



# A QUICker Internet?

On pitfalls, attacks, and discovering hypergiant infrastructures with QUIC

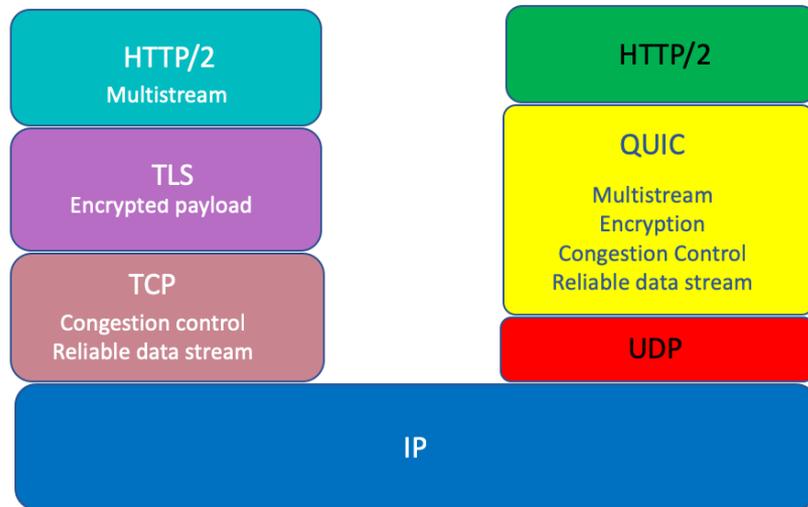
Jonas Mücke <[jonas.muecke@fu-berlin.de](mailto:jonas.muecke@fu-berlin.de)>





# 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

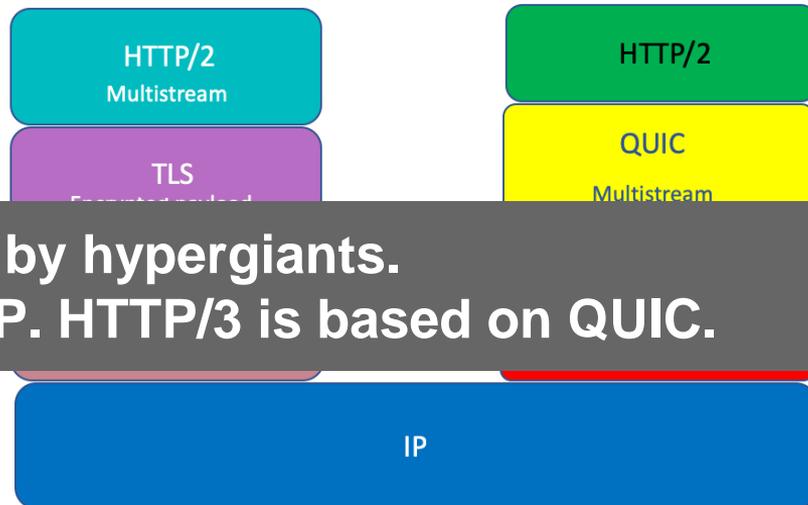


2020: 75 % of Facebook's Internet Traffic is QUIC.



# 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.

You access [www.youtube.com](https://www.youtube.com) to stream a christmas song

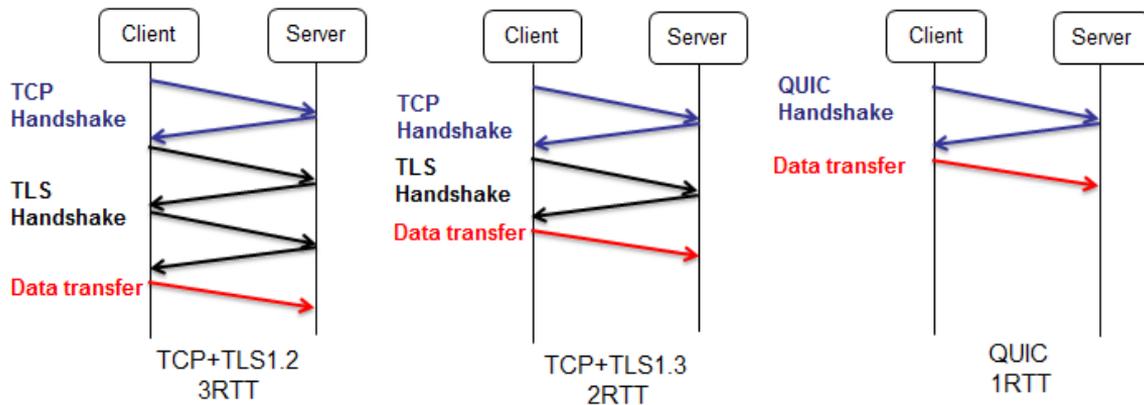
You want encryption, because you don't want your colleagues to know that you want to stream Last Christmas.

The screenshot shows the YouTube interface for the video 'Wham! - Last Christmas (Official 4K Video)'. The video player is the central focus, showing the band Wham! performing on a ship's deck. The video progress is at 0:27 / 4:38. Below the player, the video title and channel name 'Wham!' are visible, along with a 'Subscribe' button and engagement metrics (386K likes). To the right, a 'Show chat replay' button and a list of recommended videos are shown, including Mariah Carey's 'All I Want for Christmas' and George Michael's 'Careless Whisper'.

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



# Why is QUIC faster?



The TLS-handshake is embedded into the QUIC-handshake.

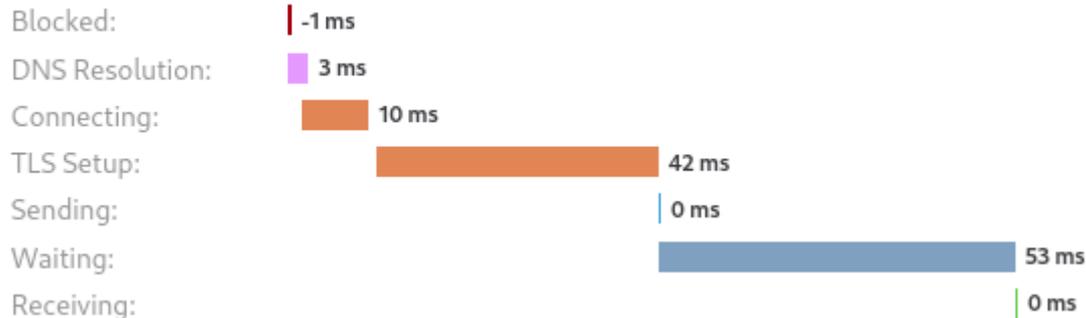
QUIC-Handshake = TCP-Handshake and TLS-Handshake

**Reduction of 1-2 RTTs.**



# How does it look in your browser?

## Request Timing



## 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.

The screenshot shows a YouTube video player for the song "Last Christmas" by Wham!. The video is paused at 0:27 / 4:38. The player interface includes a search bar, a sign-in button, and a "Show chat replay" button. Below the video, there are social media links, a subscribe button, and a comment section with 17,657 comments.

#EmiliaClarke #Wham #GeorgeMichael

### Wham! - Last Christmas (Official 4K Video)

Wham! 1.46M subscribers [Subscribe](#) [386K](#) [Share](#) [...](#)

25M views 2 years ago  
Wham! - Last Christmas | Stream Wham! here: <https://Wham.Ink.to/Playlists>  
Listen to all the Christmas Hits here: <https://christmashits.Ink.to/2022>  
Subscribe to the Wham! YouTube Channel: [https://Wham.Ink.to/YT\\_Subscribe](https://Wham.Ink.to/YT_Subscribe) [Show more](#)

17,657 Comments [Sort by](#)

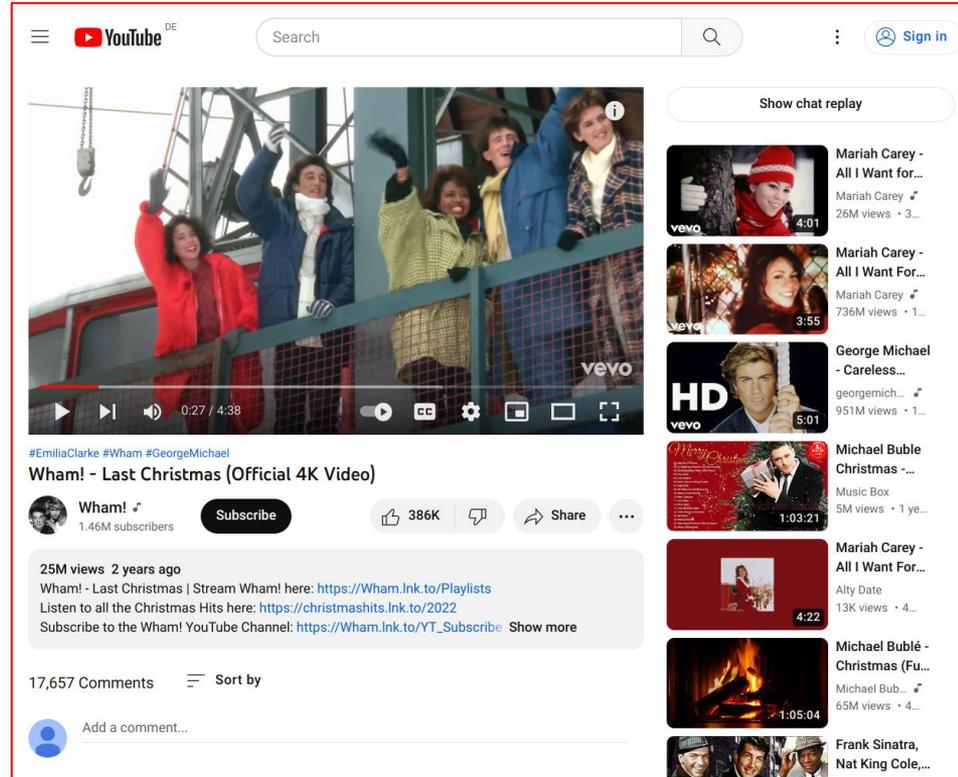
[Add a comment...](#)

Recommended videos:

- Mariah Carey - All I Want for Christmas (Official Music Video) 4:01 26M views · 3...
- Mariah Carey - All I Want For Christmas (Official Music Video) 3:55 736M views · 1...
- George Michael - Careless Whisper (Official Music Video) 5:01 951M views · 1...
- Michael Buble - Christmas (Official Music Video) 1:03:21 5M views · 1 ye...
- Mariah Carey - All I Want For Christmas (Official Music Video) 4:22 13K views · 4...
- Michael Bubl  - Christmas (Official Music Video) 1:05:04 65M views · 4...
- Frank Sinatra, Nat King Cole,...

# Loading content

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



The screenshot shows a YouTube video player for the song "Last Christmas" by Wham!. The video is in 4K quality and has 25M views. The player interface includes a search bar at the top, a "Show chat replay" button, and a list of recommended videos on the right. The recommended videos include "All I Want for Christmas" by Mariah Carey, "Careless Whisper" by George Michael, and "Christmas (Fu...)" by Michael Bublé. The video player itself shows a progress bar at 0:27 / 4:38 and a "Subscribe" button for the channel "Wham!".



html

# Loading content

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

#EmiliaClarke #Wham #GeorgeMichael  
**Wham! - Last Christmas (Official 4K Video)**

Wham! 1.46M subscribers

25M views 2 years ago  
Wham! - Last Christmas | Stream Wham! here: <https://Wham.Ink.to/Playlists>  
Listen to all the Christmas Hits here: <https://christmashits.Ink.to/2022>  
Subscribe to the Wham! YouTube Channel: [https://Wham.Ink.to/YT\\_Subscribe](https://Wham.Ink.to/YT_Subscribe) Show more

17,657 Comments Sort by

Add a comment...

Show chat replay

Mariah Carey - All I Want For...  
Mariah Carey 26M views · 3...

Mariah Carey - All I Want For...  
Mariah Carey 736M views · 1...

George Michael - Careless...  
georgemich... 951M views · 1...

Michael Buble Christmas -...  
Music Box 5M views · 1 ye...

Mariah Carey - All I Want For...  
Alty Date 13K views · 4...

Michael Bubl  - Christmas (Fu...  
Michael Bub... 65M views · 4...

Frank Sinatra, Nat King Cole,...



html



css

# Loading content

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

#EmiliaClarke #Wham #GeorgeMichael  
**Wham! - Last Christmas (Official 4K Video)**

Wham! 1.46M subscribers

25M views 2 years ago  
Wham! - Last Christmas | Stream Wham! here: <https://Wham.Ink.to/Playlists>  
Listen to all the Christmas Hits here: <https://christmashits.Ink.to/2022>  
Subscribe to the Wham! YouTube Channel: [https://Wham.Ink.to/YT\\_Subscribe](https://Wham.Ink.to/YT_Subscribe) Show more

17,657 Comments Sort by

Add a comment...

Recommended videos:

- Mariah Carey - All I Want for Christmas (Official Video) 26M views · 3...
- George Michael - Careless Whisper (Official Video) 951M views · 1...
- Michael Buble - Christmas (Official Video) 5M views · 1 ye...
- Mariah Carey - All I Want For Christmas (Official Video) 13K views · 4...
- Michael Bubl  - Christmas (Fu... 65M views · 4...
- Frank Sinatra, Nat King Cole,...



html



css



images

# Loading content

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

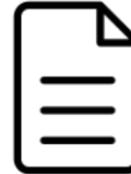
The screenshot shows a YouTube interface. At the top is the YouTube logo and a search bar. The main video player displays a scene from the 'Last Christmas' music video. Below the player, the video title 'Wham! - Last Christmas (Official 4K Video)' is shown, along with the channel name 'Wham!' and its subscriber count. The video description contains promotional links. To the right, a vertical list of recommended videos is visible, featuring artists like Mariah Carey and George Michael.



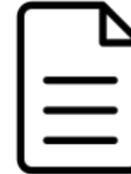
html



css



images



js

# Loading content

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

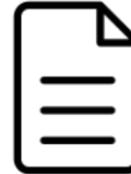
The screenshot shows a YouTube video player for the song "Last Christmas" by Wham!. The video is in 4K resolution and is 4:38 minutes long. The channel has 1.46M subscribers and the video has 25M views. The interface includes a search bar, a "Show chat replay" button, a list of recommended videos on the right, and a comment section at the bottom.



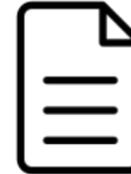
html



css



images



js



video13

# 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).

The screenshot shows a YouTube video player interface. At the top, there is a search bar and a 'Sign in' button. The main video player displays a scene from the 'Wham! - Last Christmas' music video. Below the video, there are social media links for #EmiliaClarke, #Wham, and #GeorgeMichael. The video title is 'Wham! - Last Christmas (Official 4K Video)'. The channel name is 'Wham!' with 1.46M subscribers. There is a 'Subscribe' button and a '386K' likes count. Below the video, there is a description box with 25M views and 2 years ago. The description includes links to stream Wham! and listen to Christmas hits. There is also a comment section with 17,657 comments and a 'Sort by' dropdown. The right sidebar shows a list of recommended videos, including 'Mariah Carey - All I Want For Christmas', 'George Michael - Careless Whisper', 'Michael Buble - Christmas', and 'Frank Sinatra, Nat King Cole, ...'.



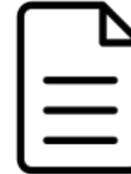
html



css



images

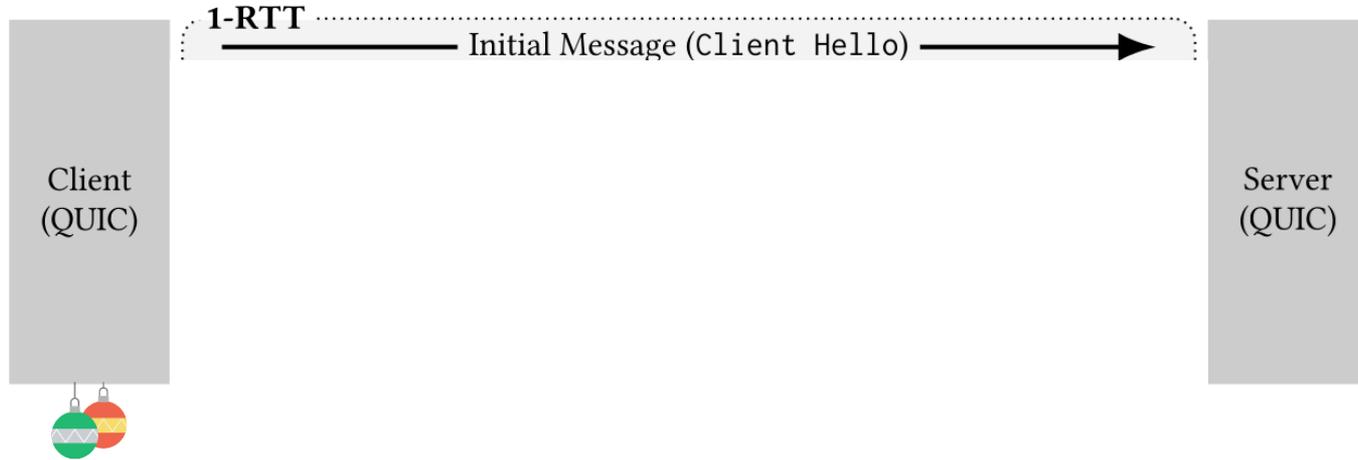


js



video14

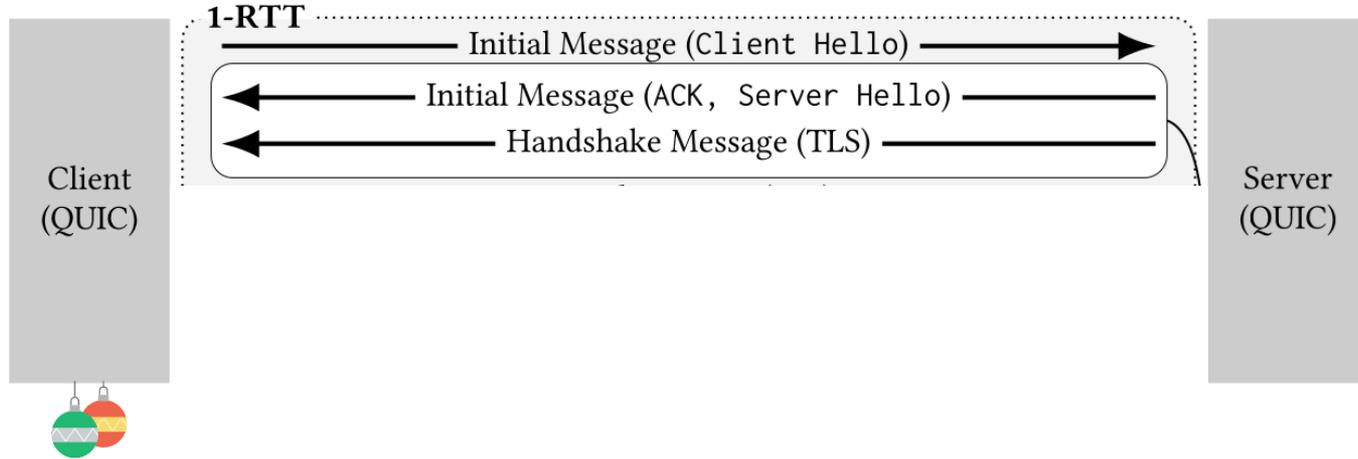
# How does the Handshake look like?



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 the Handshake look like?

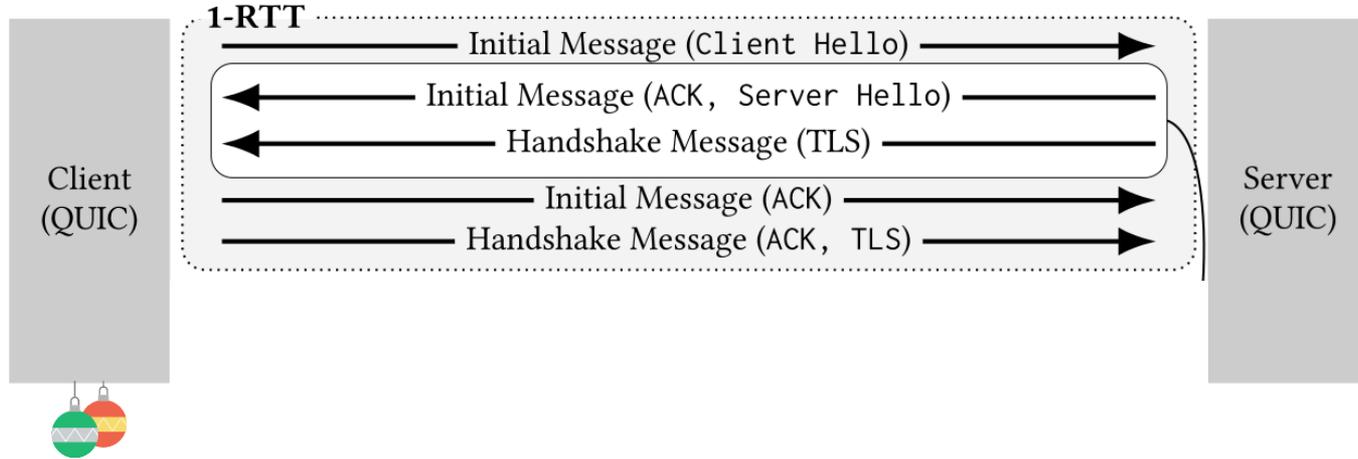


Before the  
of the UDP payload it received.

the size

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

# How does the Handshake look like?

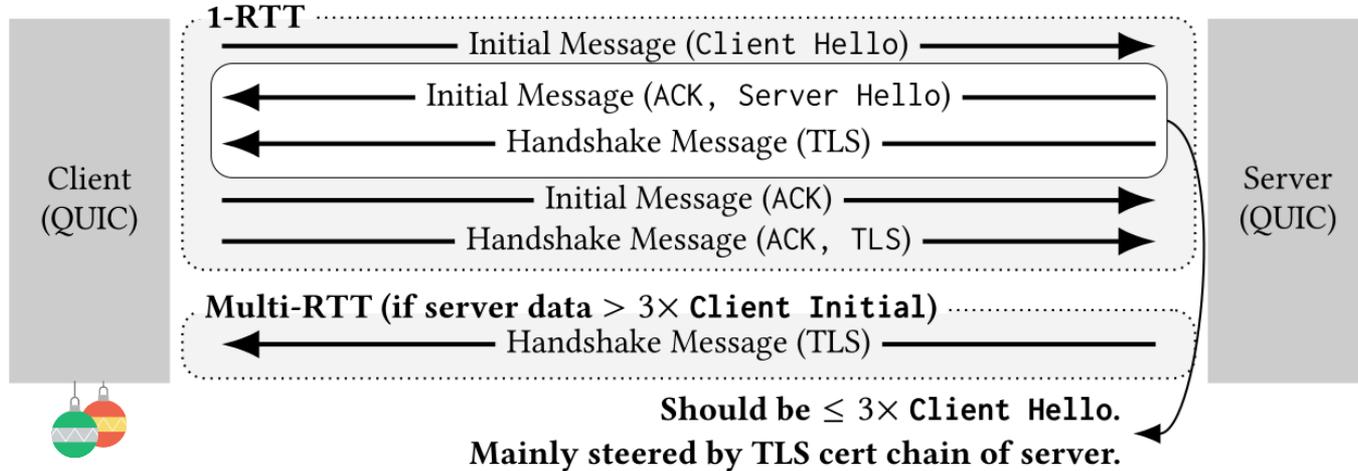


Before the  
of the UDP payload it received.

he size

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

# How does the Handshake look like?



Before the  
of the UDP payload it received.

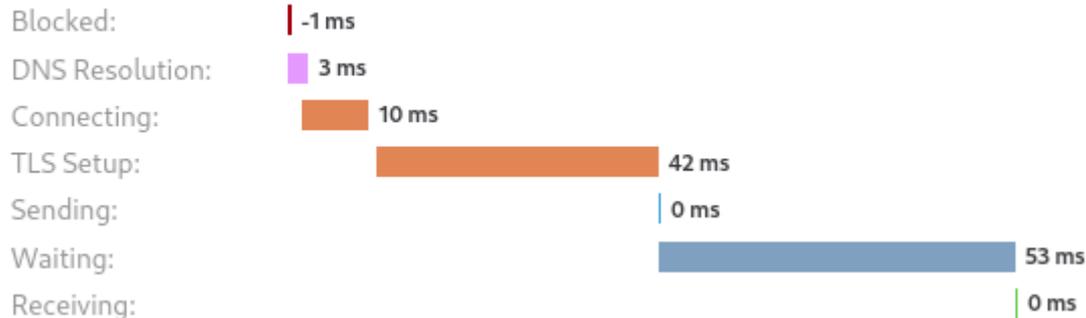
the size

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



# How does it look in your browser?

## Request Timing



## 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?

## Request Timing



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

QUIC-Handshake: 24ms

Total: 24ms (~54% reduction)



# The actual QUIC-handshake

## Request Timing

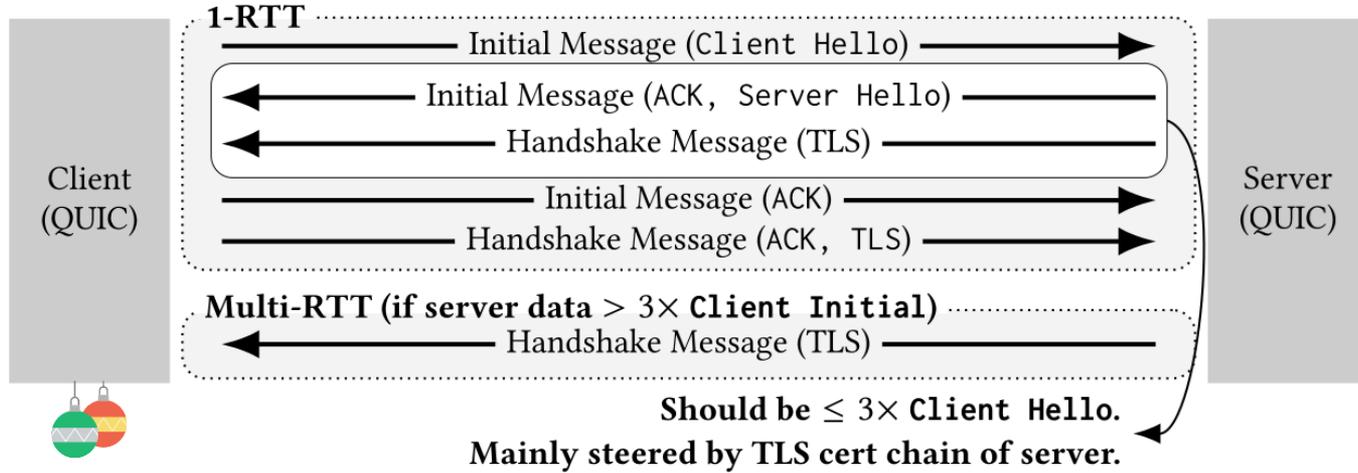


**www.youtube.com with Firefox and QUIC (HTTP/3, reality)**

QUIC-Handshake: 58ms

Total: 58ms (~11% increase)

# How does the Handshake look like?



Before the  
of the UDP payload it received.

the size

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

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

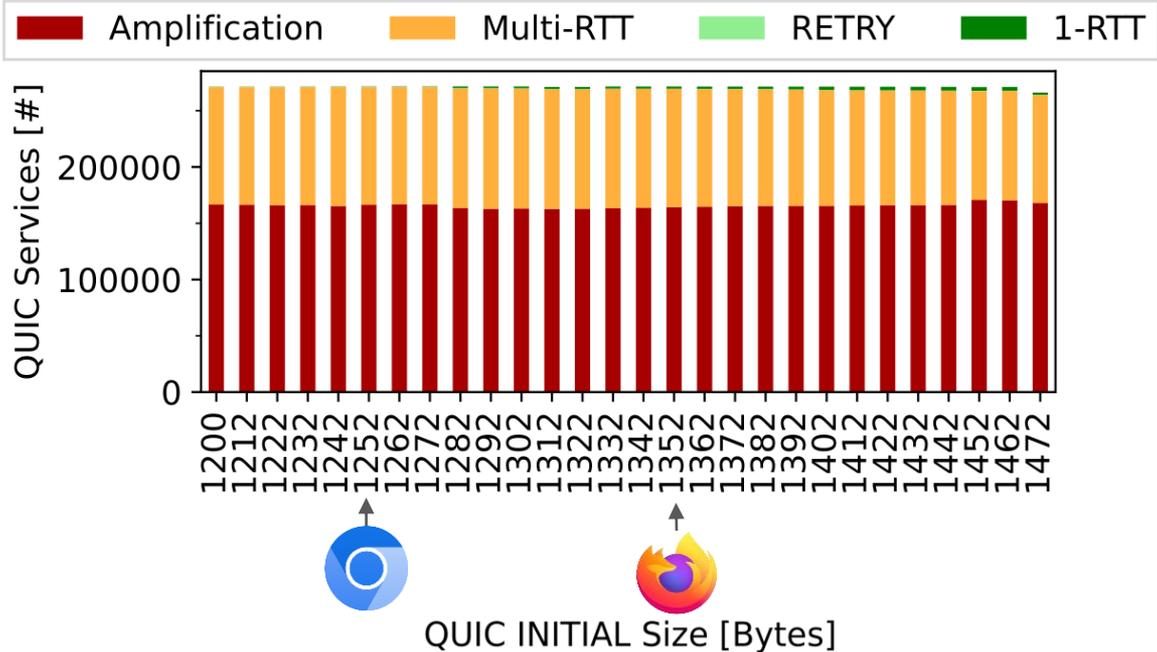
Measurement study on the Tranco Top 1M list.

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

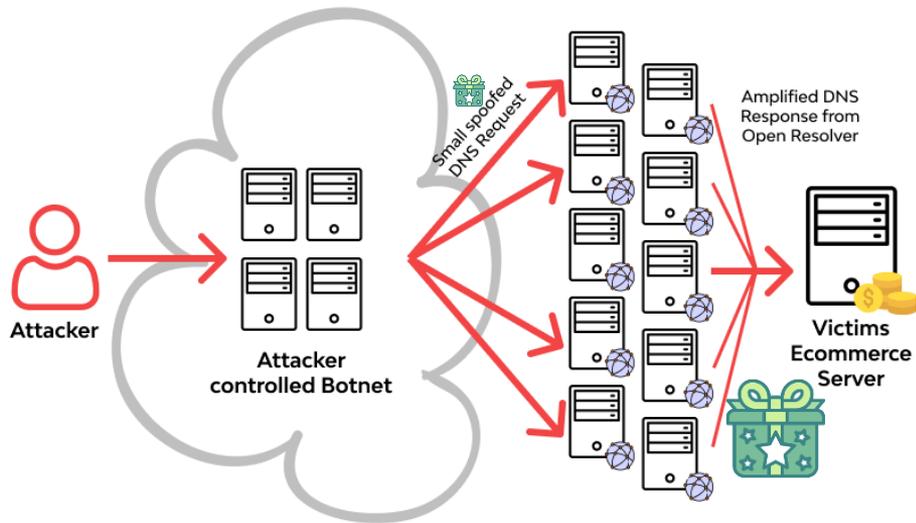
UDP operates connection-less.

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

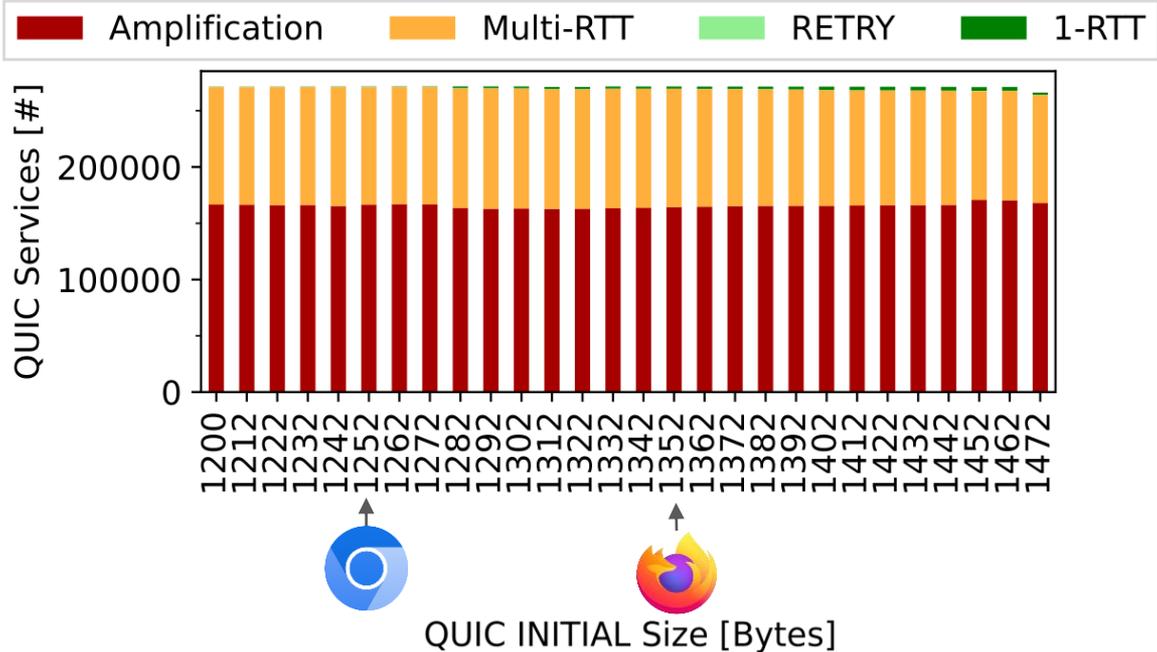
Typical amplification factor of DNS: 28X to 54X

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

QUIC should be unattractive for amplification attacks.



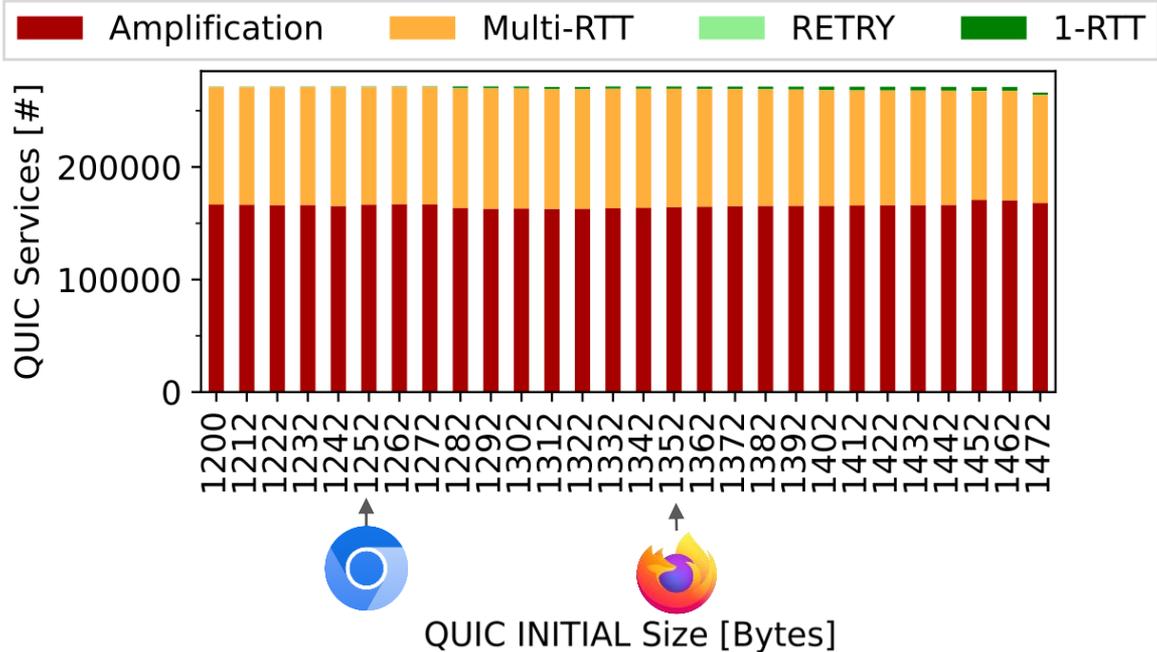
# How often do we encounter those cases?



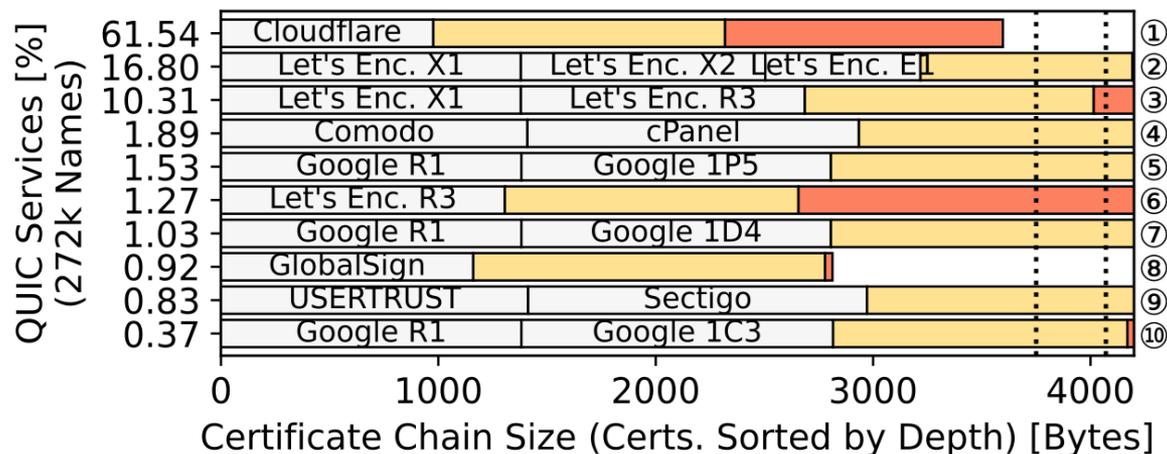
# The other two cases: RETRY and 1-RTT

- 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?



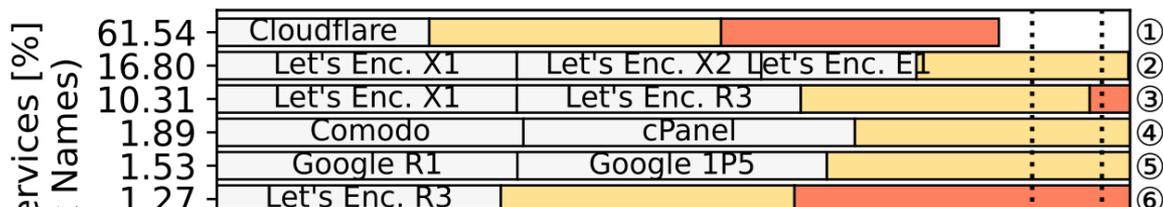
# What is the main reason?



Large certificate chains are the main reason.

Effective TLS setup influences the performance of the transport protocol.

# What is the main reason?



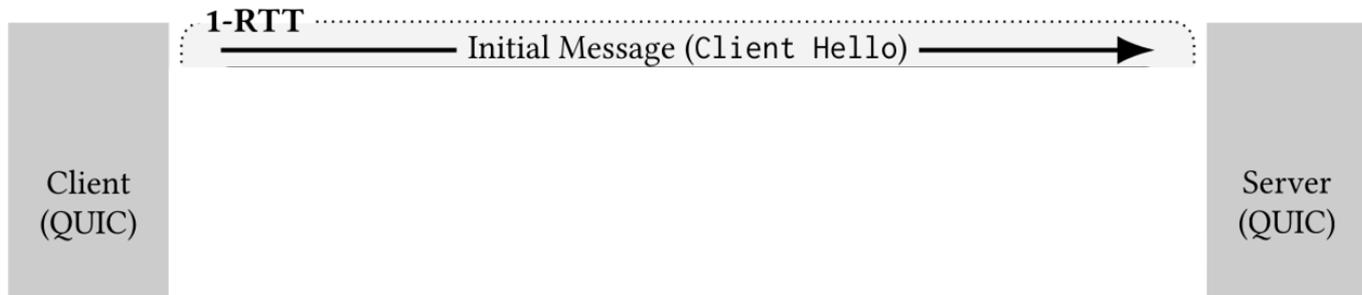
**Amplification during complete handshakes is common.  
Observed Amplification: up to 4.4X.**

0 1000 2000 3000 4000  
Certificate Chain Size (Certs. Sorted by Depth) [Bytes]

Large certificate chains are the main reason.

Effective TLS setup influences the performance of the transport protocol.

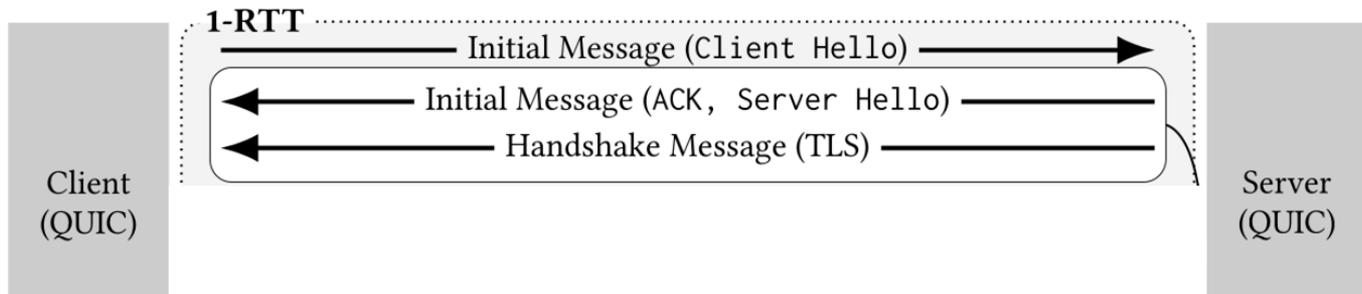
# What about incomplete handshakes?



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.

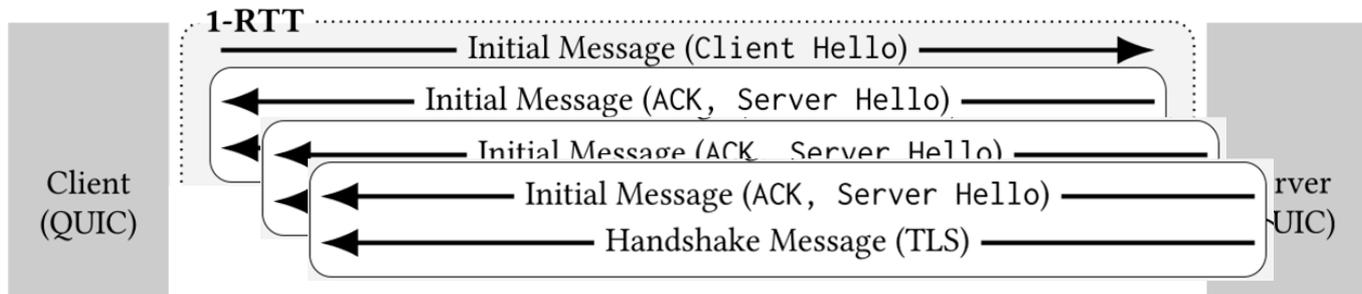
# What about incomplete handshakes?



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.

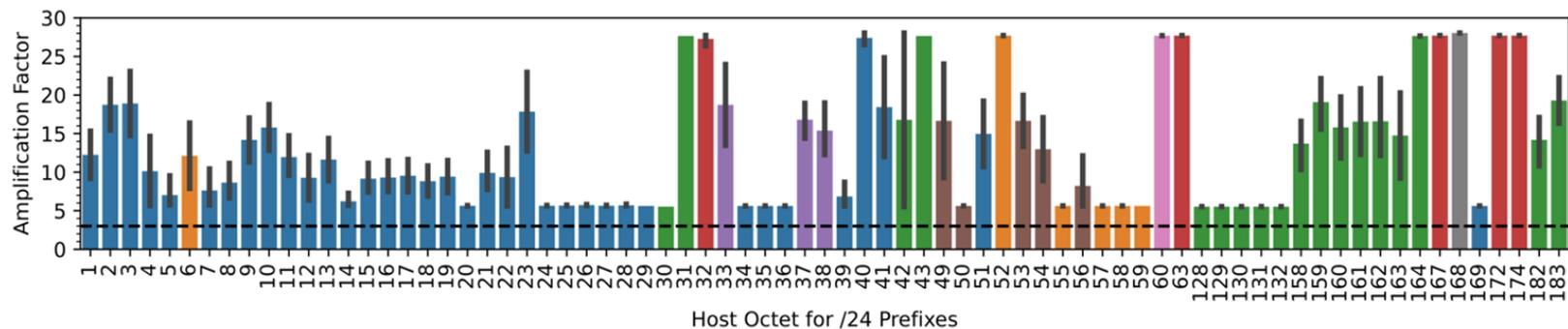
# What about incomplete handshakes?



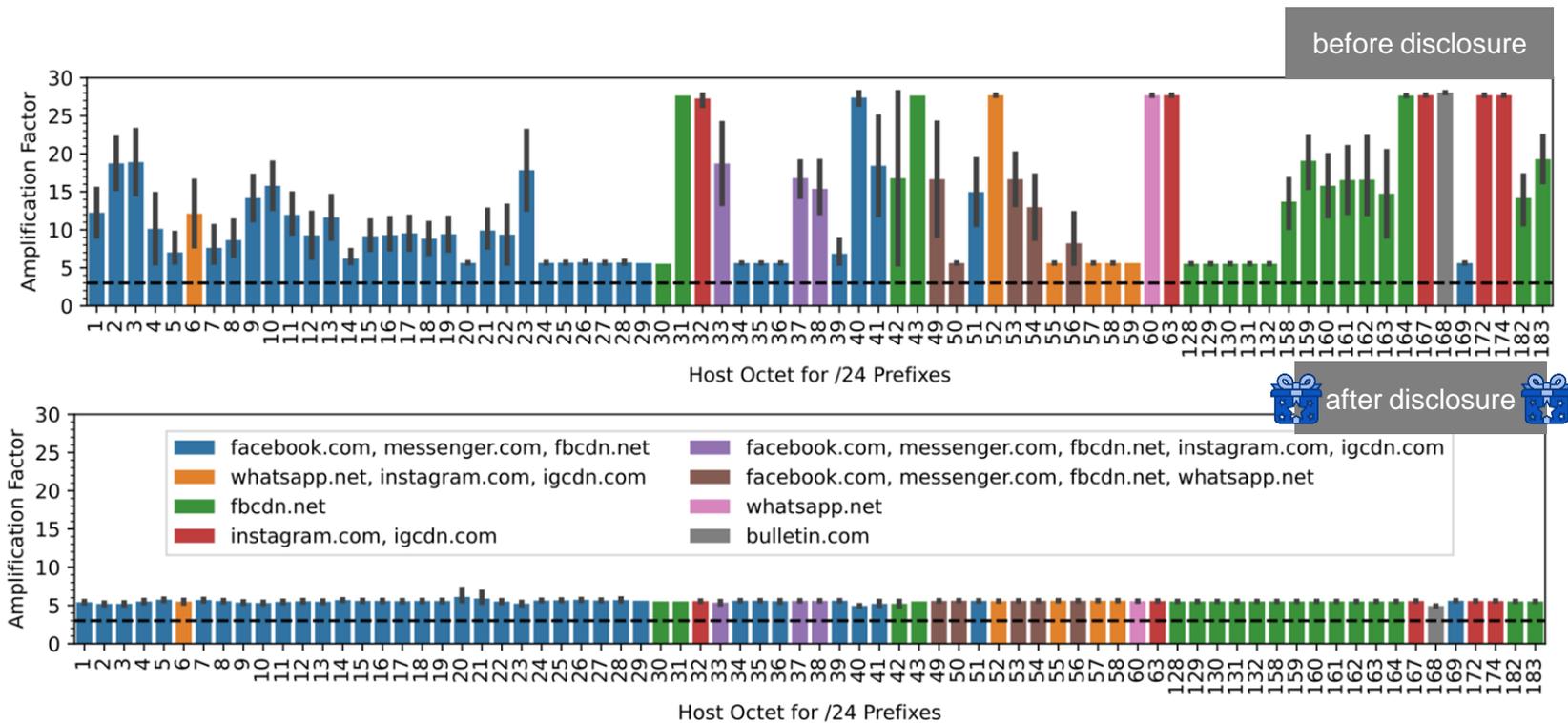
Send a client Initial packet to a server and collect response traffic, but do not send any other packets.

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.

[Analyze](#)

[Show advanced options](#)

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

### Client Initial 1250 Bytes (Chromium default)

<b>Multi-RTT</b> Handshake	<b>24.410ms</b> RTT	<b>3.3x</b> 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)

<b>Multi-RTT</b> Handshake	<b>24.081ms</b> RTT	<b>3.1x</b> 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 Compression algorithm not supported.	zstd Compression algorithm not supported.	<b>39.4 %</b> reduction
--	--	----------------------------

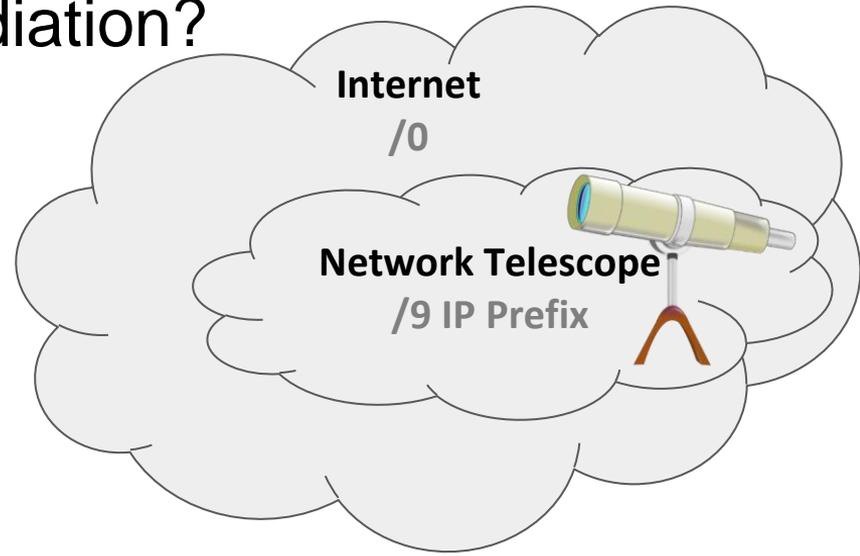


[understanding-quic.net](https://understanding-quic.net)

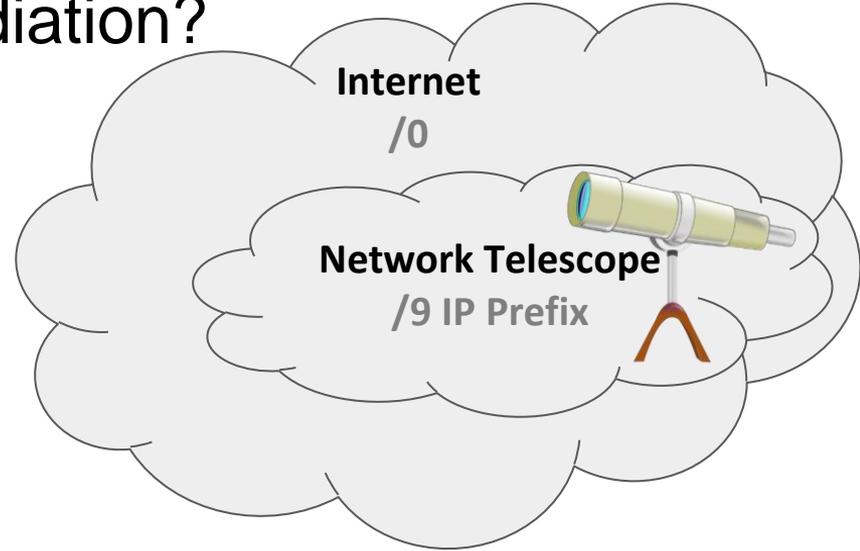
Active measurements are great. But what can we do with passive measurements?

We use Internet Background Radiation

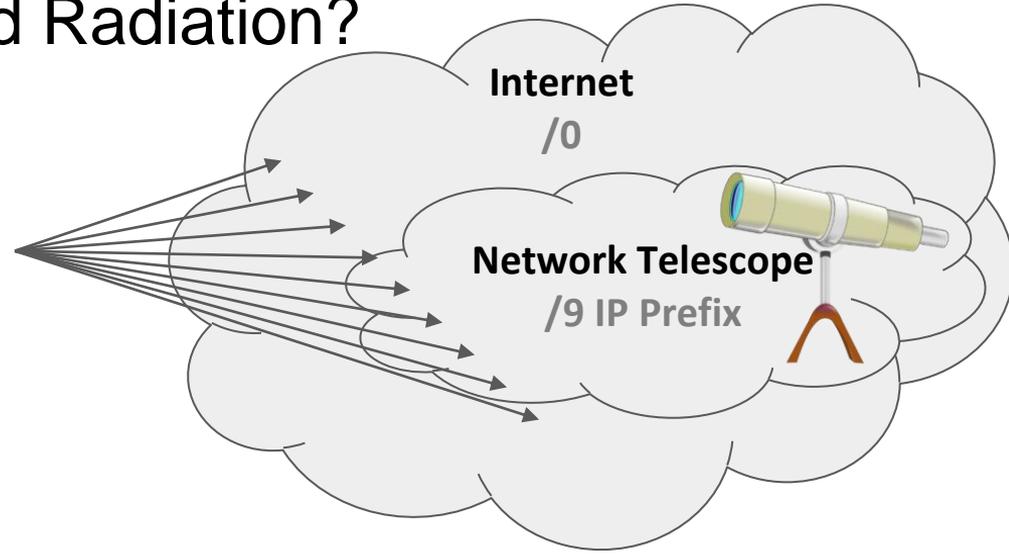
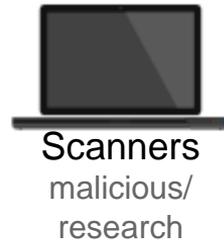
# What is Internet Background Radiation?



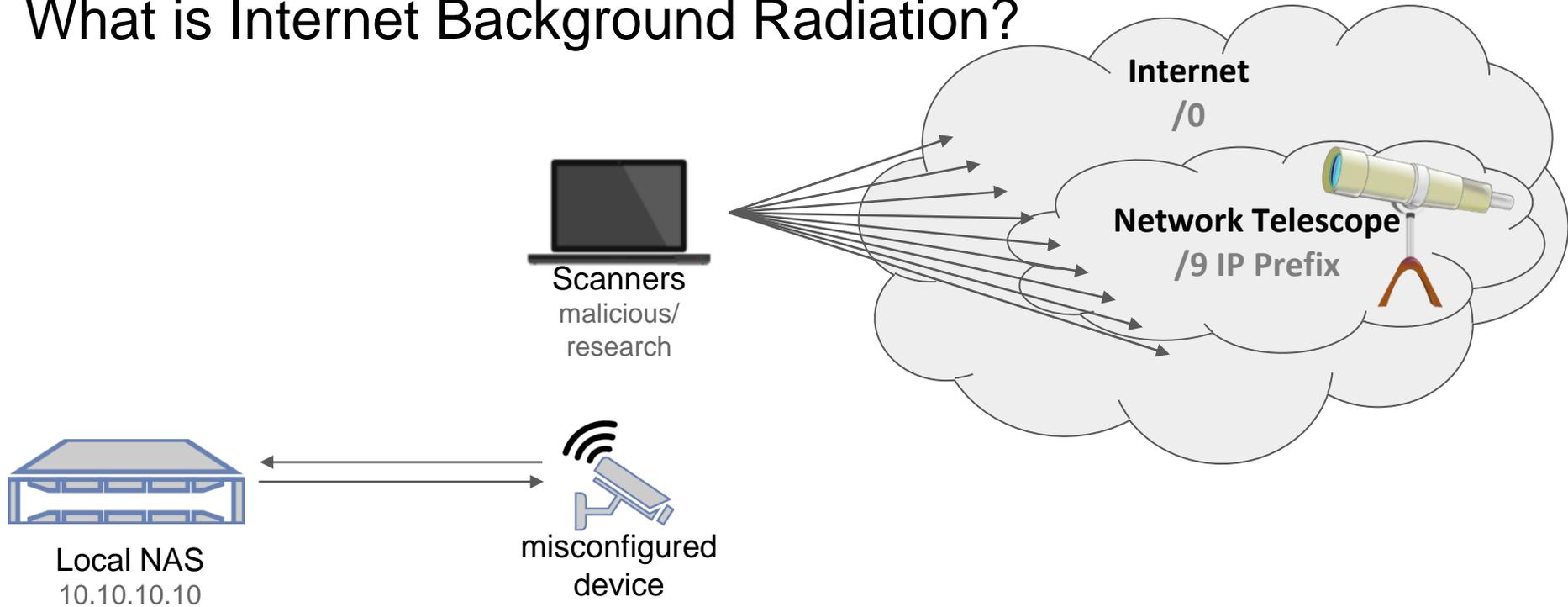
# What is Internet Background Radiation?



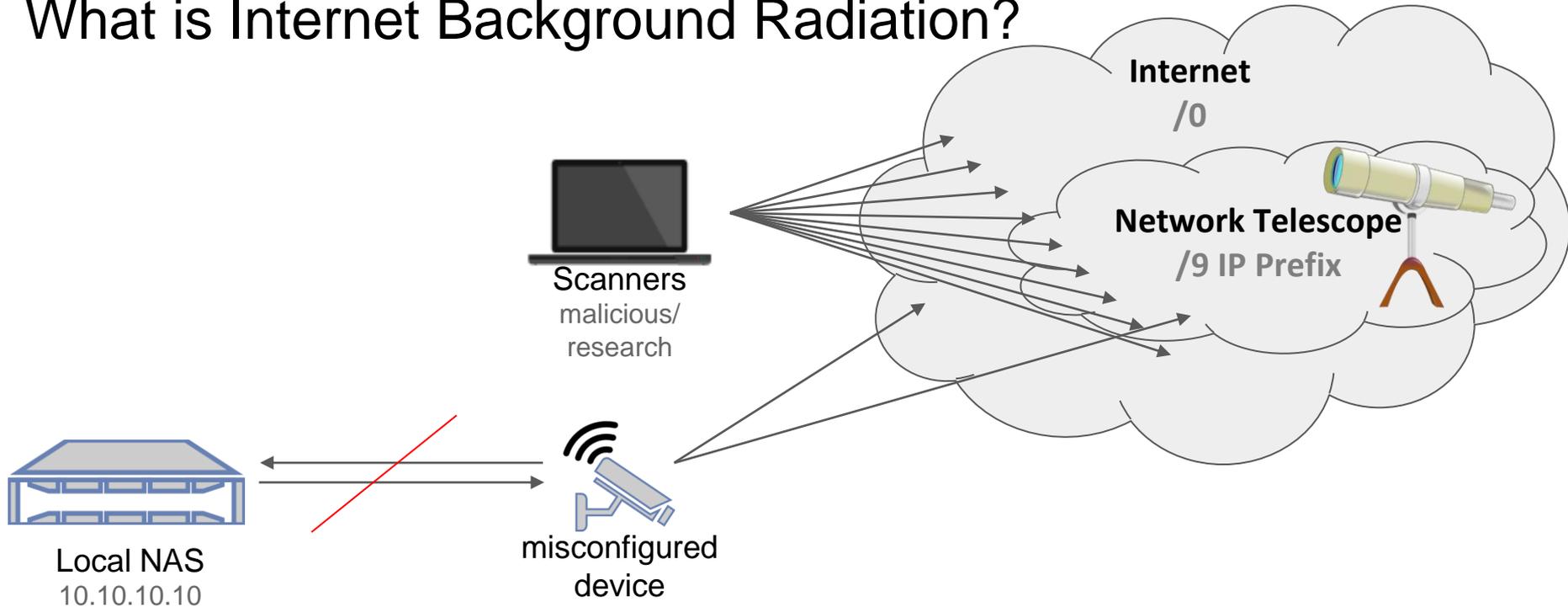
# What is Internet Background Radiation?



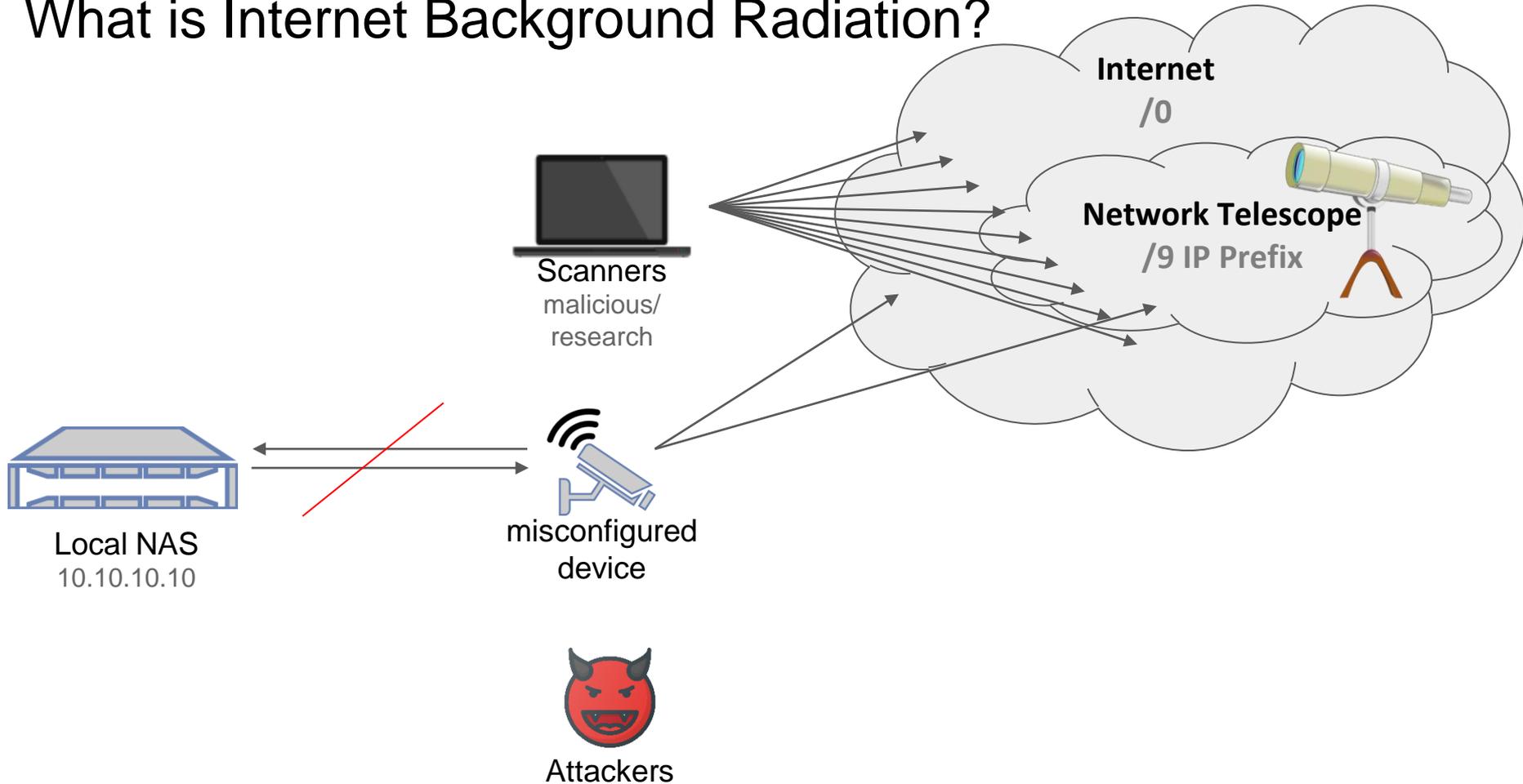
# What is Internet Background Radiation?



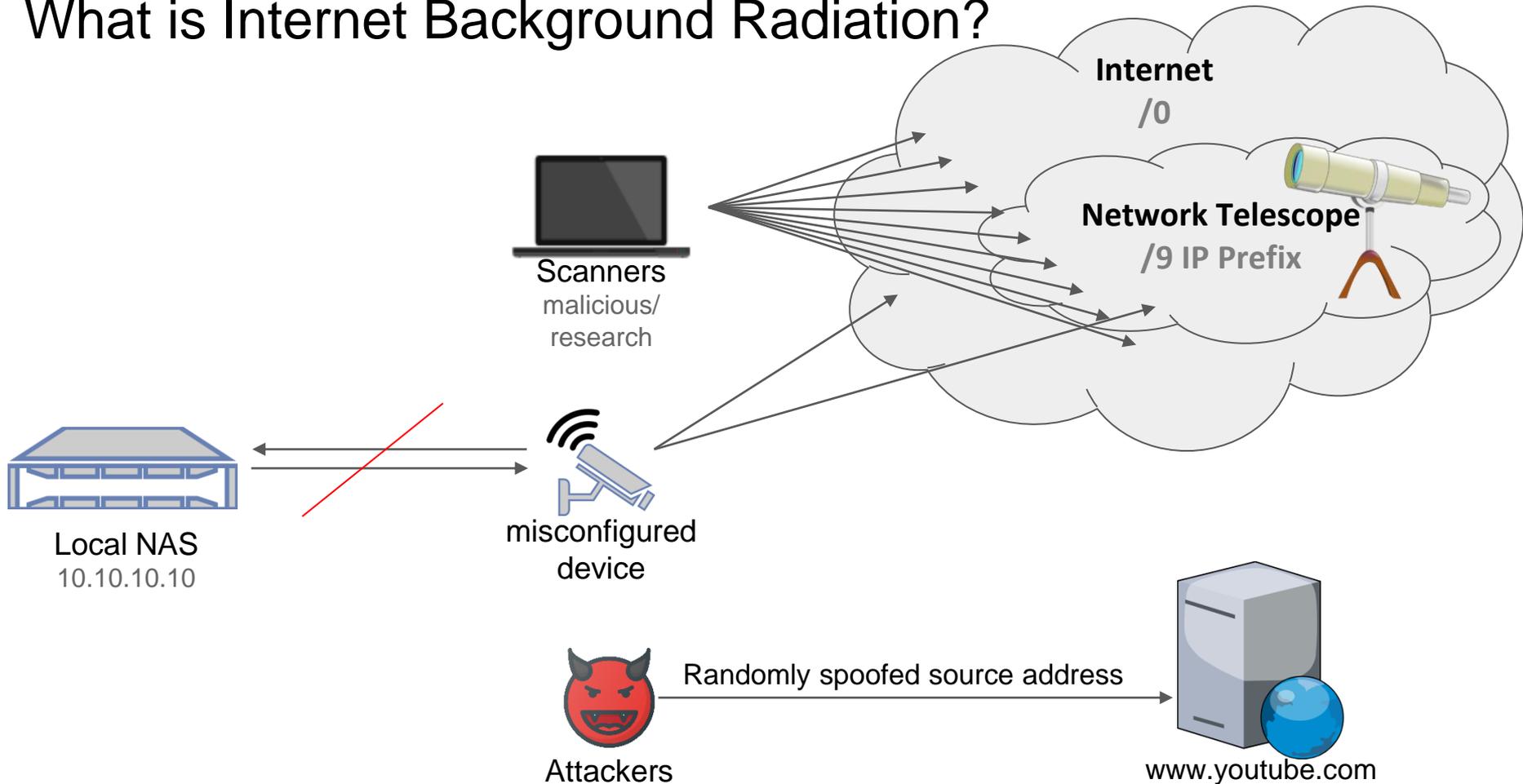
# What is Internet Background Radiation?



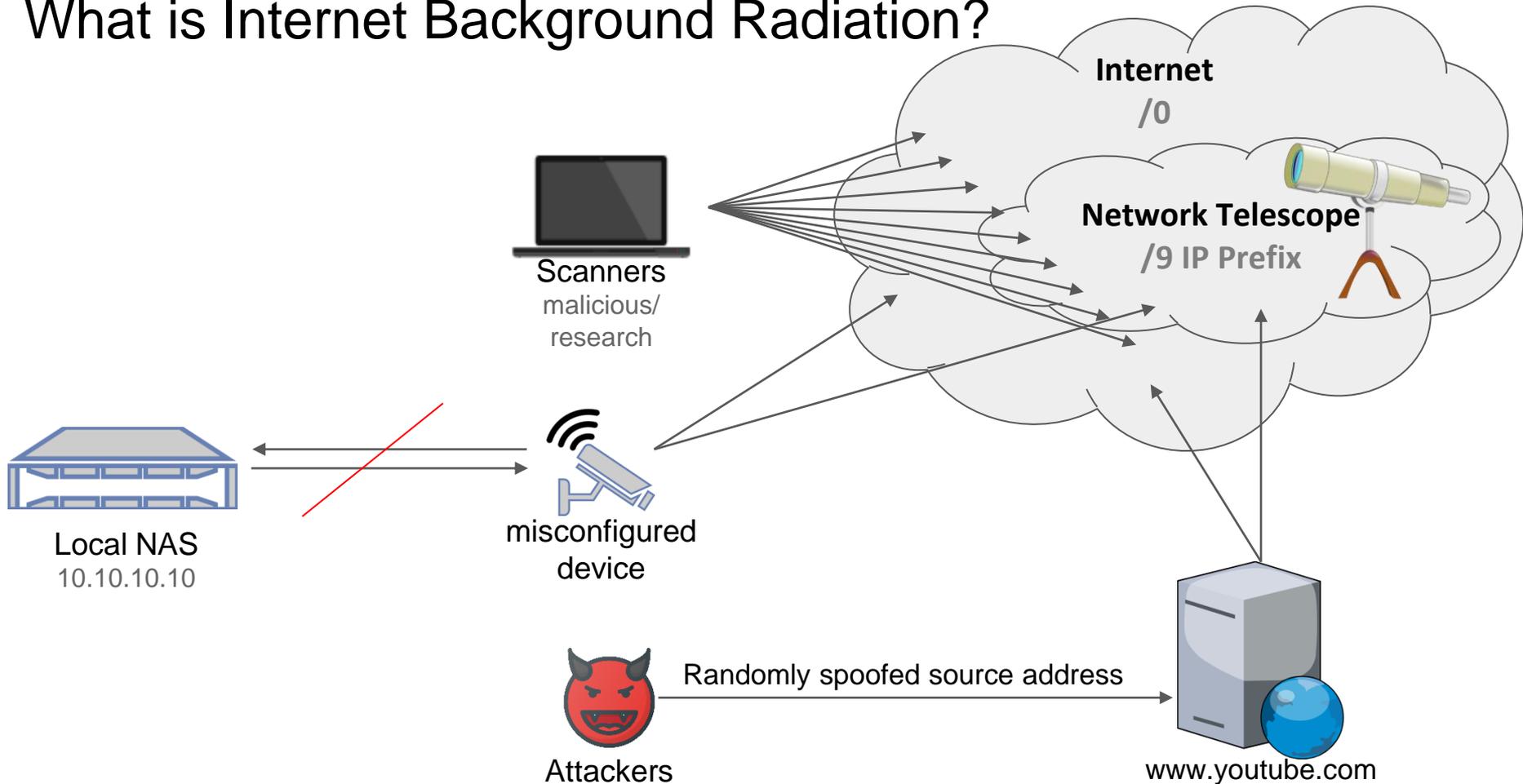
# What is Internet Background Radiation?



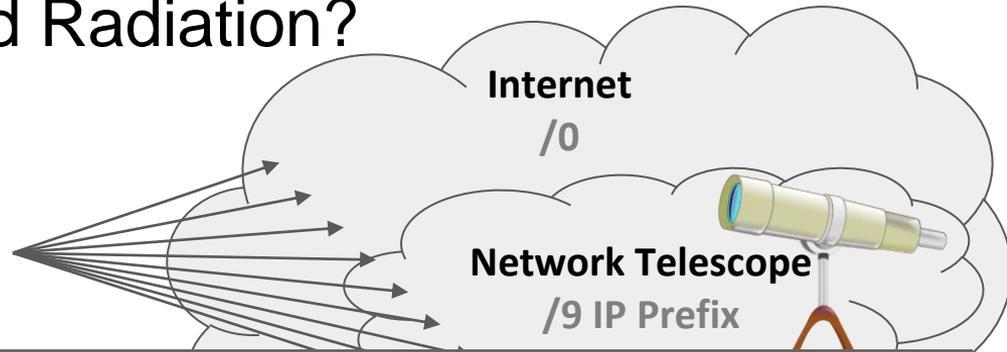
# What is Internet Background Radiation?



# What is Internet Background Radiation?



# What is Internet Background Radiation?



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

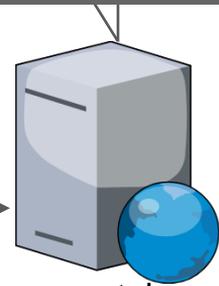
Local NAS  
10.10.10.10

misconfigured  
device



Attackers

Randomly spoofed source address



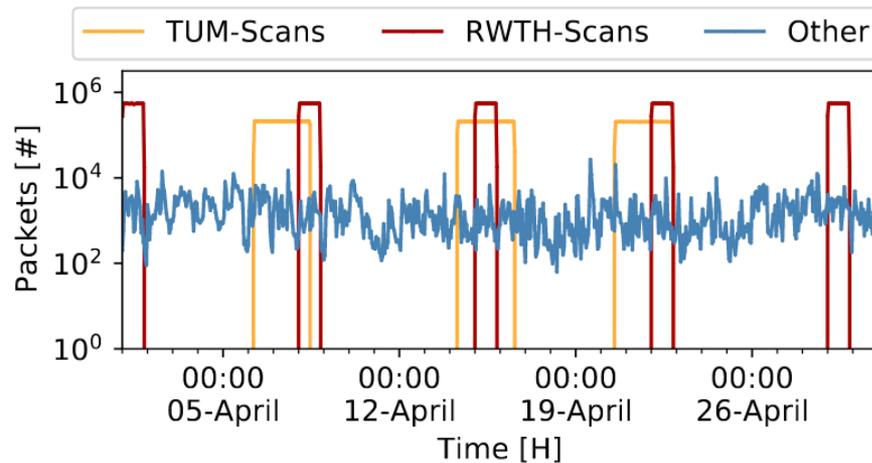
www.youtube.com

# We observe scanners.

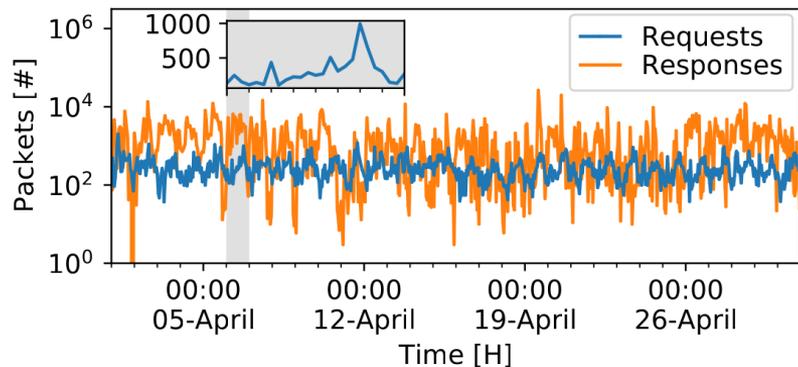


Scanners look for QUIC servers:

- Connection attempts to port 443 (requests)



# We can group responses into attack sessions.

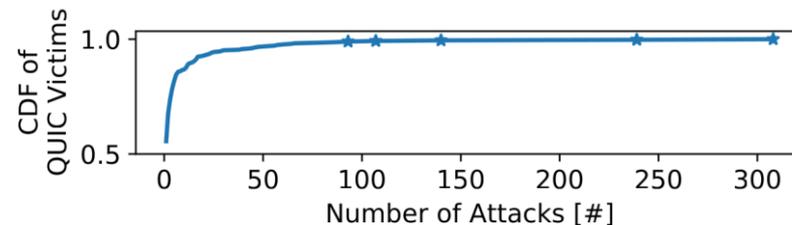


We group responses into attack sessions with the following thresholds:

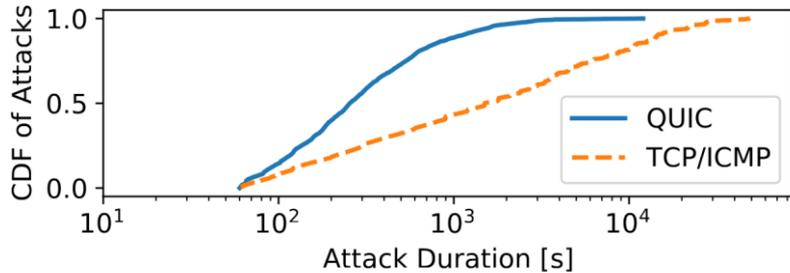
- more than 25 packets
- longer than 60s
- maximum packet rate  $> 0.5\text{pps}$

2905 attacks (394 IP addresses)

More than half are attacked only a single time.



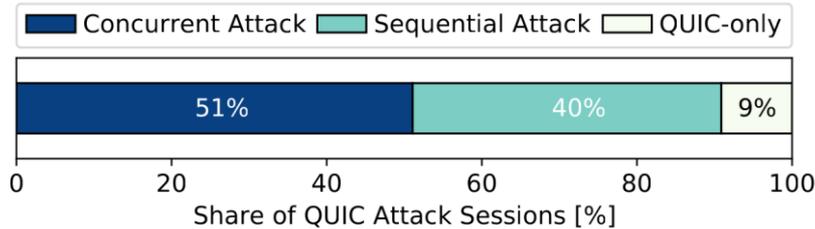
# We observe QUIC attacks in parallel with TCP/ICMP attacks.



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 webserver is impacted by requests.

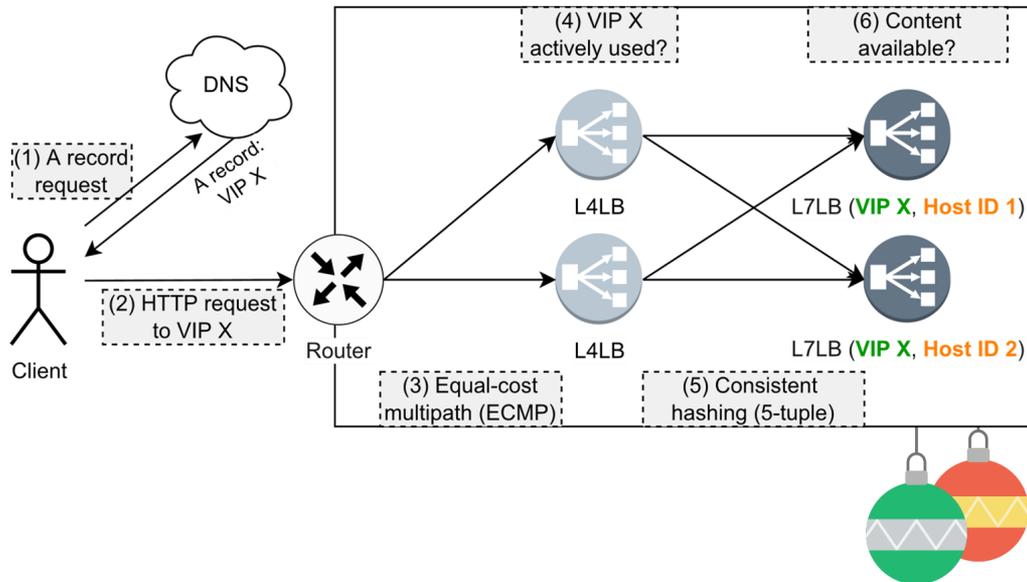
Attack	NGINX Config		Results				
	Volume [pps]	QUIC Retry	Workers [#]	Client [# Req]	Server [# Resp]	Service Available	Extra RTT
	10	✗	4	3,001	12,004	100%	✗
	100	✗	4	30,001	81,680	68%	✗
	1,000	✗	4	300,001	81,680	7%	✗
	1,000	✗	auto=128	300,001	1,200,004	100%	✗
	10,000	✗	auto=128	500,000	522,752	26%	✗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



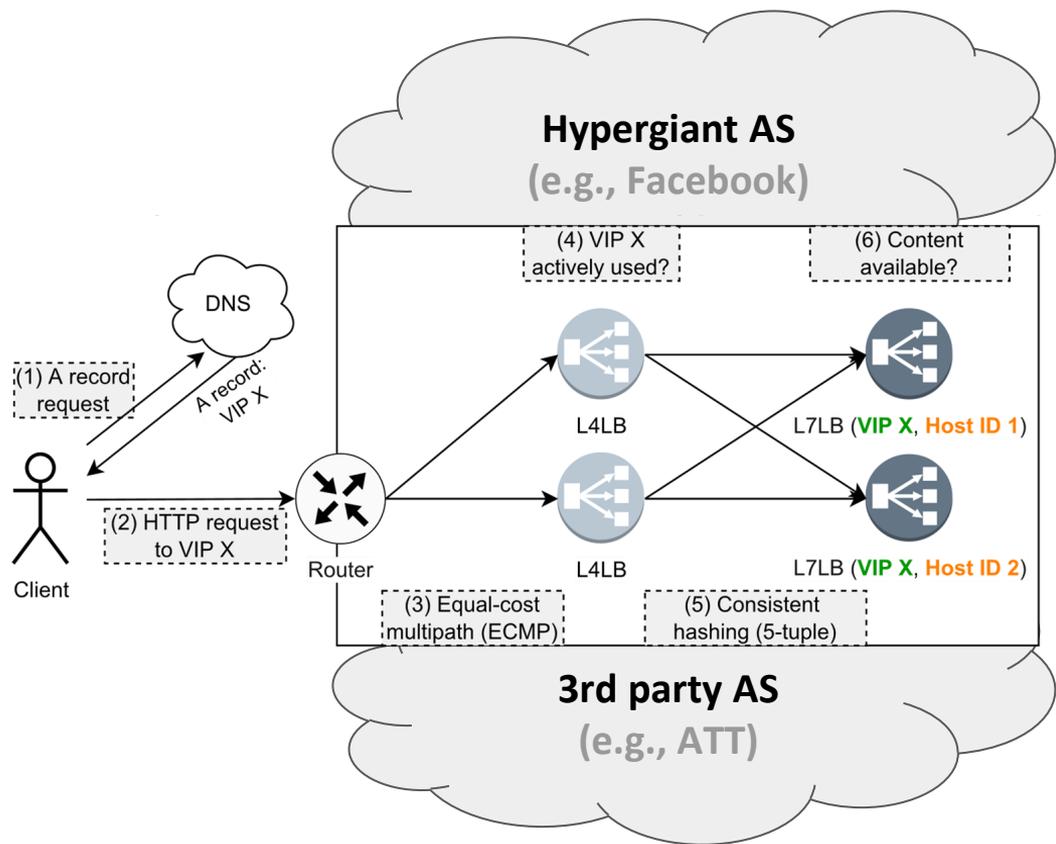
We just analyzed attacks. But can we also use backscatter to learn more about hypergiants?

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 [%]			
<b>QUIC packet type</b>	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
<b>Coalescing packets</b>				
Initial, Handshake	3.289	0.000	52.730	9.081
Handshake, Initial	0.000	0.000	0.000	0.001

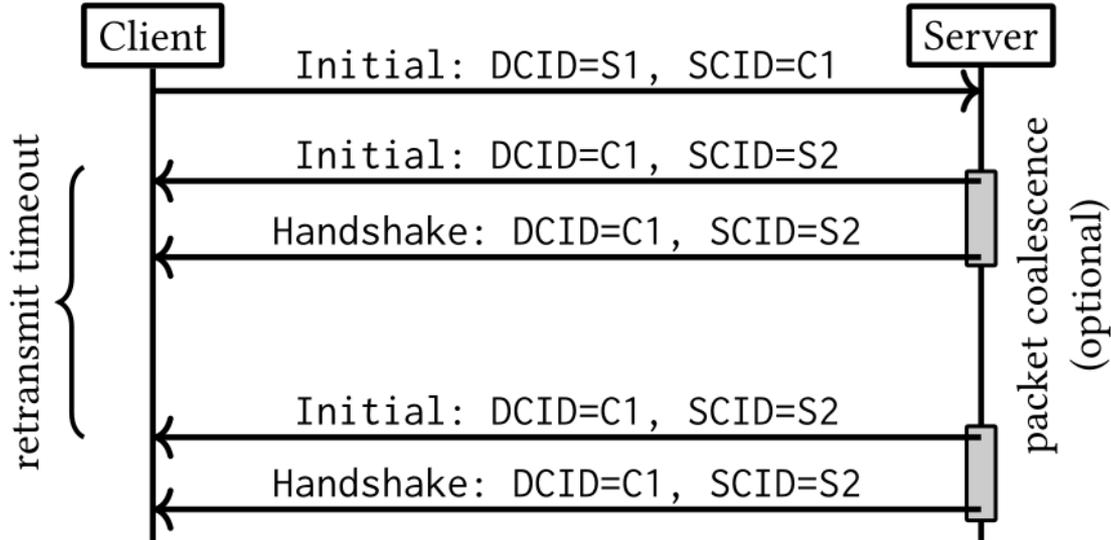
# Merging multiple QUIC packets into a single UDP datagram

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

Cloudflare and Google enable packet coalescing.  
Facebook does not.

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

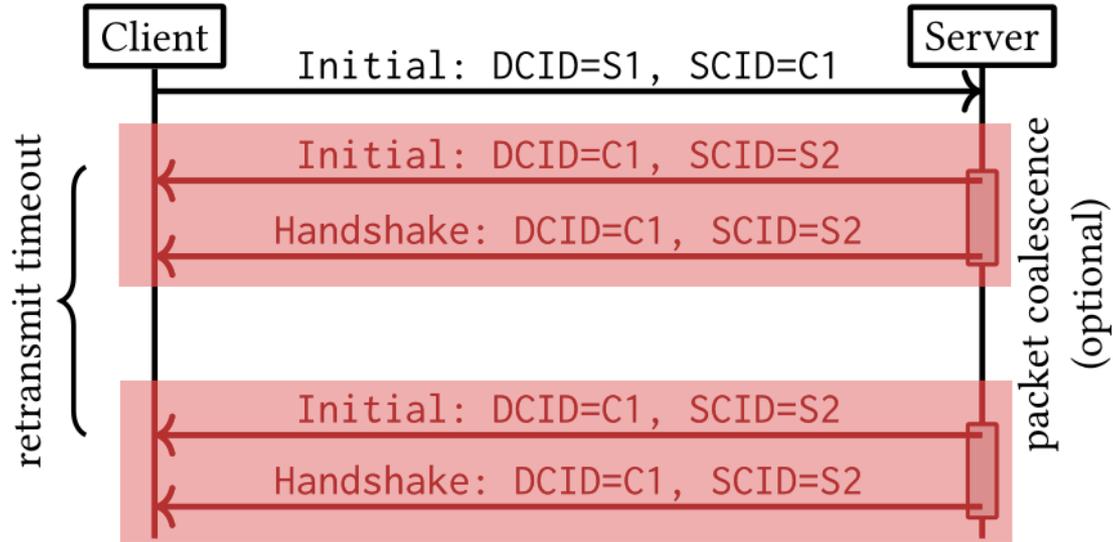
# Incomplete handshakes cause resends.



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.

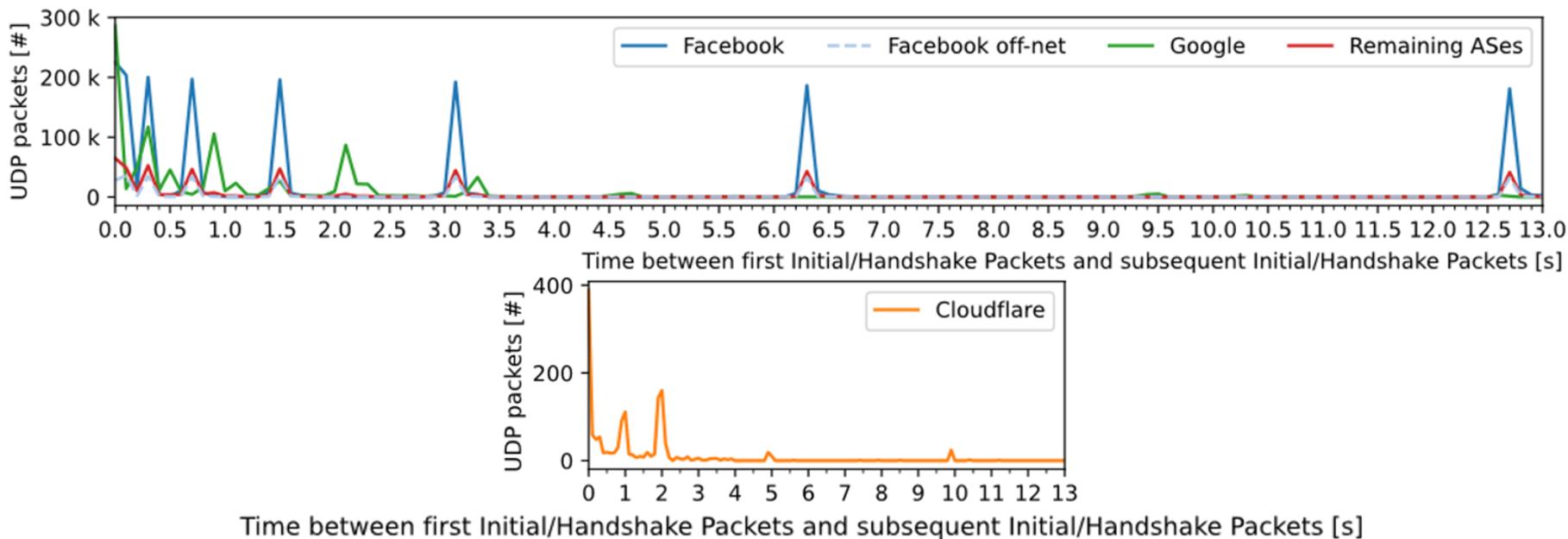


QUIC connections are identified by **connection IDs** and not ports.

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



# Inter-arrival times of incomplete Handshakes

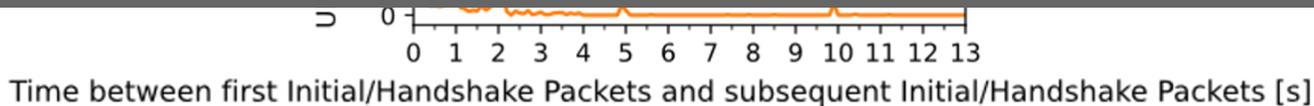




# Inter-arrival times of incomplete Handshakes



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

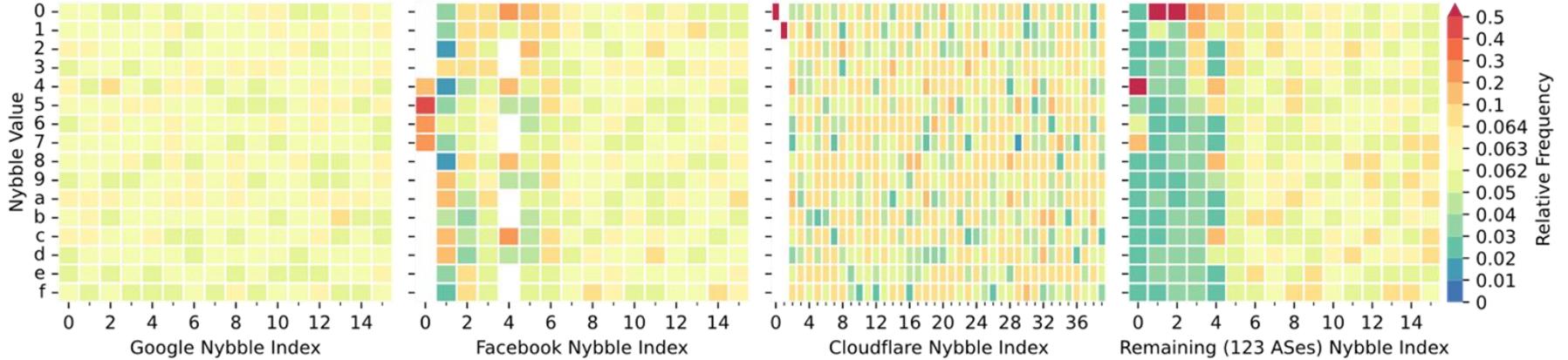


# Structure of QUIC Server Connection IDs (SCIDs)



XXXXXXXX...XXXXXXXXXX max. length 20 Byte  
(half Byte, Nybble) 0...f |

# Structure of QUIC Server Connection IDs (SCIDs)

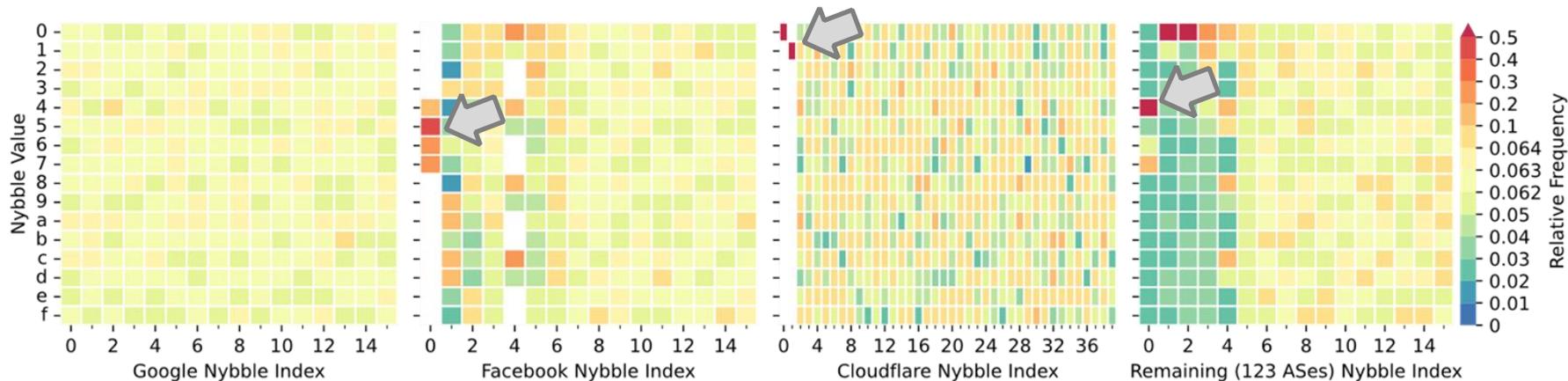


XXXXXXXX...XXXXXXXXXX

max. length 20 Byte

(half Byte, Nybble) 0...f |

# Structure of QUIC Server Connection IDs (SCIDs)

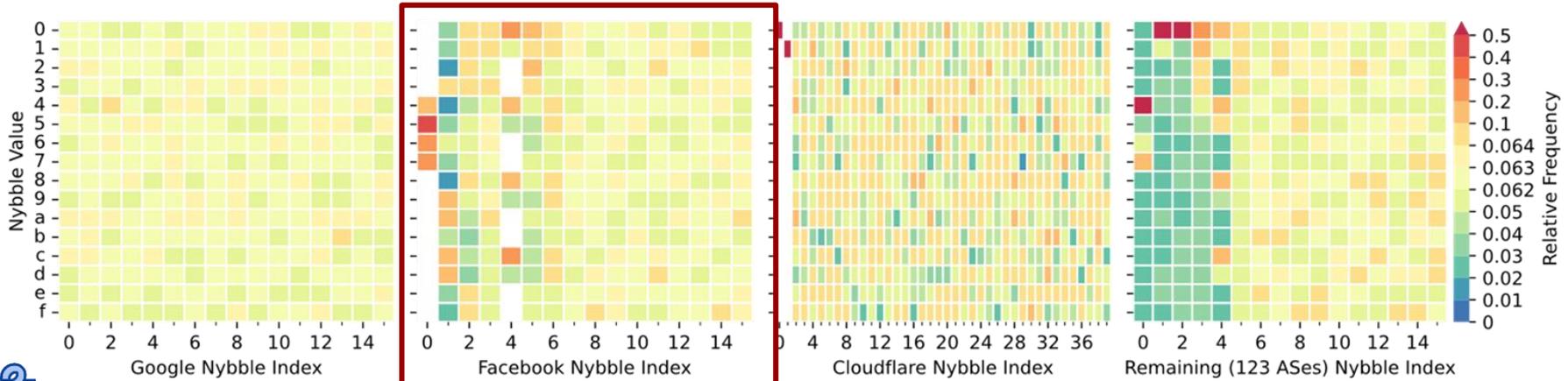


XXXXXXXX...XXXXXXXXXX

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.

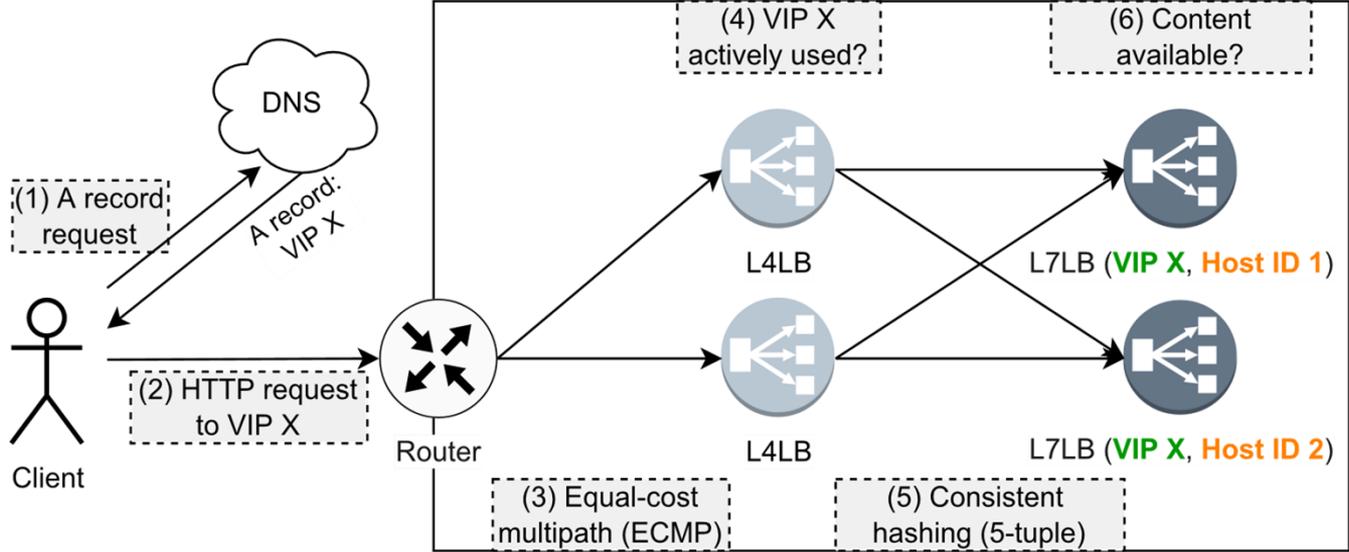
SCID Version	Version	Host ID	Worker ID	Process ID	Remaining (random)
1	0-1	2-17	18-25	26	27-63
2	0-1	8-31	32-39	40	2-7,41-63

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

# Detecting Facebook off-net servers

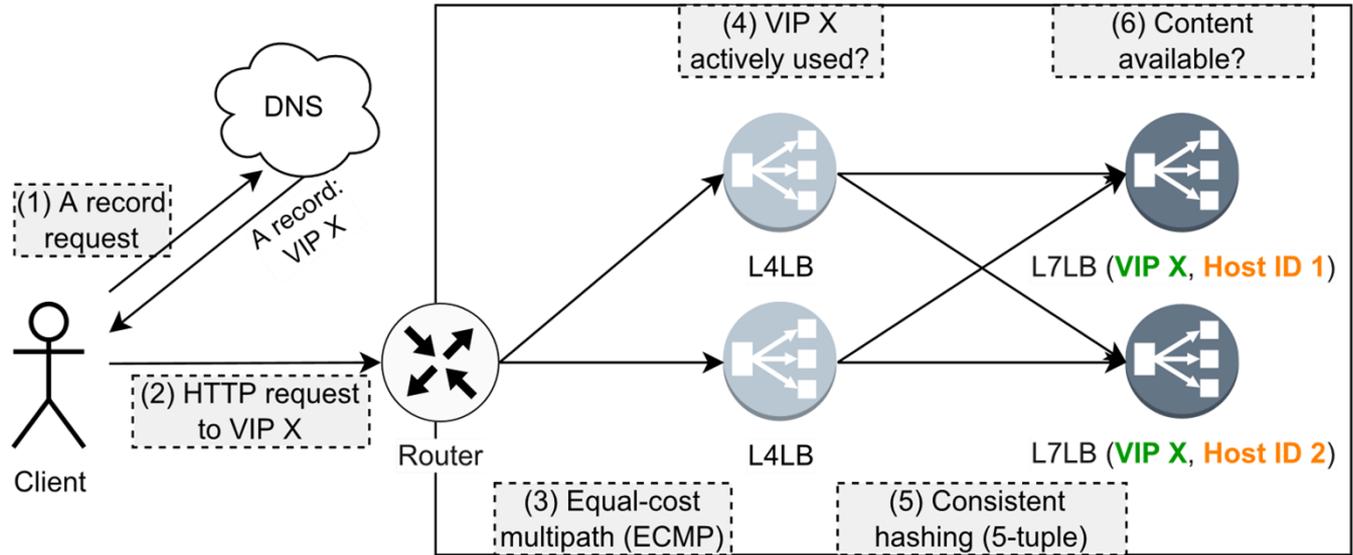
Classifier	TPR	FPR	TNR	FNR	Precision	Recall
Inter-Arrival Time (IAT)	0.772	0.268	0.732	0.228	0.645	0.772
SCID, IAT	0.772	0.046	0.954	0.228	0.914	0.772
Packet Length	0.997	0.328	0.672	0.003	0.657	0.997
Coalescence	1.000	0.931	0.069	0.000	0.403	1.000
<b>SCID</b>	<b>1.000</b>	<b>0.193</b>	<b>0.807</b>	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

# 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

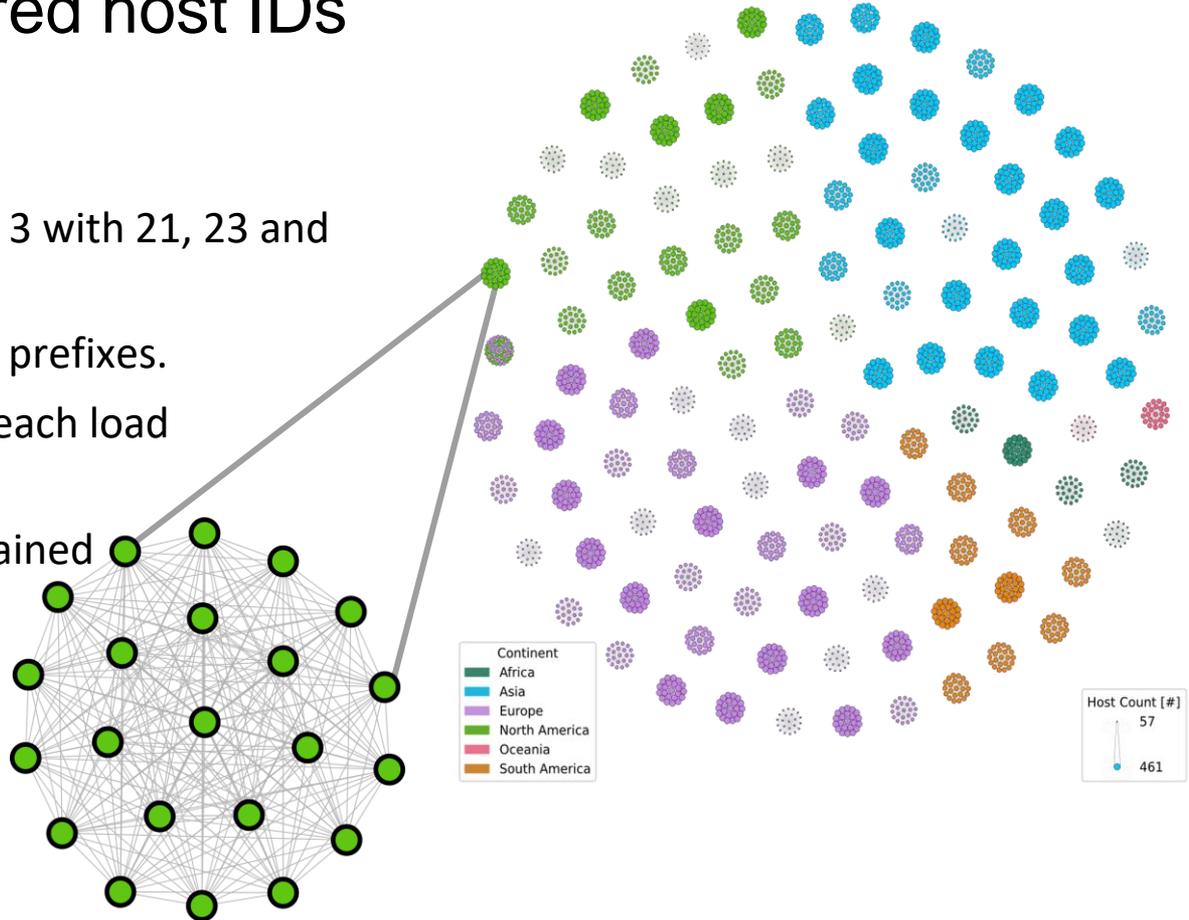
1 IP address = 1 node

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

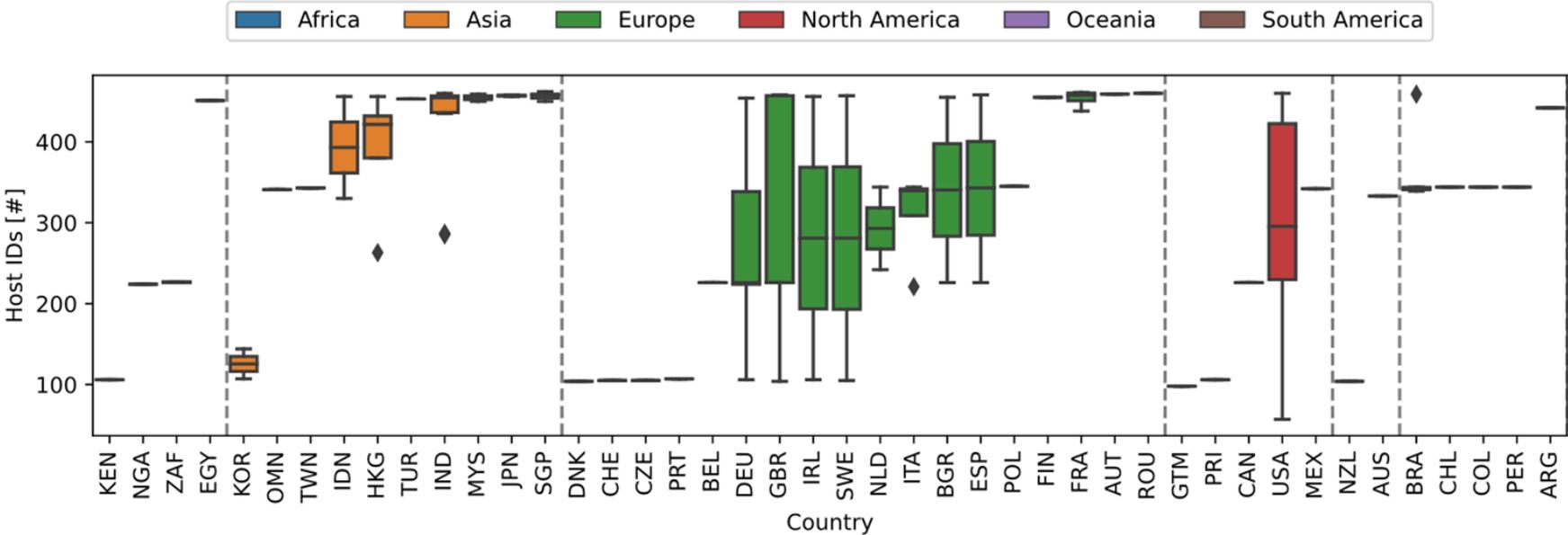
Clusters are organized in /24 prefixes.

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

QUIC can reduce latency. Inefficient handshakes (Multi-RTT) increase handshake duration. The anti-amplification-limit is often violated by implementations.

Certificate compression, signing algorithms and packet coalescence can improve the handshake.

QUIC attacks happen and the found amplification factors compare to often used protocols.

The RETRY option is an effective mitigation.

Passive measurements can be used for off-net detection. Server connection IDs allow detailed insights into server deployments.

Information encoding in connection IDs will be used for efficient stateless load balancing.

# More details



## QUICsand: Quantifying QUIC Reconnaissance Scans and DoS Flooding Events

Marcin Nawrocki  
Freie Universität Berlin

### ABSTRACT

In this paper, we present a fully designed and implemented radiation originator. Our analysis is based on active measurements of the QUIC scan non-benign source handshake is prone to SYN floods. We collected a vector is already in use on the Internet is exposed to attacks occur concurrently as TCP/ICMP floods.

**CCS CONCEPTS**  
• Security and privacy → Transport

**ACM Reference Format**  
Marcin Nawrocki, Matthias Wählisch. 2021. QUIC DoS Flooding Events

### ABSTRACT

In this paper, we setup and relate the common Web deployment with 272k QUIC certificates under connection setup since amplification limit lead to larger amplification even further for all involved sites.

**CCS CONCEPTS**  
• Networks → Transport  
• Security and privacy → Web

**ACM Reference Format**  
Marcin Nawrocki, Patrick Sattler, Thomas C. Schmidt. 2022. QUIC DoS Flooding Events

Jonas Mücke  
Freie Universität Berlin

## 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

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

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

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

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

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

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

Matthias Wählisch  
m.waehlich@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.**

Feature	Hypergiant		
	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.

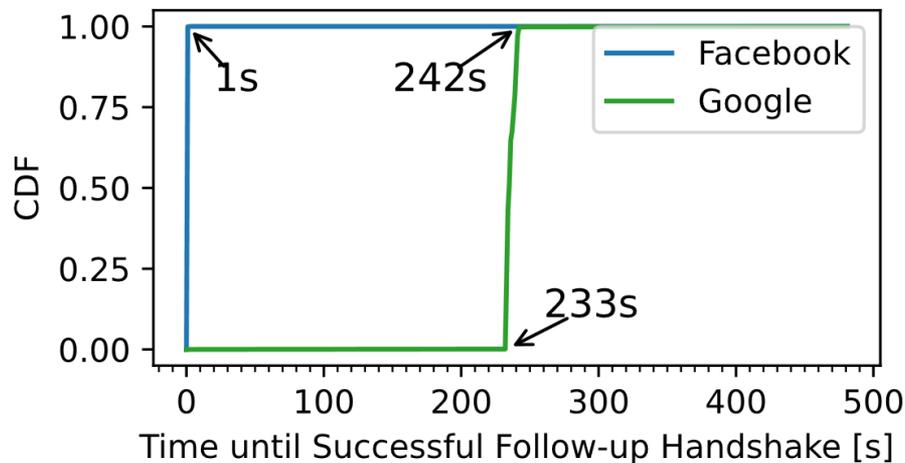
# Facebook and Google use different load balancing methods

## Google uses CID-aware load balancing.

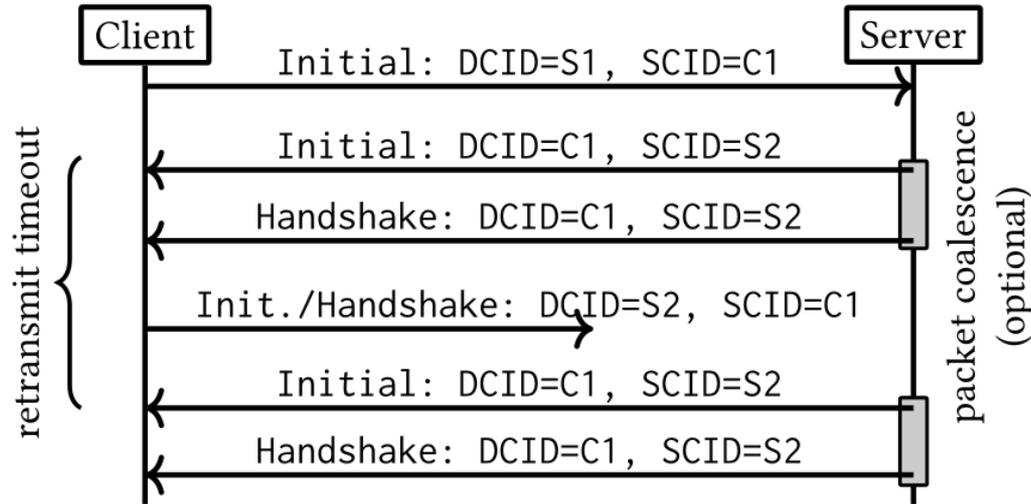
Facebook allows reconnection with client-chosen server connection ID because it uses server-chosen connection IDs.

## 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.

