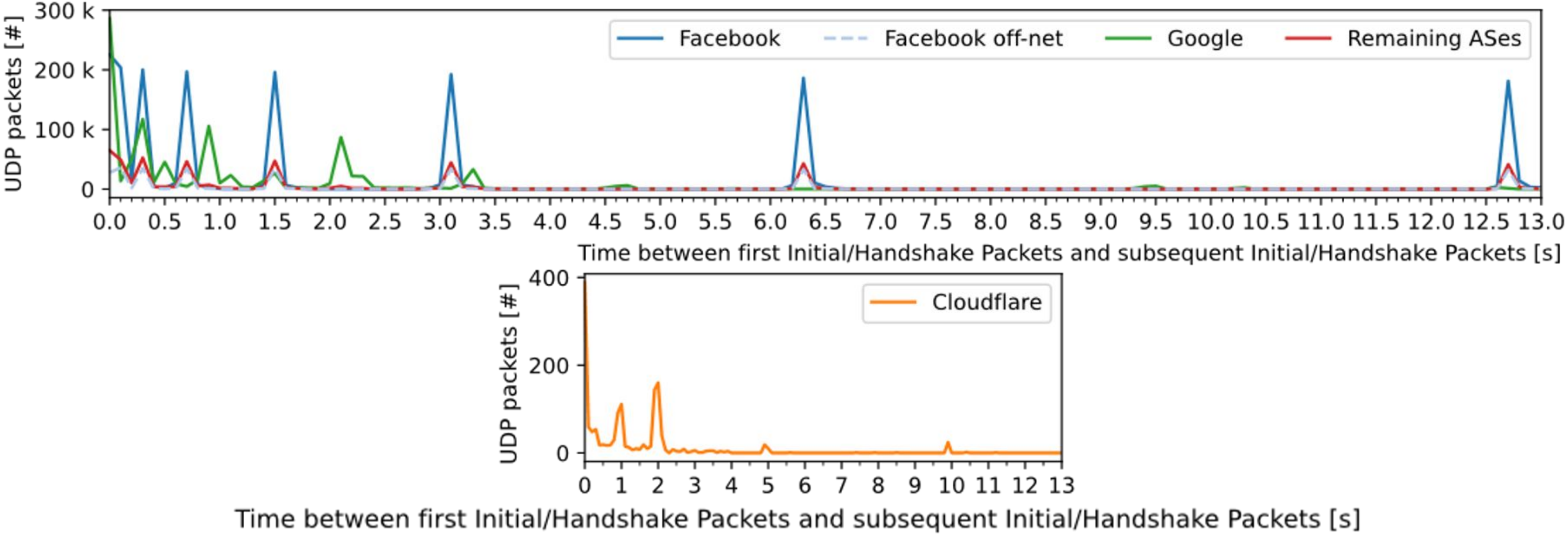
Incomplete handshakes cause resends



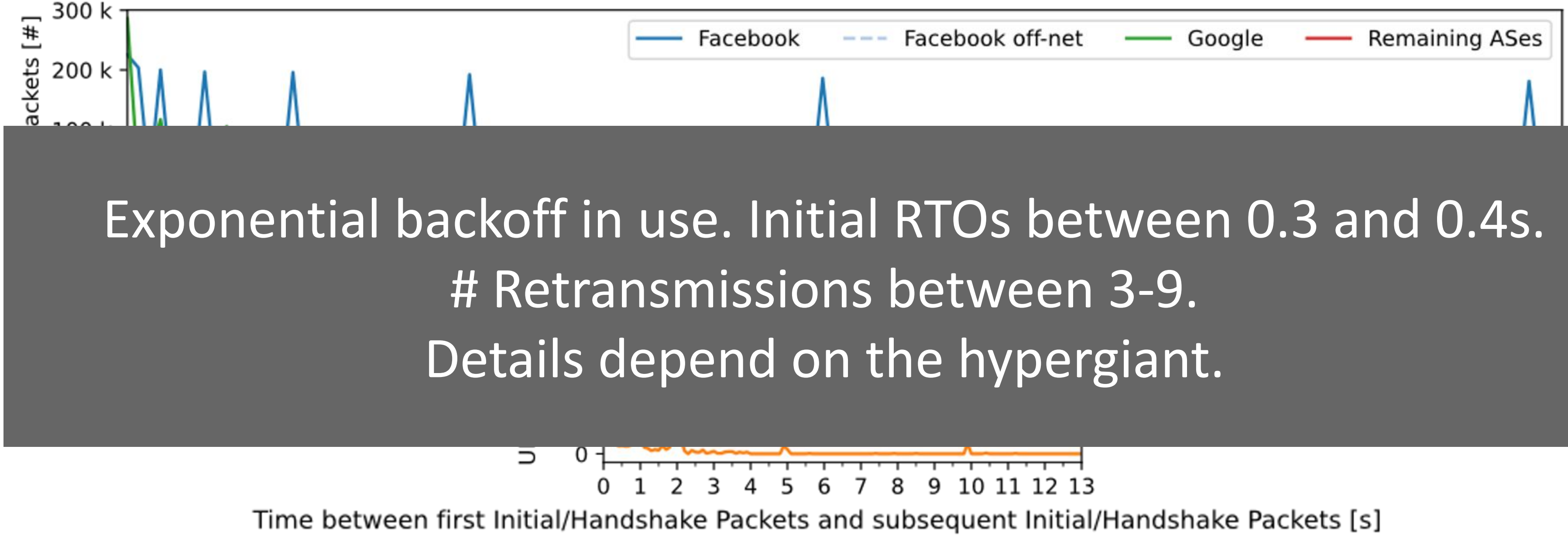
QUIC connections are identified by **connection IDs** and not ports.

Attackers can only perform incomplete handshakes, since information from the server response is required to complete the handshake.

Inter-arrival times of incomplete Handshakes



Inter-arrival times of incomplete Handshakes



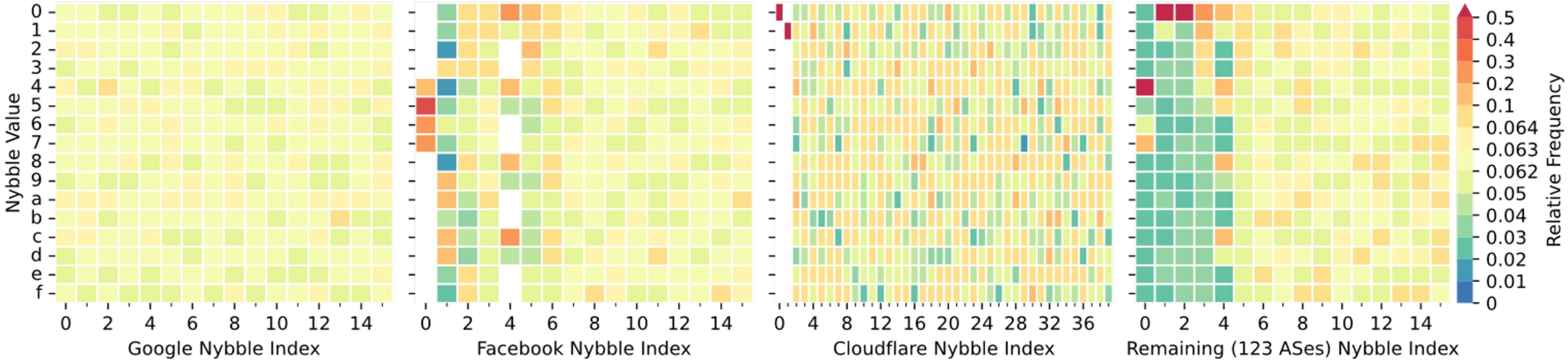
Exponential backoff in use. Initial RTOs between 0.3 and 0.4s.
Retransmissions between 3-9.
Details depend on the hypergiant.

Structure of QUIC Server Connection IDs (SCIDs)

XXXXXXX…XXXXXXXXX max. length 20 Byte

(half Byte, Nybble) 0…f

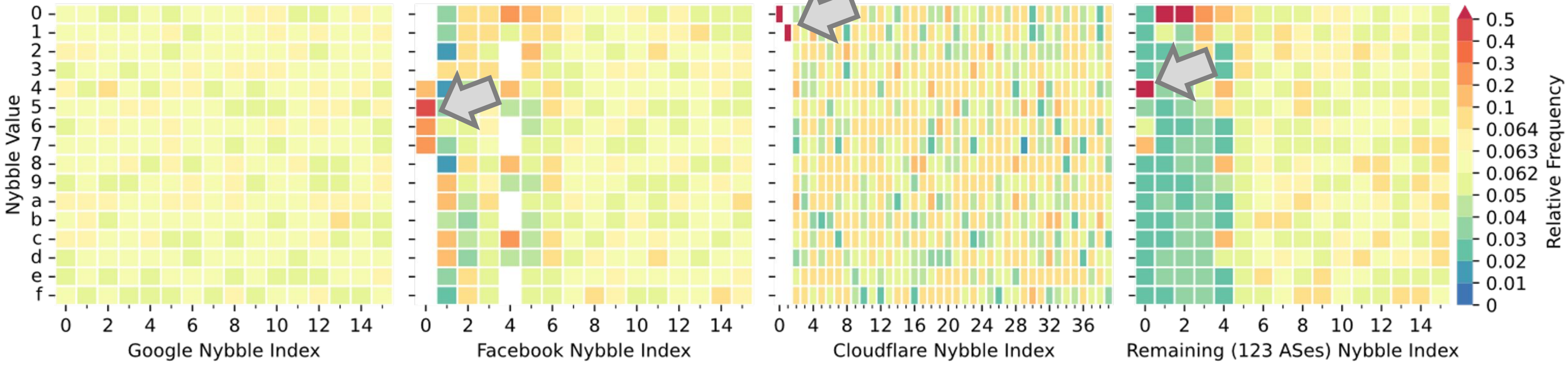
Structure of QUIC Server Connection IDs (SCIDs)



XXXXXXX…XXXXXXXXX max. length 20 Byte

(half Byte, Nybble) 0…f

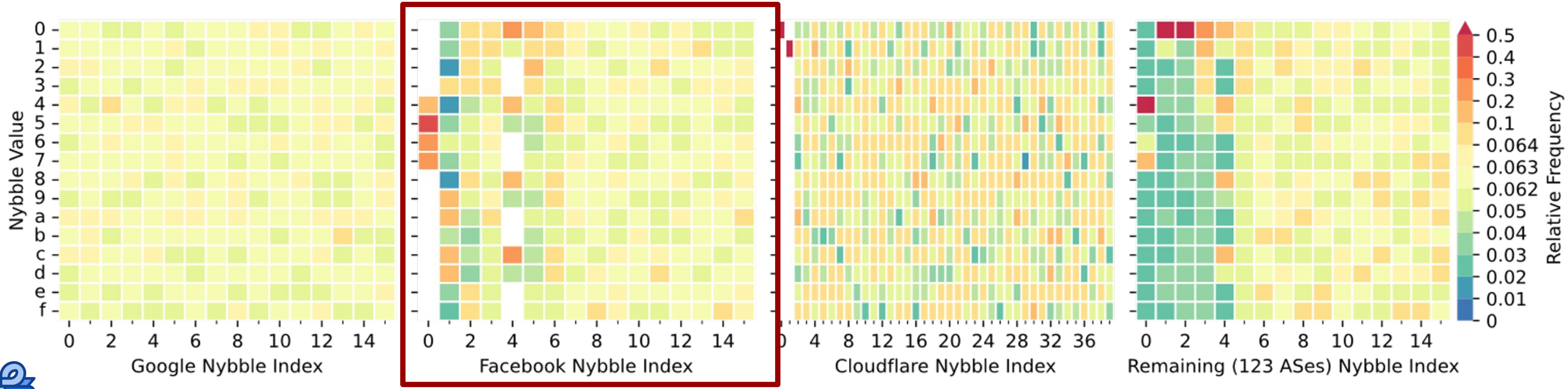
Structure of QUIC Server Connection IDs (SCIDs)



XXXXXXX...XXXXXXXXX max. length 20 Byte

(half Byte, Nybble) 0...f

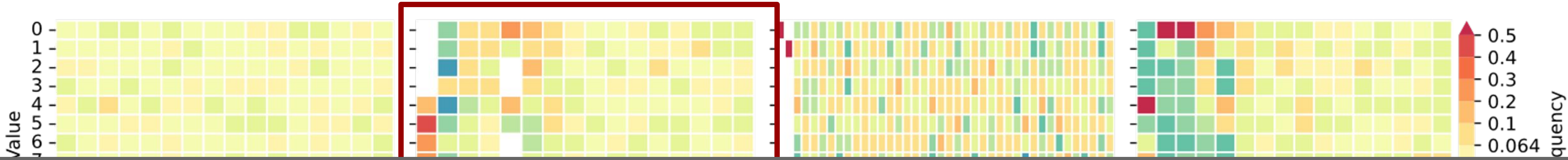
Structure of QUIC Server Connection IDs (SCIDs)



	Bits of the SCID				
SCID Version	Version	Host ID	Worker ID	Process ID	Remaining (random)
1	0-1	2-17	18-25	26	27-63
2	0-1	8-31	32-39	40	2-7,41-63

Facebook's SCID Structure according to their QUIC Implementation mvfst.

Structure of QUIC Server Connection IDs (SCIDs)



Facebook and Cloudflare use structured Connection IDs.
Encoded information can be used to fingerprint HG deployments and for stateless load balancing.

SCID Version	Version	Host ID	Worker ID	Process ID	Remaining (random)
1	0-1	2-17	18-25	26	27-63
2	0-1	8-31	32-39	40	2-7,41-63

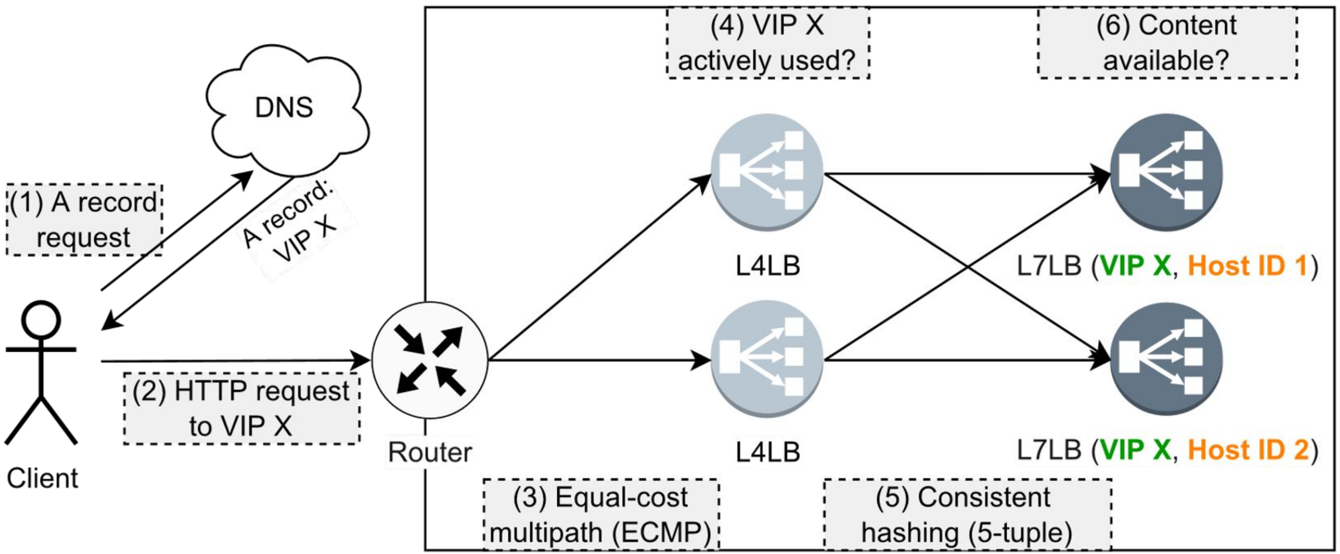
Facebook's SCID Structure according to their QUIC Implementation mvfst.

Detecting Facebook off-net servers

Verified by SAN in TLS certificates

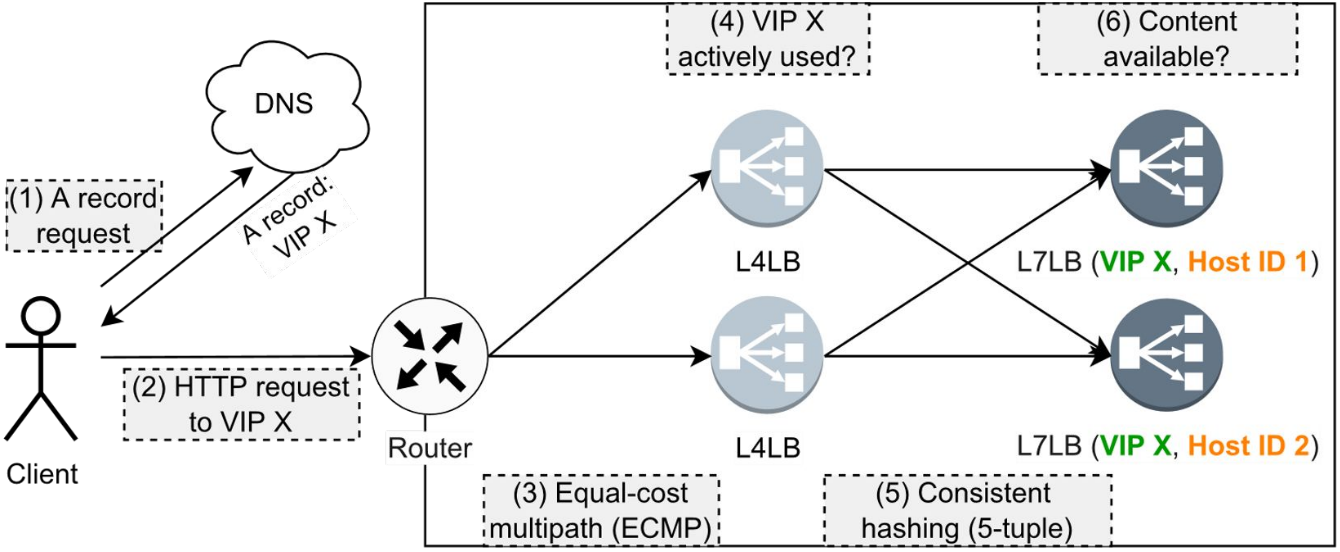
Classificator	TPR	FPR	TNR	FNR	Precision	Recall
Inter-Arrival Time (IAT)	0.772	0.268	0.732	0.228	0.645	0.772
SCID, IAT	0.772	0.046	0.954	0.228	0.914	0.772
Packet Length	0.997	0.328	0.672	0.003	0.657	0.997
Coalescence	1.000	0.931	0.069	0.000	0.403	1.000
SCID	**1.000**	**0.193**	**0.807**	**0.000**	**0.765**	**1.000**
SCID, Coalescence	**1.000**	**0.179**	**0.821**	**0.000**	**0.779**	**1.000**
SCID off-net	**1.000**	**0.027**	**0.973**	**0.000**	**0.959**	**1.000**

Facebook frontend cluster deployment



Facebook frontend cluster deployment

Method: Currently, using active QUIC measurements by probing 20,000 consecutive source ports to reach different L7LBs.



Clustering by shared host IDs

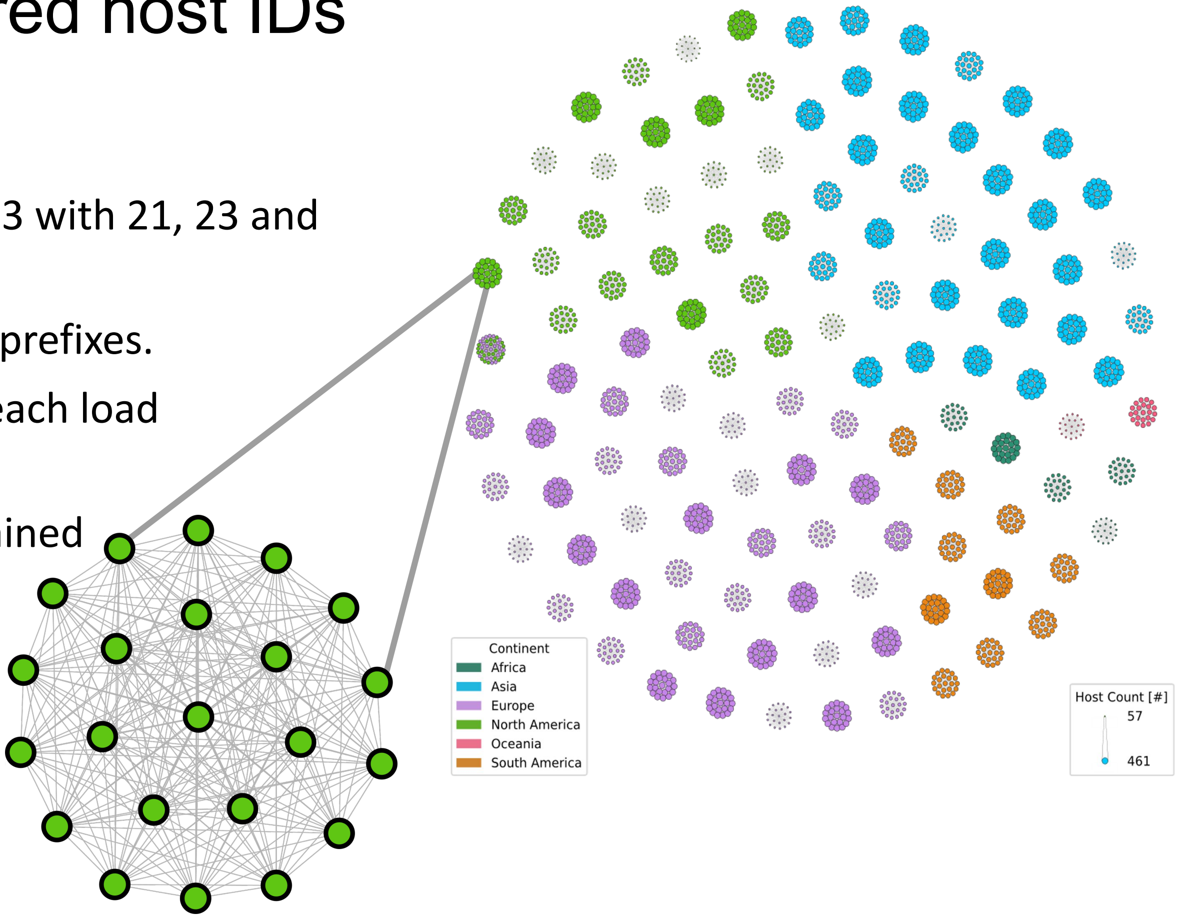
1 IP-Address = 1 node

112 clusters of 22 nodes and 3 with 21, 23 and 44 nodes.

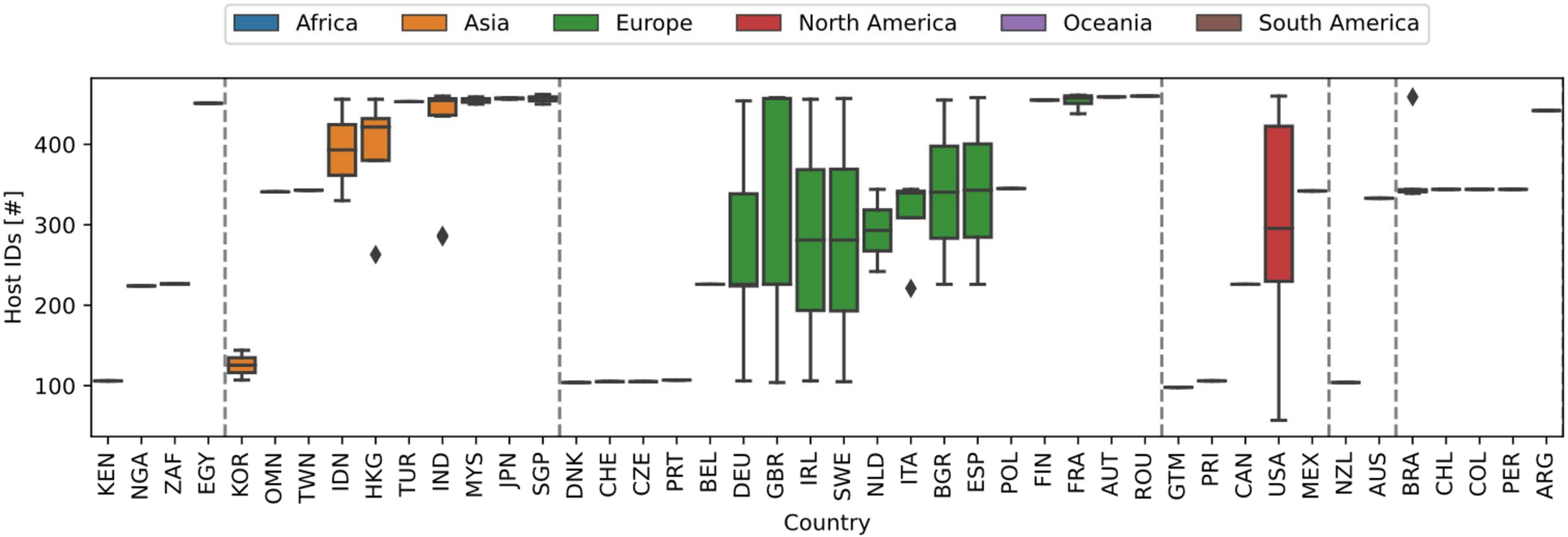
Clusters are organized in /24 prefixes.

Each IP address forwards to each load balancer/IP address.

19% of the host IDs are contained in IBR.



Facebook cluster sizes per country



Median cluster size in Asia 453 L7LBs compared to 339.5 (EU), 334 (NA), 292 (SA)

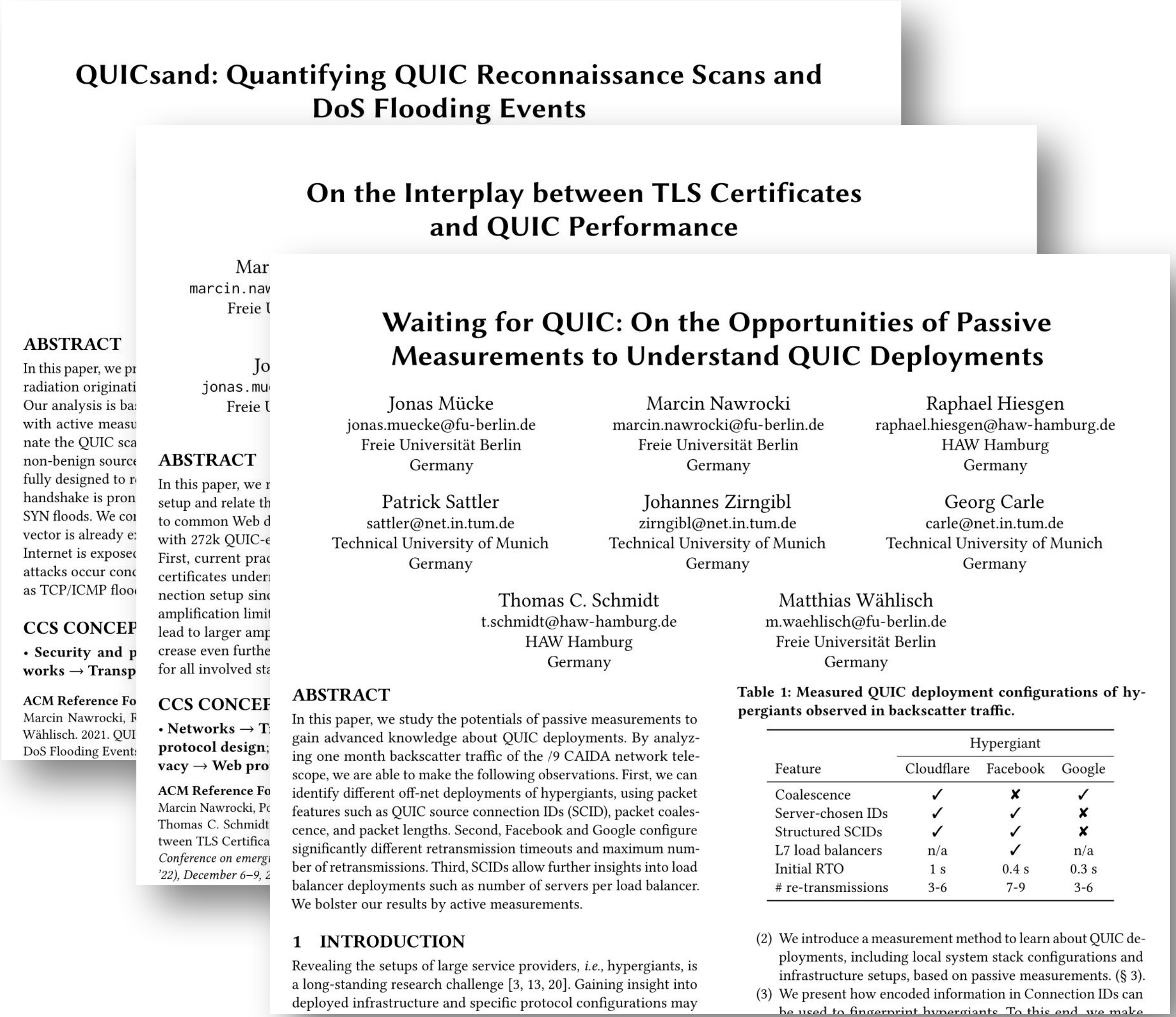
Conclusions

Certificate compression, signing algorithms and packet coalescence can improve the handshake.
QUIC can reduce latency. Inefficient handshakes (Multi-RTT) increase handshake duration.The anti-amplification-limit is often violated by implementations.

The RETRY option is an effective mitigation.
QUIC attacks happen and the found amplification factors compare to often used protocols.

Information encoding in connection IDs will be used for efficient stateless load balancing.
Passive measurements can be used for off-net detection. Server connection IDs allow detailed insights into server deployments.

More details



QUICsand: Quantifying QUIC Reconnaissance Scans and DoS Flooding Events

On the Interplay between TLS Certificates and QUIC Performance

Waiting for QUIC: On the Opportunities of Passive Measurements to Understand QUIC Deployments

Jonas Mücke
jonas.muecke@fu-berlin.de
Freie Universität Berlin
Germany

Marcin Nawrocki
marcin.nawrocki@fu-berlin.de
Freie Universität Berlin
Germany

Raphael Hiesgen
raphael.hiesgen@haw-hamburg.de
HAW Hamburg
Germany

Patrick Sattler
sattler@net.in.tum.de
Technical University of Munich
Germany

Johannes Zirngibl
zirngibl@net.in.tum.de
Technical University of Munich
Germany

Georg Carle
carle@net.in.tum.de
Technical University of Munich
Germany

Thomas C. Schmidt
t.schmidt@haw-hamburg.de
HAW Hamburg
Germany

Matthias Wählisch
m.waehlisch@fu-berlin.de
Freie Universität Berlin
Germany

ABSTRACT

In this paper, we study the potentials of passive measurements to gain advanced knowledge about QUIC deployments. By analyzing one month backscatter traffic of the /9 CAIDA network telescope, we are able to make the following observations. First, we can identify different off-net deployments of hypergiants, using packet features such as QUIC source connection IDs (SCID), packet coalescence, and packet lengths. Second, Facebook and Google configure significantly different retransmission timeouts and maximum number of retransmissions. Third, SCIDs allow further insights into load balancer deployments such as number of servers per load balancer. We bolster our results by active measurements.

1 INTRODUCTION

Revealing the setups of large service providers, *i.e.*, hypergiants, is a long-standing research challenge [3, 13, 20]. Gaining insight into deployed infrastructure and specific protocol configurations may

Table 1: Measured QUIC deployment configurations of hypergiants observed in backscatter traffic.

	Hypergiant		
Feature	Cloudflare	Facebook	Google
Coalescence	✓	✘	✓
Server-chosen IDs	✓	✓	✘
Structured SCIDs	✓	✓	✘
L7 load balancers	n/a	✓	n/a
Initial RTO	1 s	0.4 s	0.3 s
# re-transmissions	3-6	7-9	3-6

(2) We introduce a measurement method to learn about QUIC deployments, including local system stack configurations and infrastructure setups, based on passive measurements. (§ 3).
(3) We present how encoded information in Connection IDs can be used to fingerprint hypergiants. To this end, we make

Backup Slides

SCID structure of Facebook off-net servers

	CDN		
Feature	Cloudflare	Facebook	Google
Coalescence	✔	✘	✔
Server-chosen IDs	✔	✔	✘
SCID length [B]	20	8	8
Structured SCIDs	✔	✔	✘
L7 Load balancers	n/a	✔	n/a
Initial RTOs	1s	0.4s	0.3s
# re-transmissions	3-6	7-9	3-6

Which load balancing method is used?

Packets received that are inconsistent with an existing connection must be dropped

CID-aware Load Balancing:

1. Connect to IP1 with a server connection ID S1.
2. Connect to IP1 with server connection ID S1 but from a different 5-tuple at 1s intervals.

If 2. fails we learn that the connection ID S1 is used to forward the request. This is the expected behavior of QUIC servers.

5-tuple Load Balancing:

1. Connect to IP1 and record server connection ID S2
2. Connect to IP1 from a different 5-tuple with the same server connection ID S2.

If 2. fails we analyze additional information available in S2.

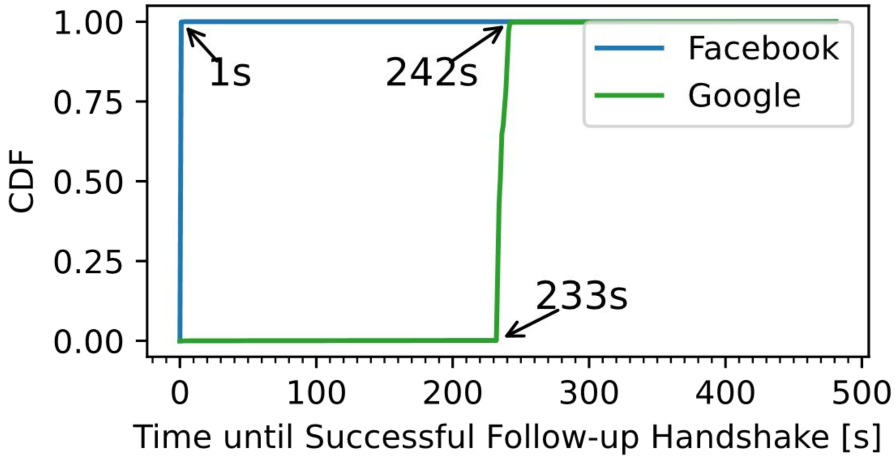
Facebook and Google use different load balancing methods

Google uses CID-aware load balancing.

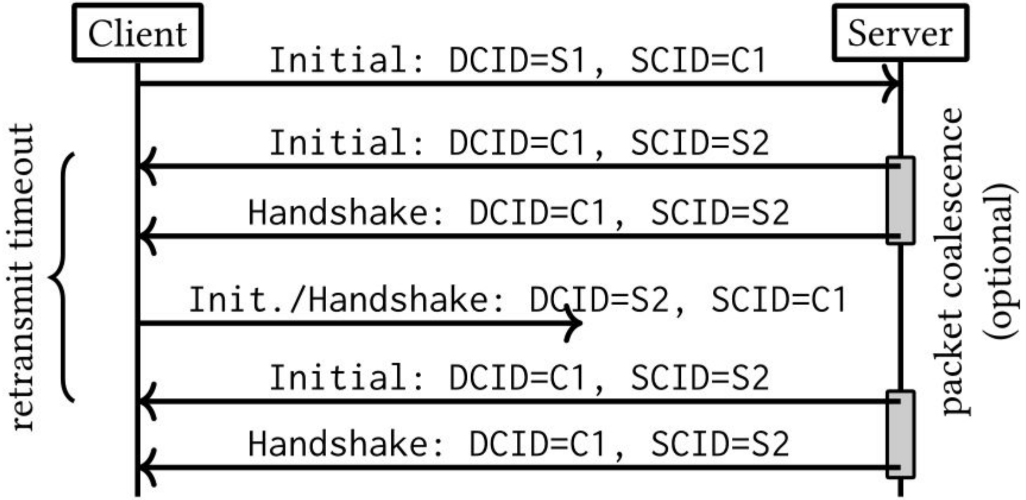
Facebook allows reconnection with client-chosen server connection ID because it uses server-chosen connection IDs.

Facebook uses 5-tuple routing.

Subsequent connections fail if the same host and worker ID are reached.



How does the Handshake look like?



Connections are identified by **connection IDs**, not ports. The underlying ports might change during connection.

The TLS certificate is included in the handshake message from the server.

Facebook frontend clusters: Load balancer fairness

Nearly equal Distribution of Traffic to Host IDs per Cluster.