Hochschule für Angewandte
Wissenschaften Hamburg
Hamburg University of Applied Sciences

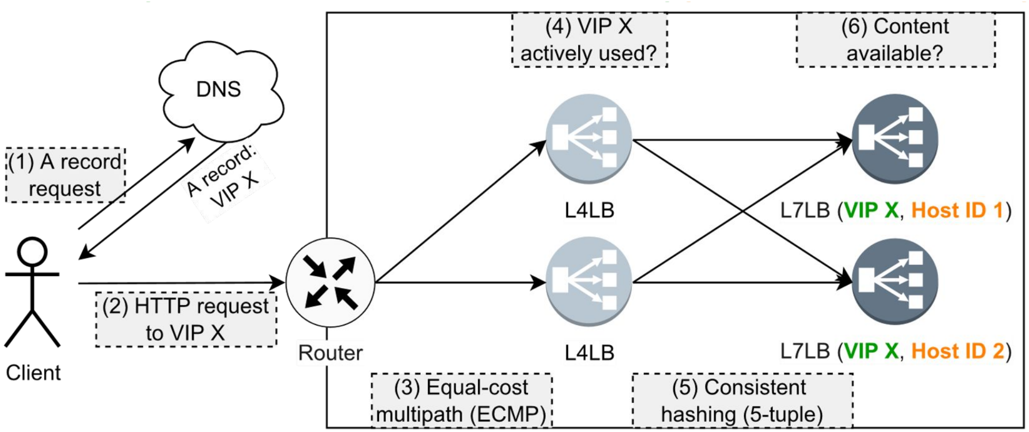
Freie Universität Berlin

On the Opportunities of Passive Measurements to Understand QUIC Deployments

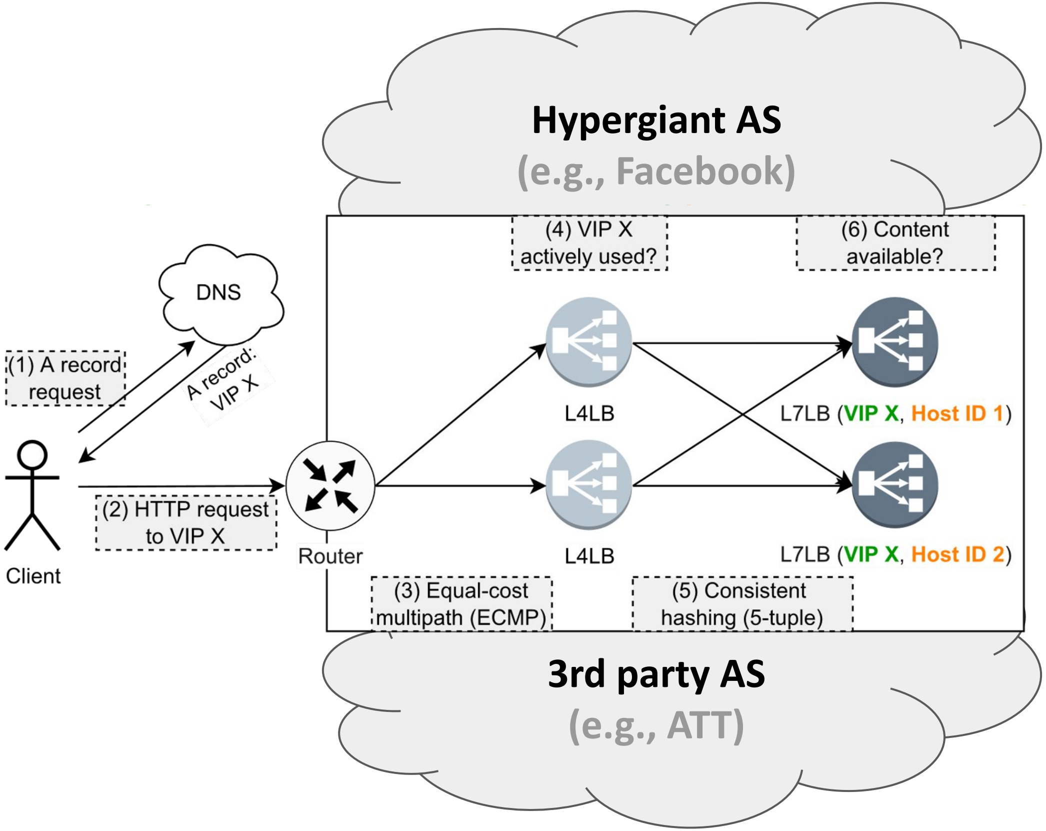
Jonas Mücke, Marcin Nawrocki, Raphael Hiesgen, Patrick Sattler, Johannes Zirngibl,
Georg Carle, Thomas C. Schmidt, Matthias Wählisch

{jonas.muecke, marcin.nawrocki, m.waehlisch}@fu-berlin.de
{raphael.hiesgen, t.schmidt}@haw-hamburg.de
{sattler, zirngibl, carle}@net.in.tum.de

Common hypergiant deployments



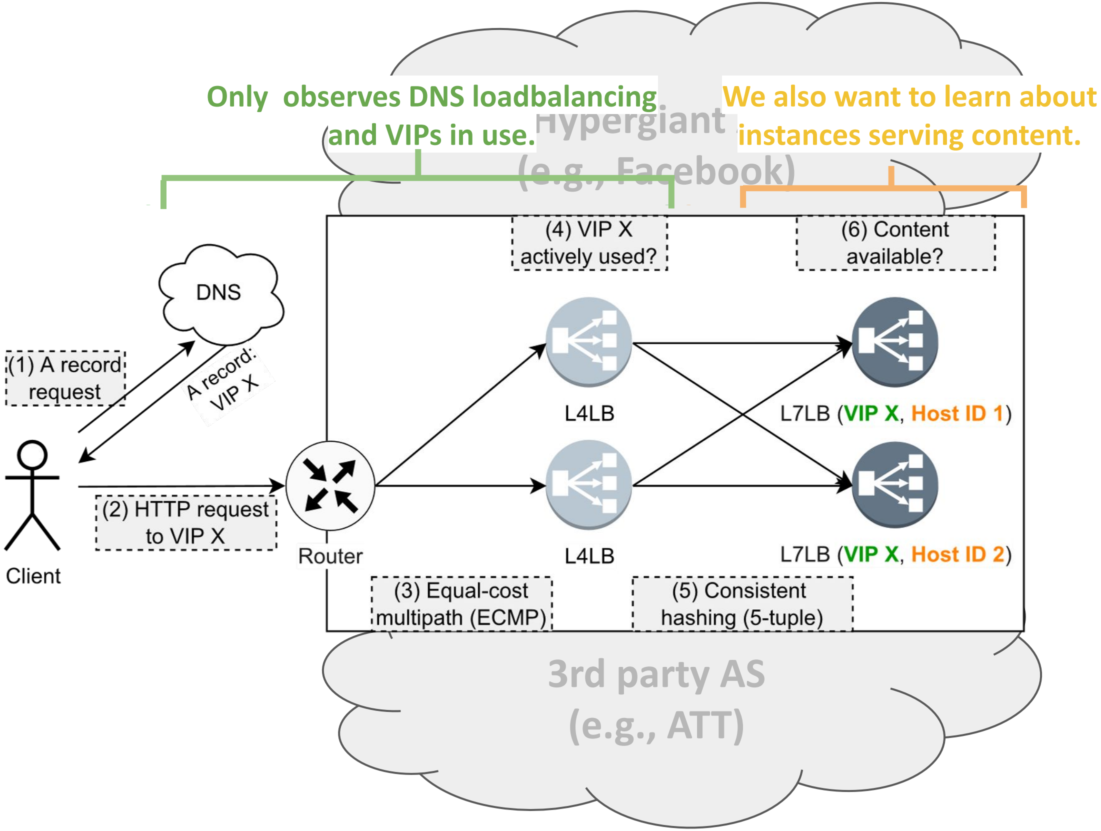
Common hypergiant deployments



On-net deployment

Off-net deployment

Prior work focused on active measurements



Scan for QUIC services, fetch TLS certificates etc.

What do we want to achieve?

Identifying servers of specific hypergiants

Identifying off-net servers

Identifying L7 load balancers

Non-intrusively

Our approach

Analyze QUIC backscatter traffic.

Our approach

Analyze QUIC backscatter traffic.

Why QUIC?

Reduces Web latencies. Broad adoption. (2020, 75% of Facebook traffic is QUIC).

Exposes additional information (compared to UDP and TCP).

Our approach

Why backscatter traffic?

Non-intrusive.

Relatively easy to capture.

Analyze QUIC backscatter traffic.

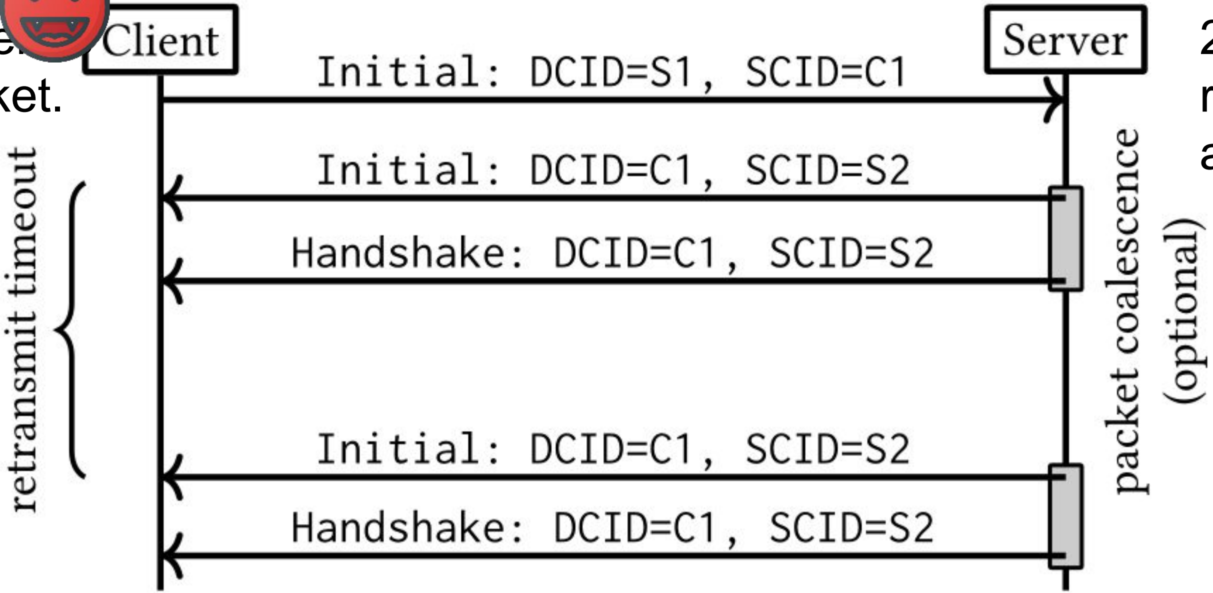
Why QUIC?

Reduces Web latencies. Broad adoption. (2020, 75% of Facebook traffic is QUIC).

Exposes additional information (compared to UDP and TCP).

Setup

1. Attacker se😈 spoofed packet.



Client

Server

Initial: DCID=S1, SCID=C1

Initial: DCID=C1, SCID=S2

Handshake: DCID=C1, SCID=S2

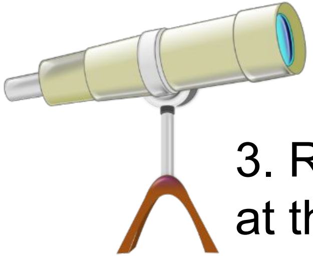
Initial: DCID=C1, SCID=S2

Handshake: DCID=C1, SCID=S2

retransmit timeout

packet coalescence (optional)

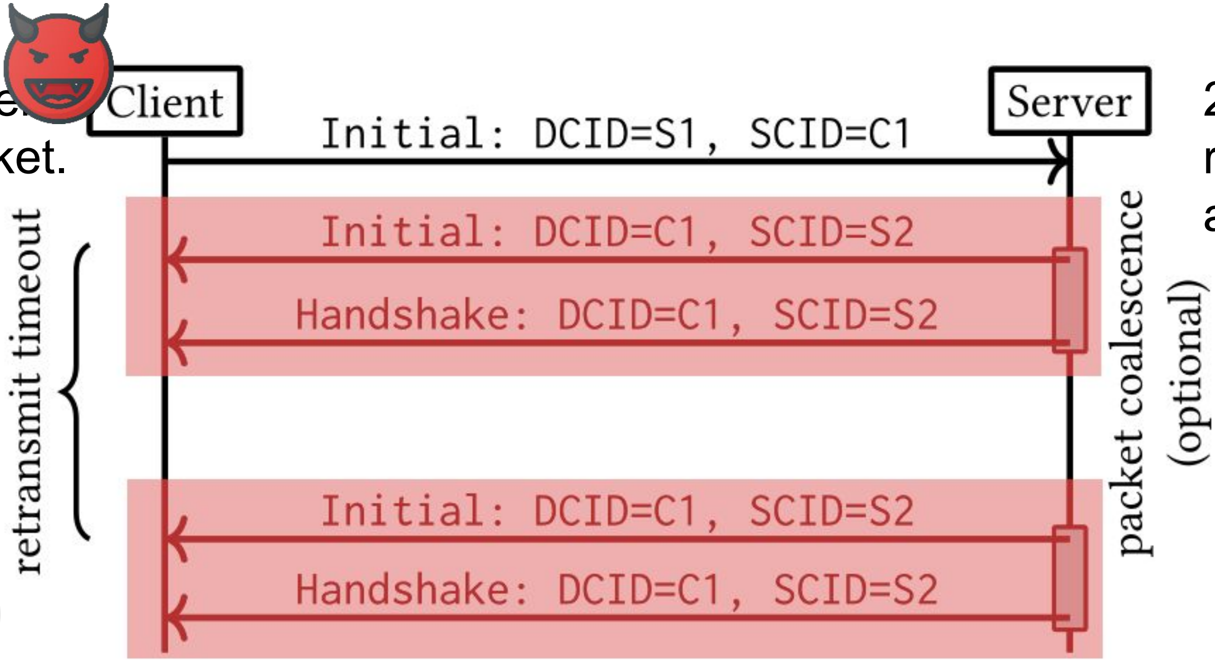
2. Server sends reply to spoofed address.



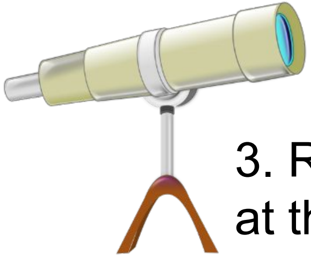
3. Reception of response traffic at the network telescope.

Setup

1. Attacker se
spoofed packet.


Client
Server
Initial: DCID=S1, SCID=C1
Initial: DCID=C1, SCID=S2
Handshake: DCID=C1, SCID=S2
Initial: DCID=C1, SCID=S2
Handshake: DCID=C1, SCID=S2
retransmit timeout
packet coalescence
(optional)

2. Server sends
reply to spoofed
address.



3. Reception of response traffic
at the network telescope.

Setup

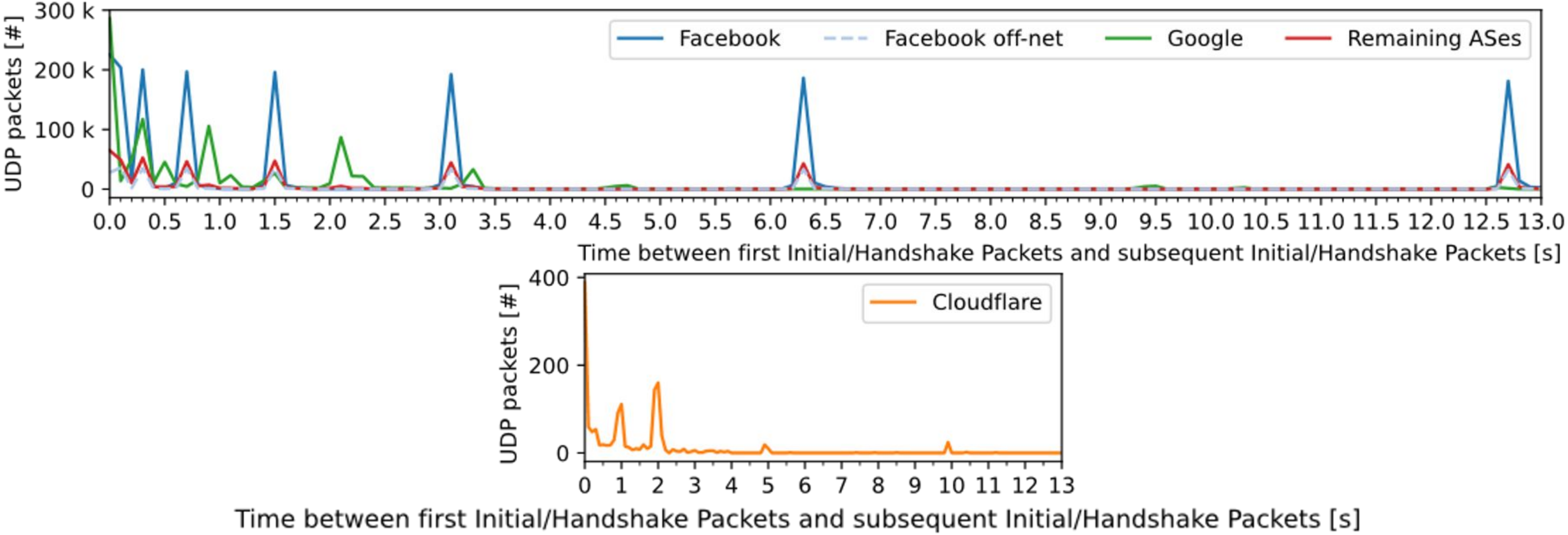
Passive measurements using the CAIDA /9 IPv4 network telescope

- January 1-31, 2022

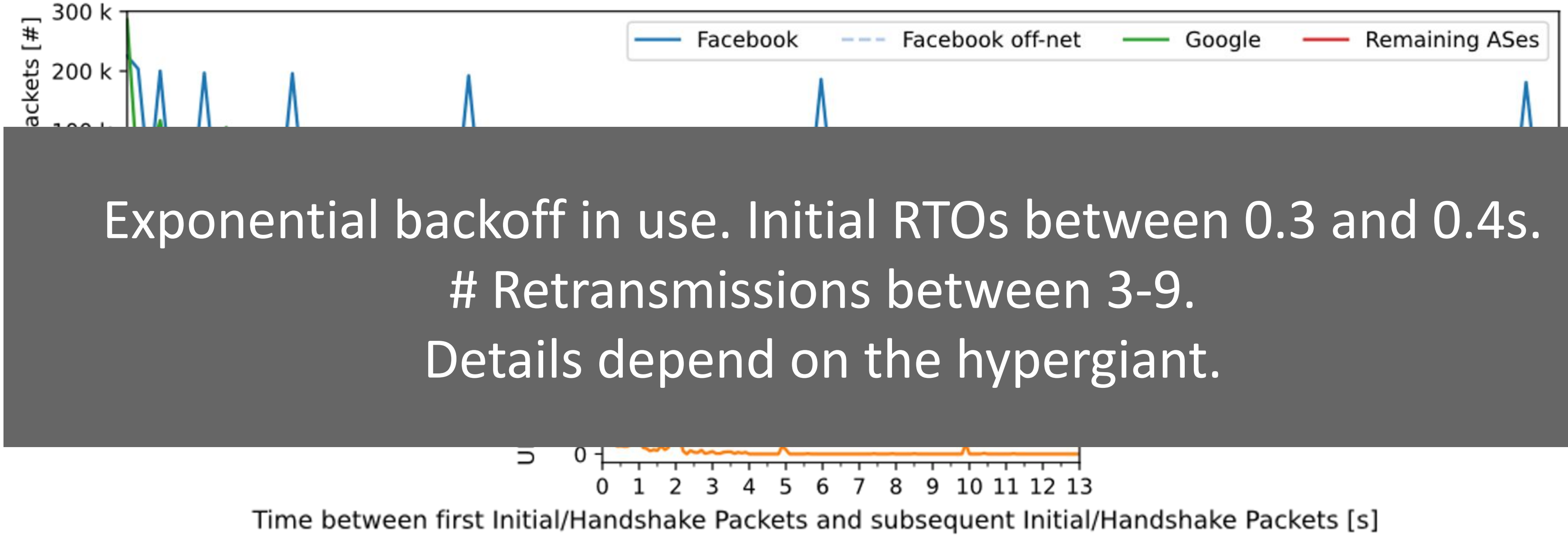
Active measurements for verification, where data is sparse, and additional information about the sender is required

- QUIC scans
- TLS scans
- DNS scans

Inter-arrival times of Initial/Handshakes packets not answered



Inter-arrival times of Initial/Handshakes packets not answered



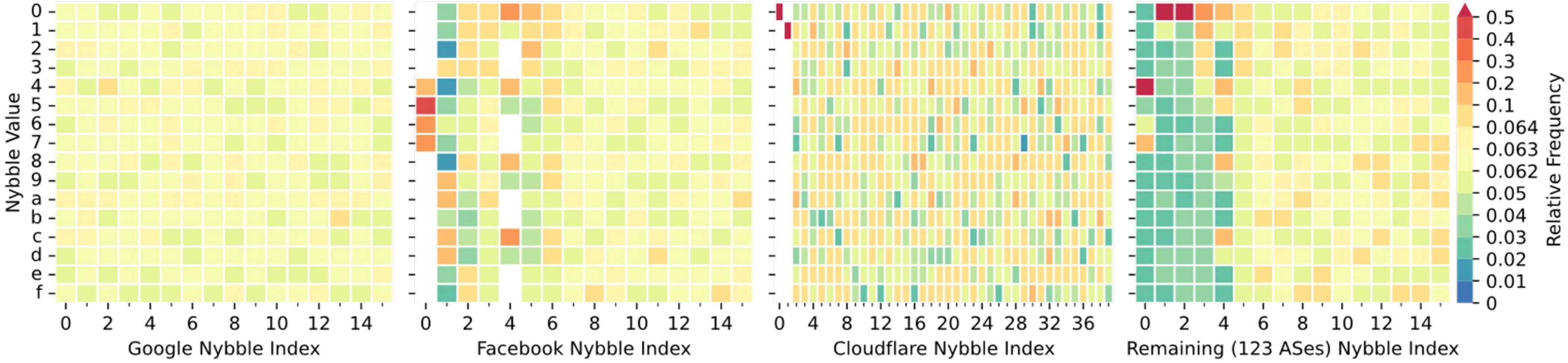
Exponential backoff in use. Initial RTOs between 0.3 and 0.4s.
# Retransmissions between 3-9.
Details depend on the hypergiant.

Structure of QUIC Server Connection IDs (SCIDs)

`XXXXXXX…XXXXXXXXX` max. length 20 Byte

(half Byte, Nybble) `0…f`

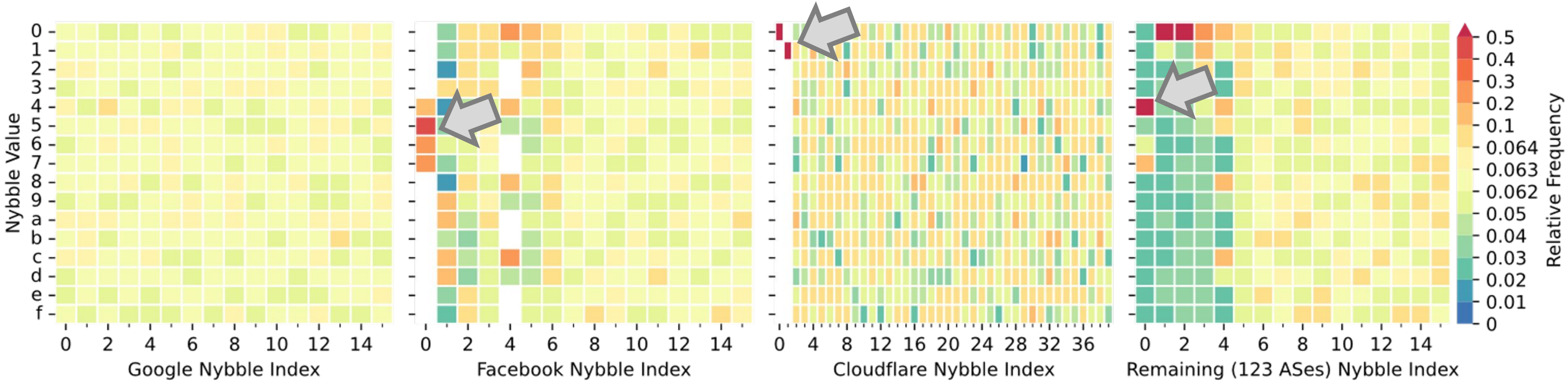
Structure of QUIC Server Connection IDs (SCIDs)



XXXXXXX…XXXXXXXXX max. length 20 Byte

(half Byte, Nybble) 0…f

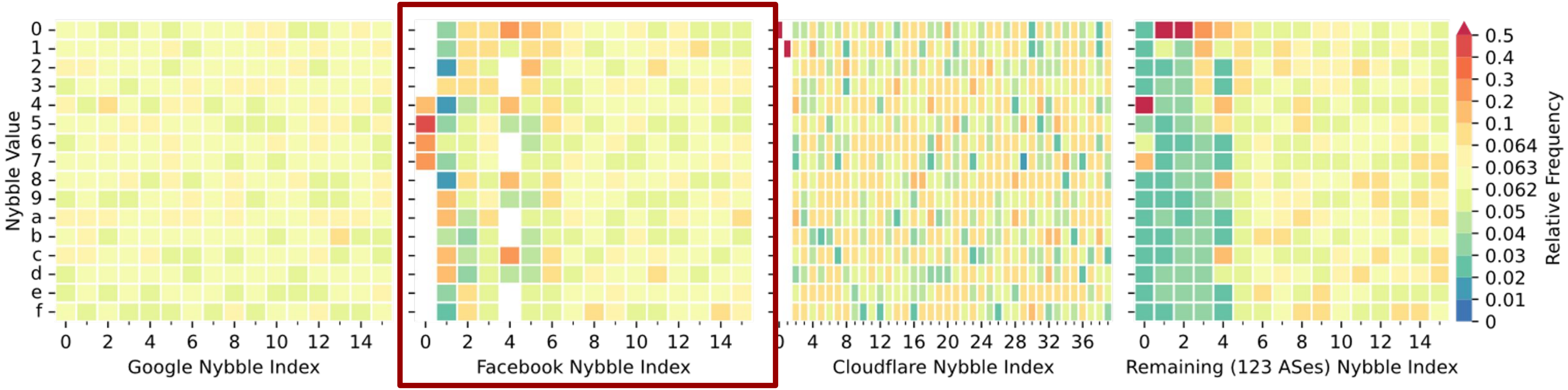
Structure of QUIC Server Connection IDs (SCIDs)



XXXXXXX…XXXXXXXXX max. length 20 Byte

(half Byte, Nybble) 0…f

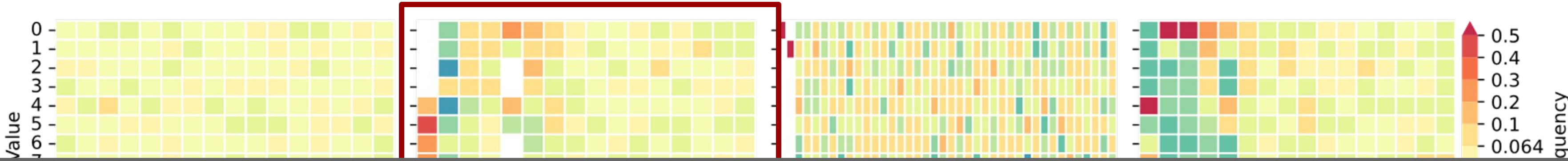
Structure of QUIC Server Connection IDs (SCIDs)



	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.

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

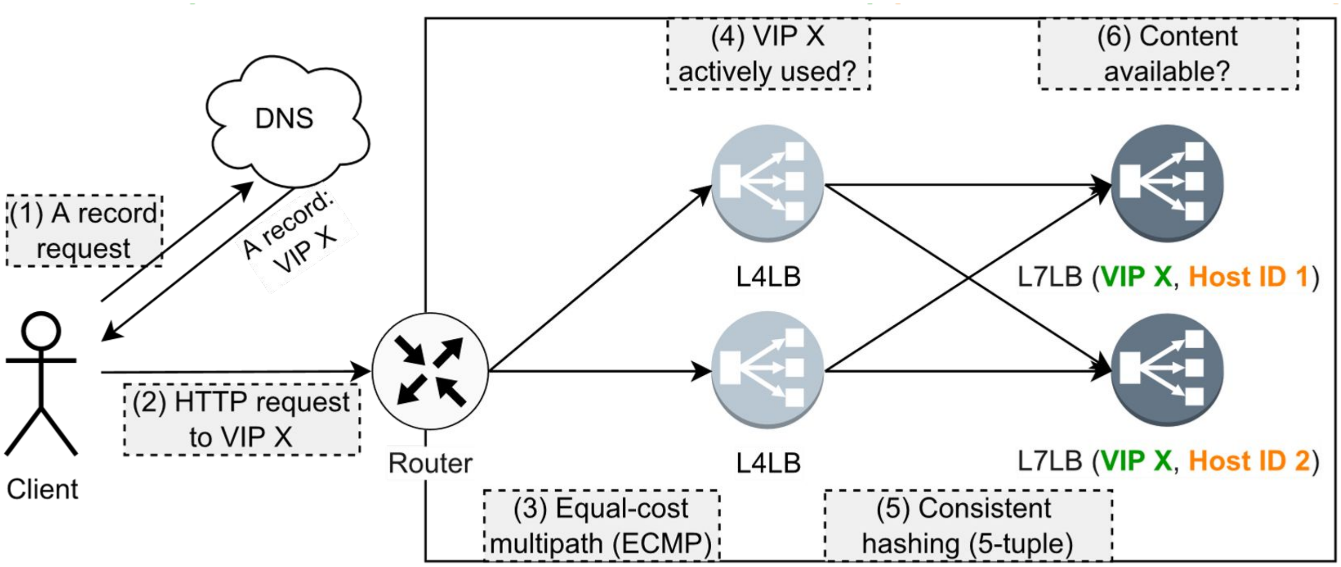
Facebook’s SCID Structure according to their QUIC Implementation mvfst.

Detecting Facebook off-net servers

Verified by SAN in TLS certificates

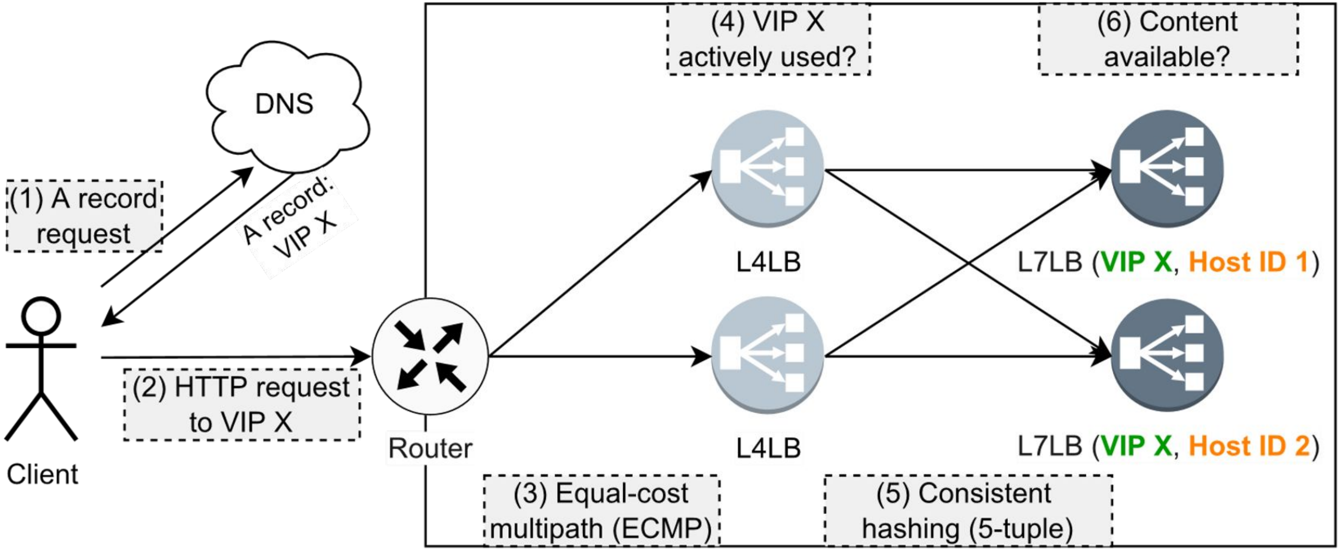
Classificator	TPR	FPR	TNR	FNR	Precision	Recall
Inter-Arrival Time (IAT)	0.772	0.268	0.732	0.228	0.645	0.772
SCID, IAT	0.772	0.046	0.954	0.228	0.914	0.772
Packet Length	0.997	0.328	0.672	0.003	0.657	0.997
Coalescence	1.000	0.931	0.069	0.000	0.403	1.000
SCID	**1.000**	**0.193**	**0.807**	**0.000**	**0.765**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

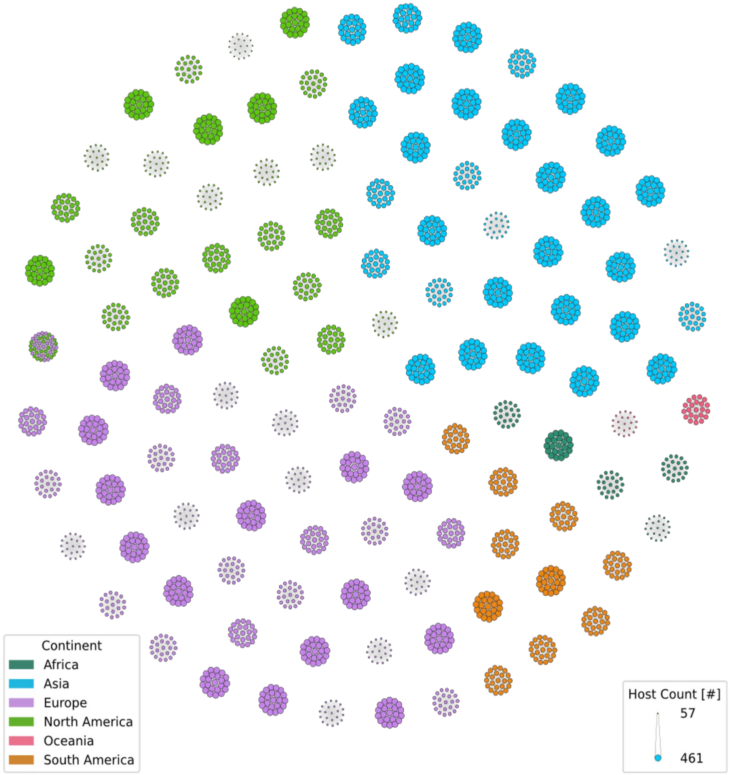


Exploring frontend clusters

Each cluster forms a complete graph

One cluster spans two /24 IP prefixes

All remaining clusters span one /24 IP prefix

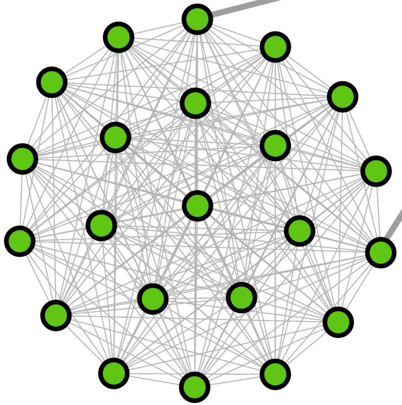


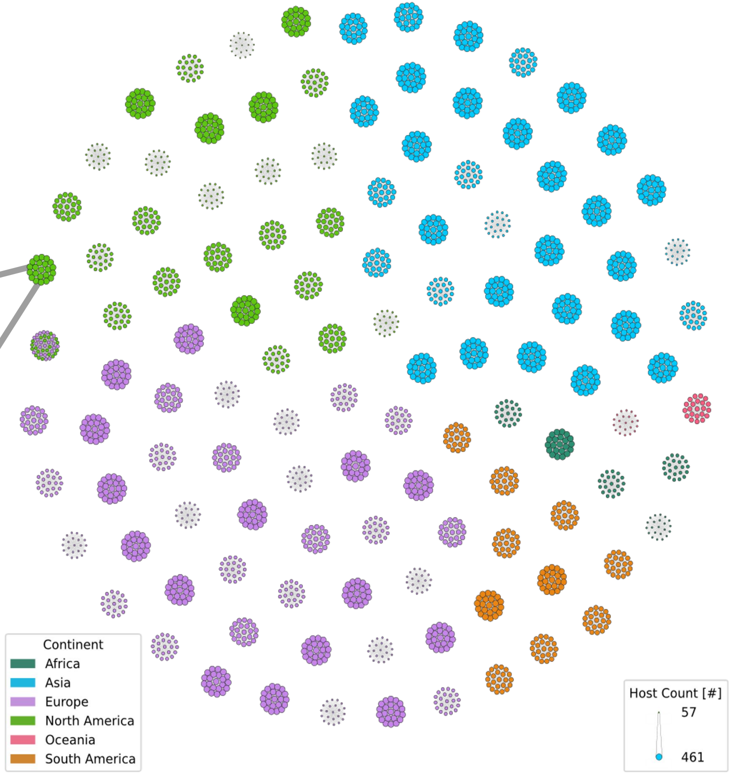
Exploring frontend clusters

Each cluster forms a complete graph

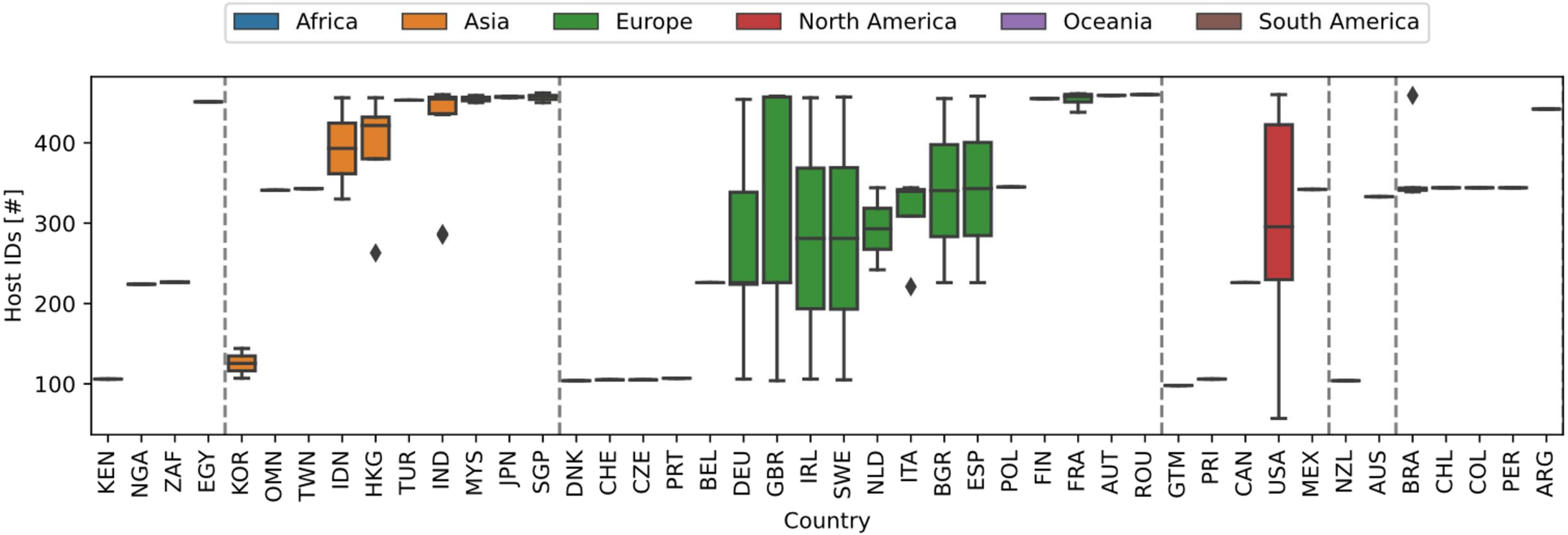
One cluster spans two /24 IP prefixes

All remaining clusters span one /24 IP prefix





Facebook cluster sizes per country



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

Will our principle approach be valid in the future?

Yes.

Backscatter data relies on malicious traffic

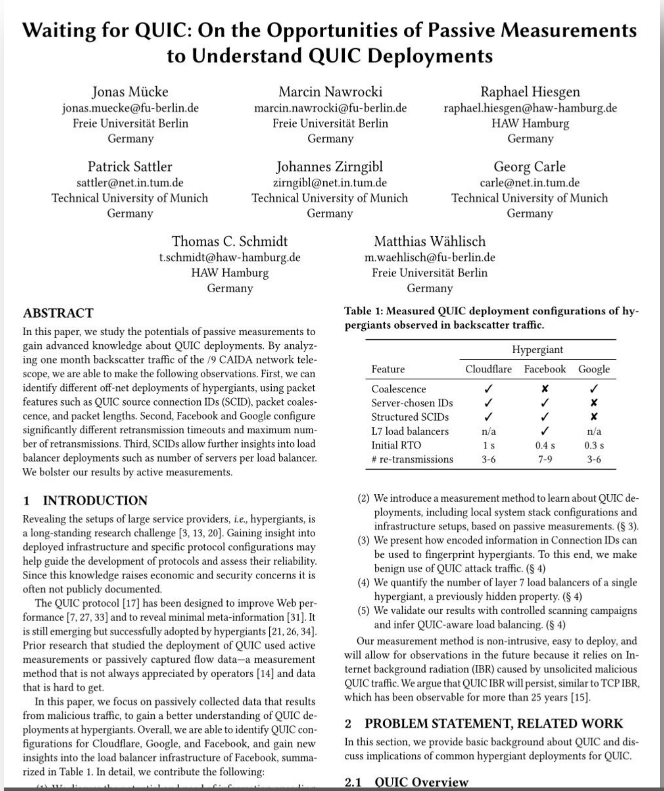
There will be no Internet w/o attackers.

Conclusion

Passive, non-intrusive measurement data
can tell us a lot about hypergiant deployments.
Use QUIC features to create fingerprints.

Structured Connection IDs simplify routing
e.g., ID draft-ietf-quic-load-balancers.

More details



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 help guide the development of protocols and assess their reliability. Since this knowledge raises economic and security concerns it is often not publicly documented.

The QUIC protocol [17] has been designed to improve Web performance [7, 27, 33] and to reveal minimal meta-information [31]. It is still emerging but successfully adopted by hypergiants [21, 26, 34]. Prior research that studied the deployment of QUIC used active measurements or passively captured flow data—a measurement method that is not always appreciated by operators [14] and data that is hard to get.

In this paper, we focus on passively collected data that results from malicious traffic, to gain a better understanding of QUIC deployments at hypergiants. Overall, we are able to identify QUIC configurations for Cloudflare, Google, and Facebook, and gain new insights into the load balancer infrastructure of Facebook, summarized in Table 1. In detail, we contribute the following:

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 benign use of QUIC attack traffic. (§ 4)
(4) We quantify the number of layer 7 load balancers of a single hypergiant, a previously hidden property. (§ 4)
(5) We validate our results with controlled scanning campaigns and infer QUIC-aware load balancing. (§ 4)

Our measurement method is non-intrusive, easy to deploy, and will allow for observations in the future because it relies on Internet background radiation (IBR) caused by unsolicited malicious QUIC traffic. We argue that QUIC IBR will persist, similar to TCP IBR, which has been observable for more than 25 years [15].

2 PROBLEM STATEMENT, RELATED WORK

In this section, we provide basic background about QUIC and discuss implications of common hypergiant deployments for QUIC.

2.1 QUIC Overview

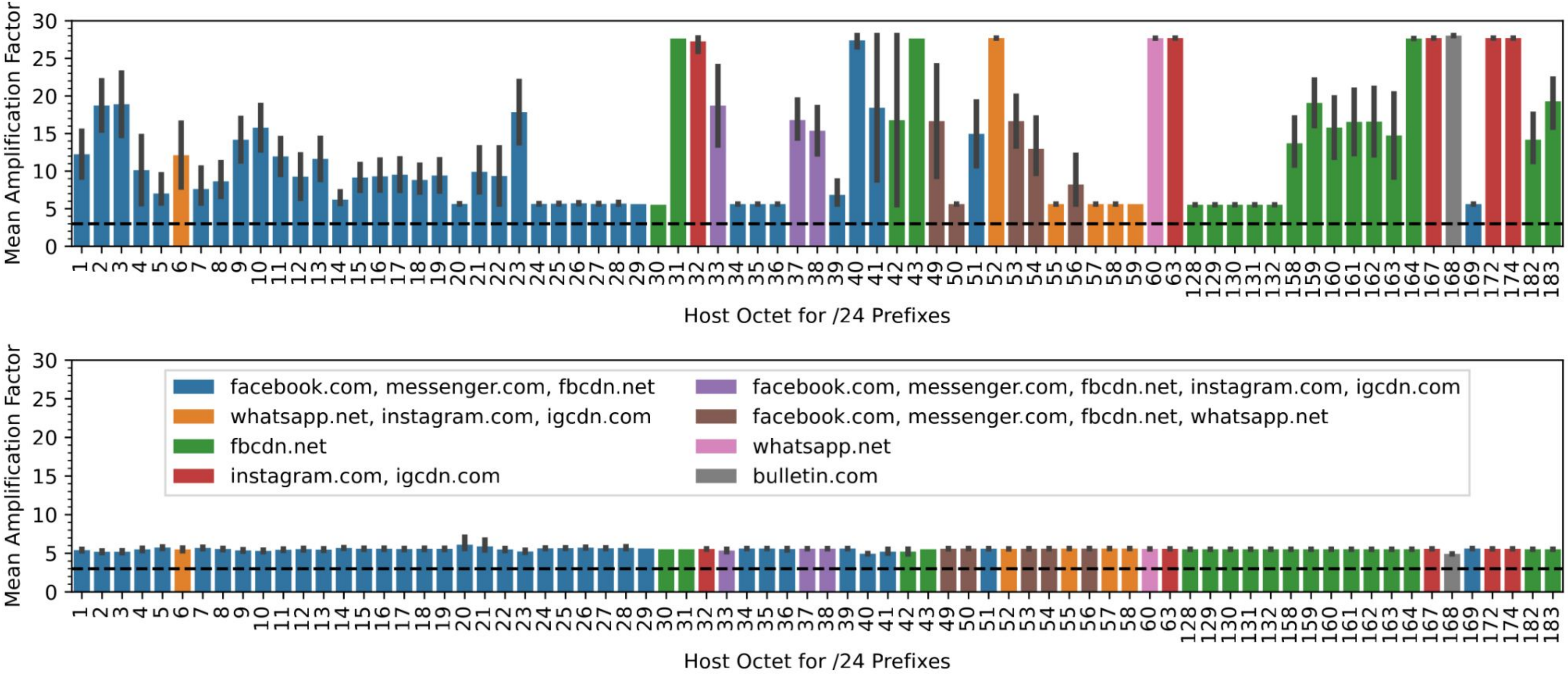
https://arxiv.org/pdf/2209.00965

Backup

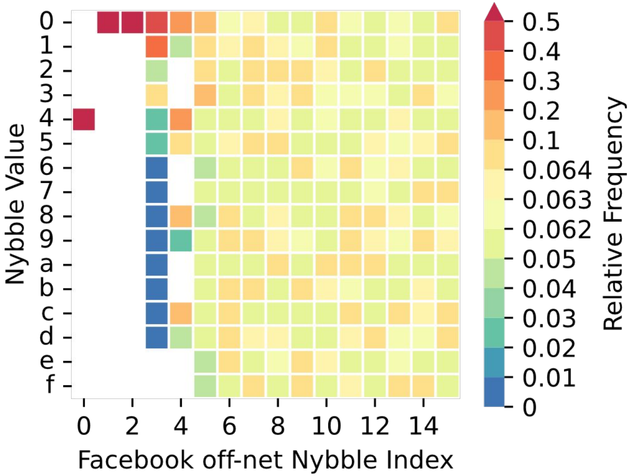
SCID structure of Facebook off-net servers

	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

Facebook Amplification Factors per Service

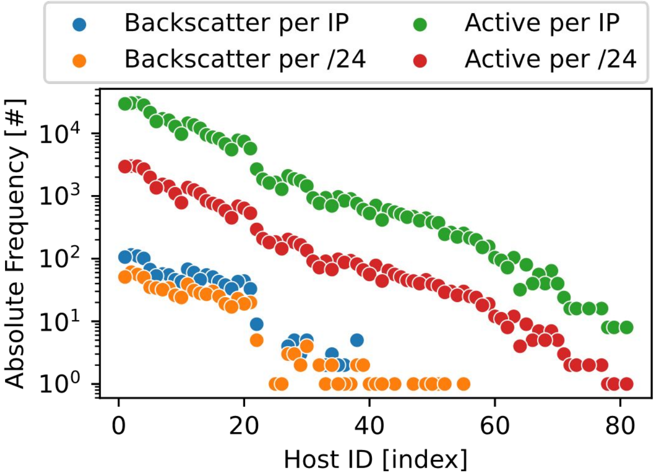


SCID structure of Facebook off-net servers



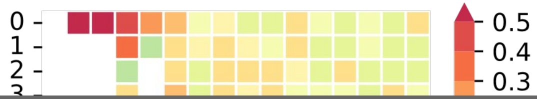
Heatmap of SCIDs of Facebook Off-net Deployments in 2022 Backscatter Traffic.

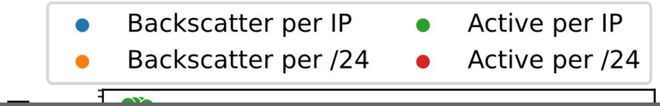
Facebook off-net servers use host IDs < 83.



Host ID Usage of Facebook Off-net Deployments in 2022 Backscatter Traffic and Enumeration Measurement.

SCID structure of Facebook off-net servers





We can use the first 9 bits of off-net host IDs for off-net detection!

Heatmap of SCIDs of Facebook Off-net Deployments in 2022 Backscatter Traffic.

Facebook off-net servers use host IDs < 83.

Host ID Usage of Facebook Off-net Deployments in 2022 Backscatter Traffic and Enumeration Measurement.

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

Cloudflare and Google enable packet coalescing.
Facebook does not.

What is in the data set?

January 1-31, 2022:

1655 Google IP addresses (1.3%)

246 Facebook IP addresses (8.3%)

78 Cloudflare IP addresses (0.01%)

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.

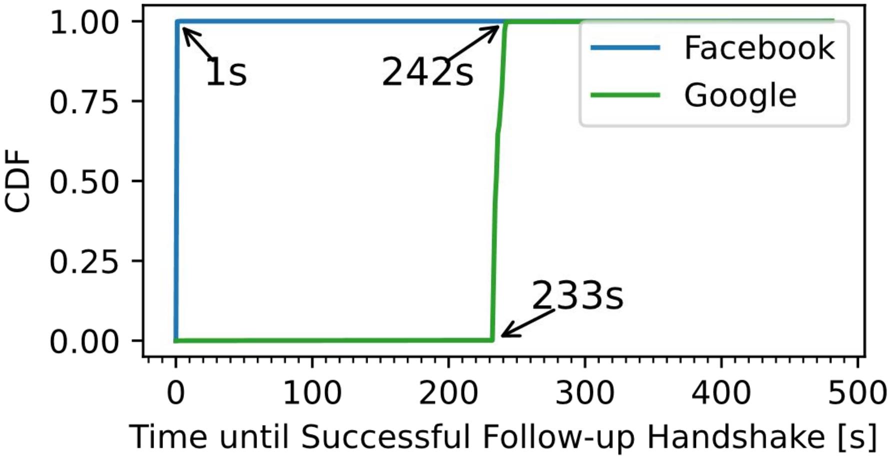
Facebook and Google use different load balancing methods

Google uses CID-aware load balancing.

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

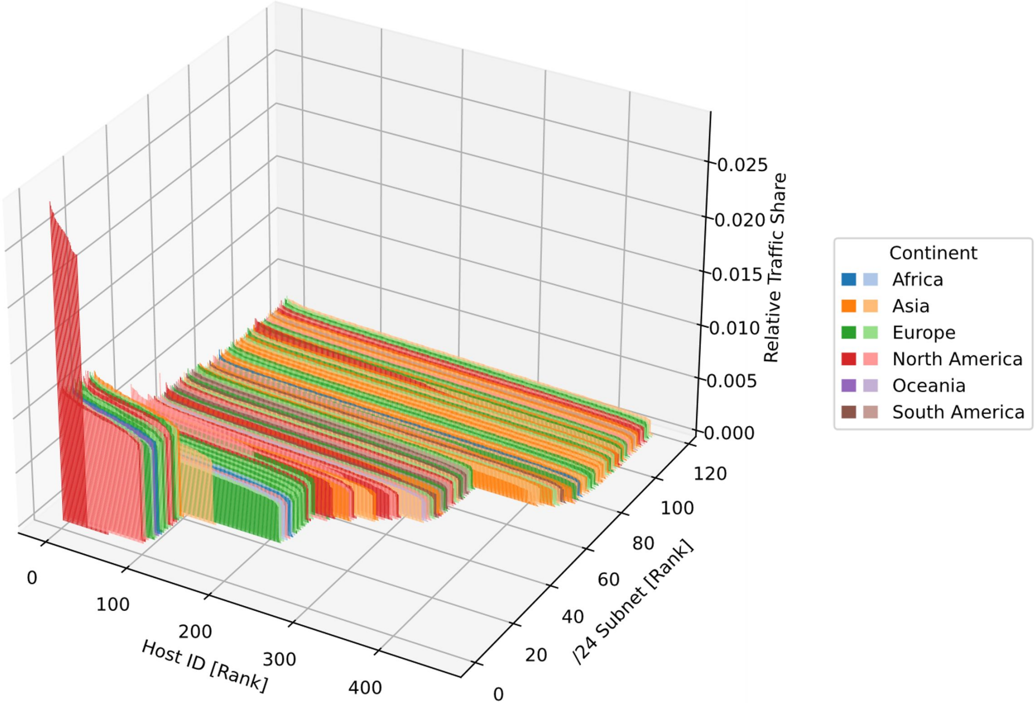
Facebook uses 5-tuple routing.

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



Facebook frontend clusters: Load balancer fairness

Nearly equal Distribution of Traffic to Host IDs per Cluster.


Relative Traffic Share
Host ID [Rank]
/24 Subnet [Rank]
Continent
Africa
Asia
Europe
North America
Oceania
South America