Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Freie Universität Berlin

TECHNISCHE UNIVERSITÄT DRESDEN

Listening to the noise: Understanding QUIC deployments using passive measurements

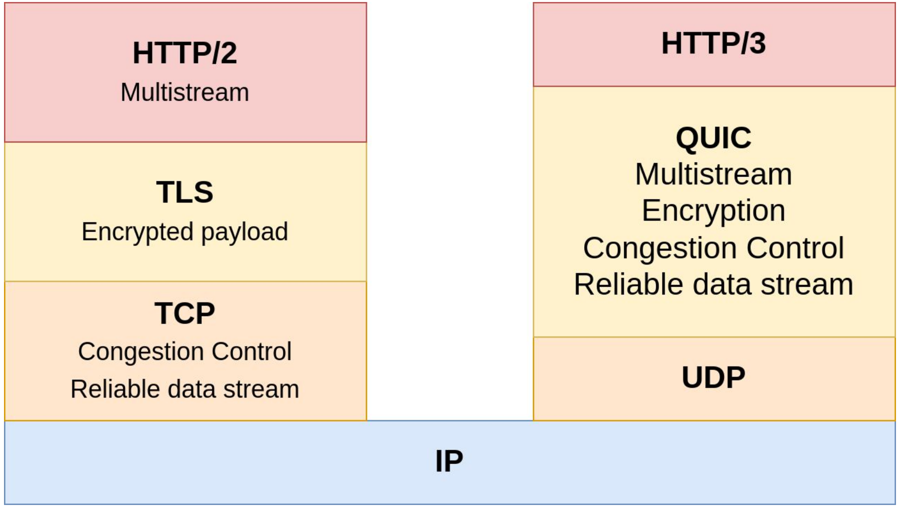
Jonas Mücke, Marcin Nawrocki, Raphael Hiesgen, Patrick Sattler, Johannes Zirngibl, Georg Carle, Thomas C. Schmidt, Matthias Wählisch

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

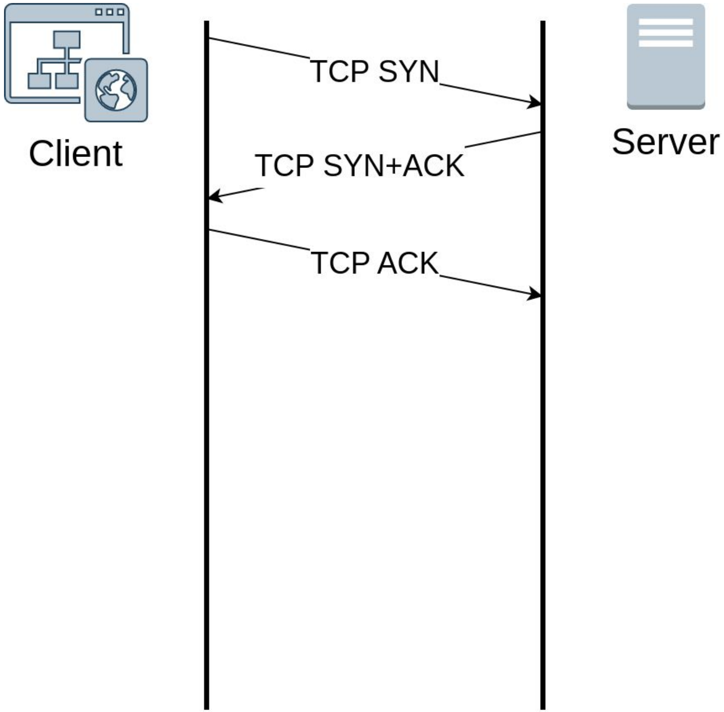
What is QUIC?

- A new transport protocol
- UDP based but implements reliability and congestion control
- Privacy-friendly and encryption built-in

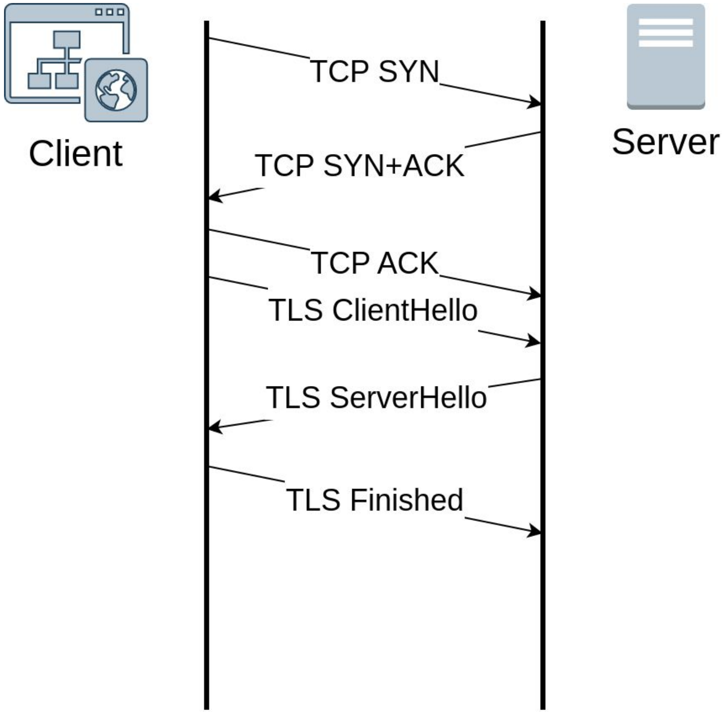
One of the motivations of QUIC is to prevent ossification of the transport layer by hiding as much meta data as possible.



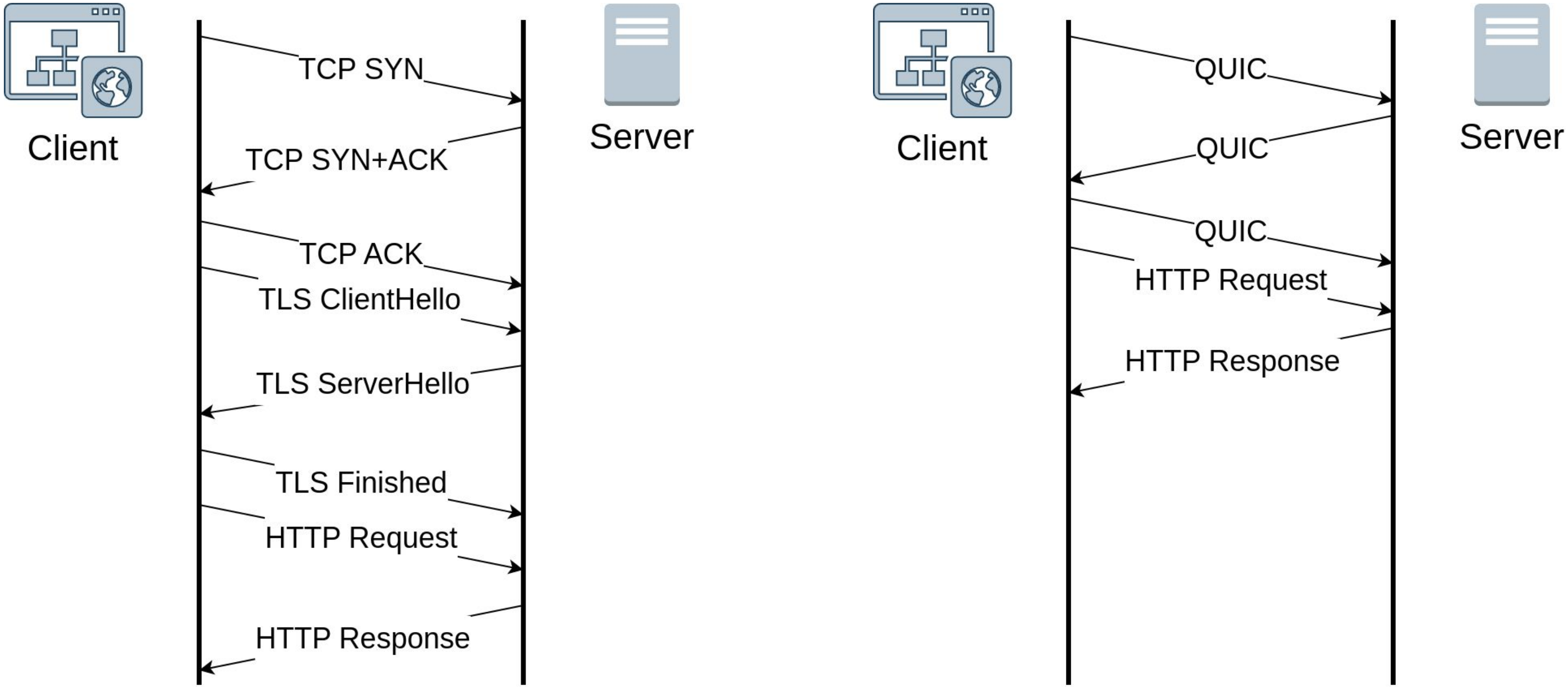
Why is QUIC faster?



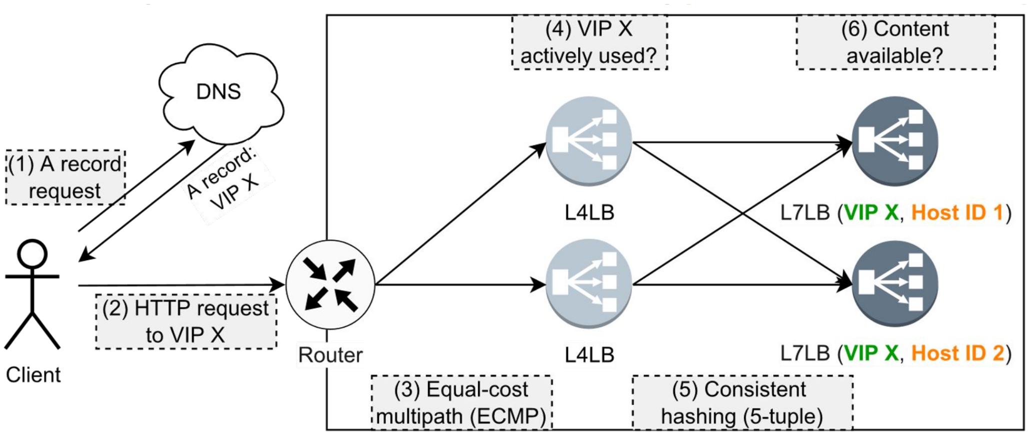
Why is QUIC faster?



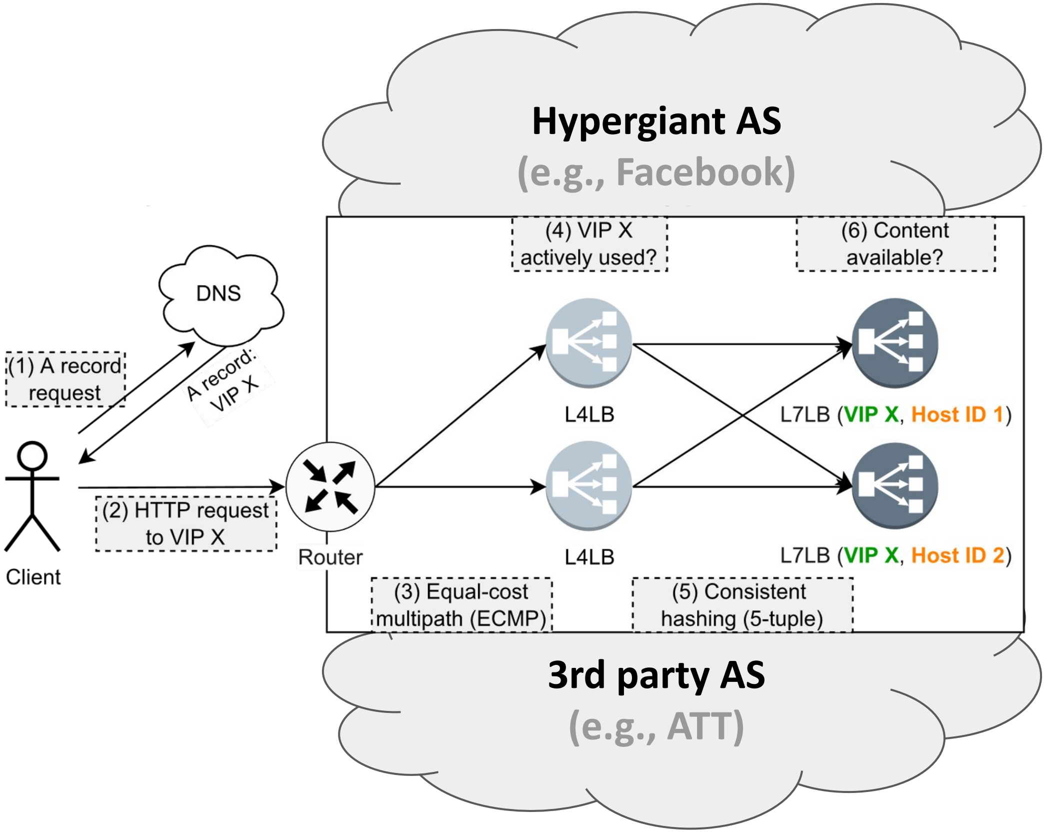
Why is QUIC faster? Combining handshakes.



Common hypergiant deployments



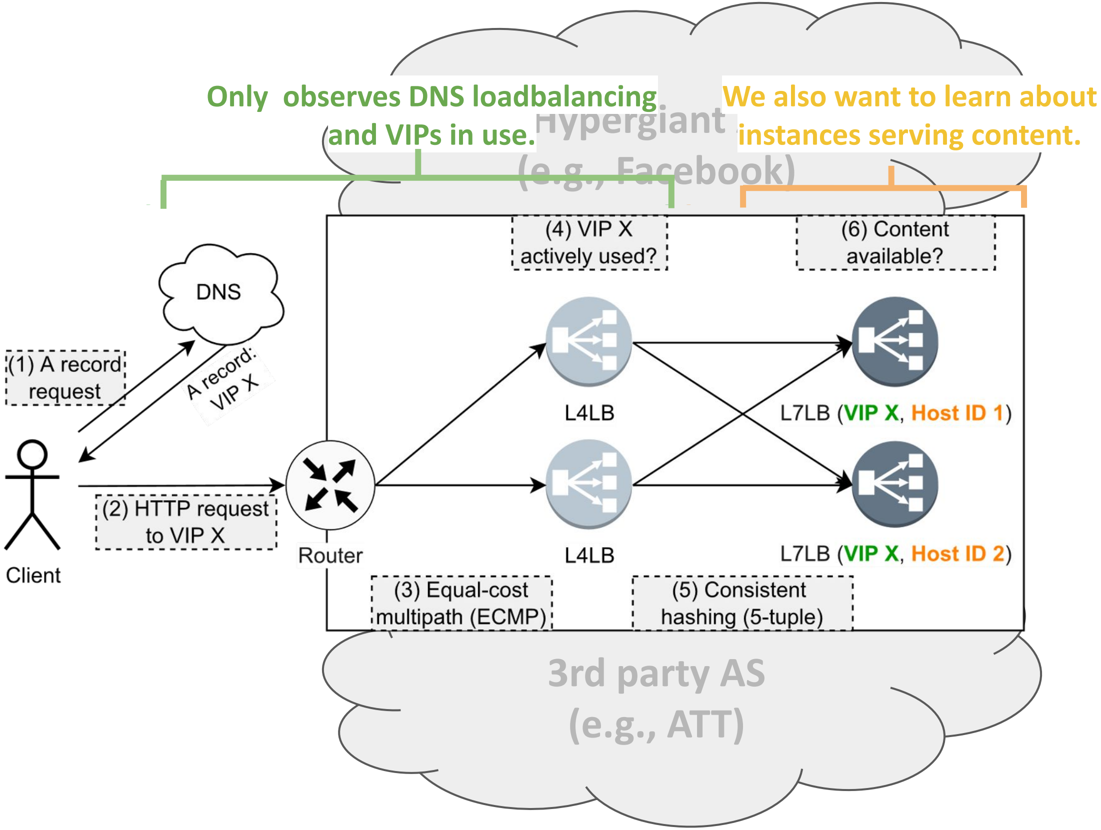
Common hypergiant deployments



On-net deployment

Off-net deployment

Prior work focused on active measurements



Scan for QUIC services, fetch TLS certificates etc.

The beauty of passive measurements

Passive measurements are non-intrusive.
You wait for incoming data and analyze.
Reduces measurement footprint.
A competitor or customer will not know about it.

Just to remind us … an excerpt from NANOG

INBOX - Alpine

File Edit Go Message Send Config Help

ALPINE 2.00 MESSAGE TEXT Folder: INBOX Message 104,839 of 108,030 ALL +

Date: Sun, 19 Jun 2022 03:13:32 -0700
From: Ronald F. Guilmette <rfg@tristatelogic.com>
To: nanog@nanog.org
Subject: Scanning the Internet for Vulnerabilities

I would like to solicit the opinions of network operators on the practice
of scanning all of, or large chunks of the internet for known vulnerabilities.

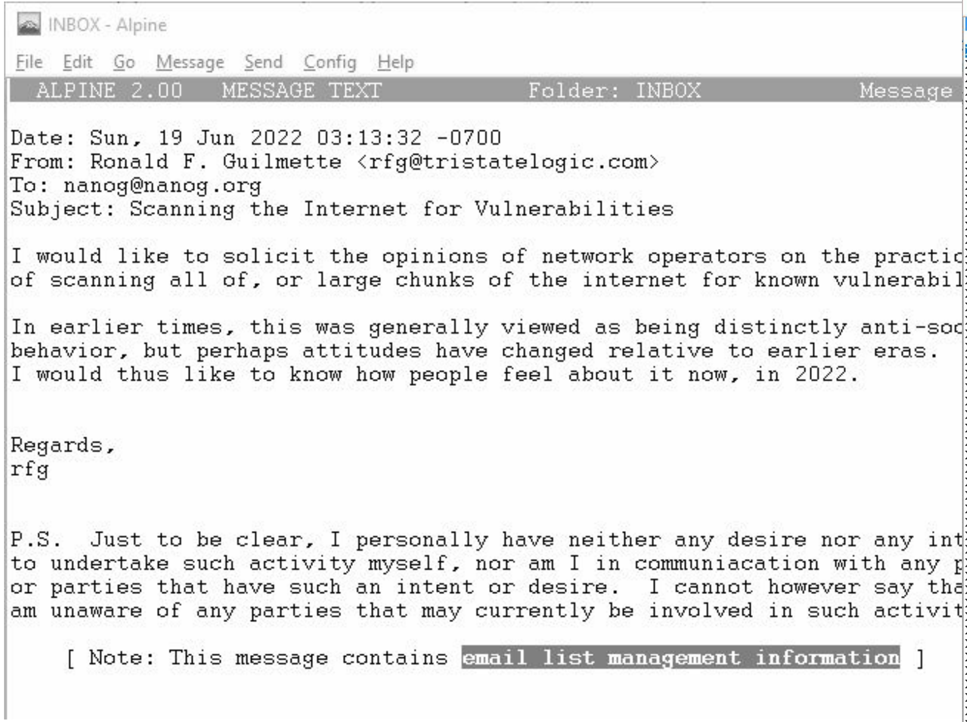
In earlier times, this was generally viewed as being distinctly anti-social
behavior, but perhaps attitudes have changed relative to earlier eras.
I would thus like to know how people feel about it now, in 2022.

Regards,
rfg

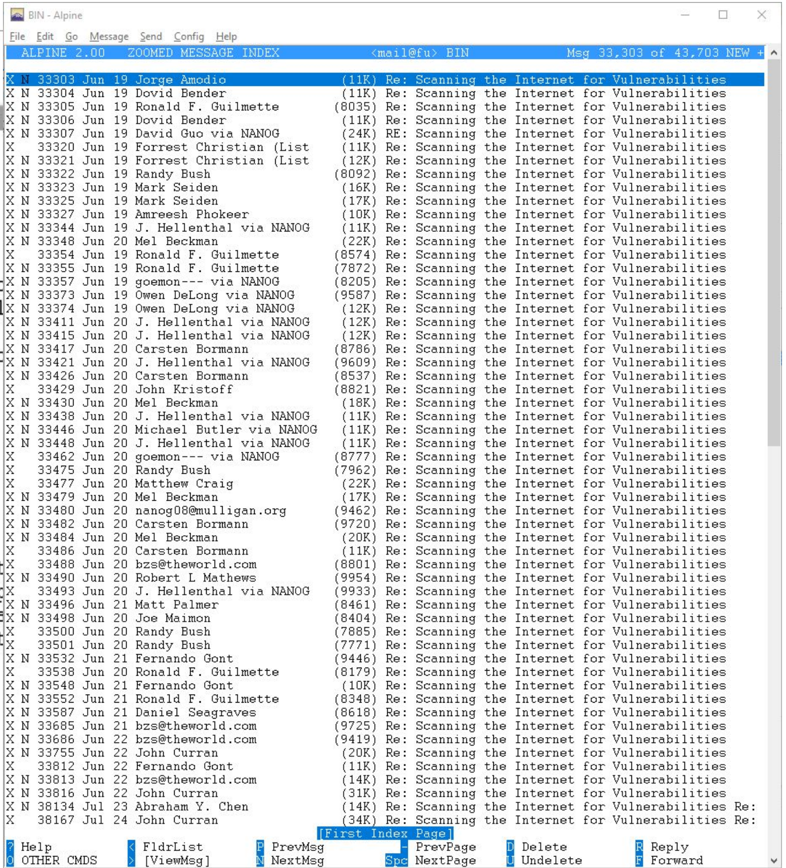
P.S. Just to be clear, I personally have neither any desire nor any intent
to undertake such activity myself, nor am I in communiacation with any party
or parties that have such an intent or desire. I cannot however say that I
am unaware of any parties that may currently be involved in such activities.

[Note: This message contains email list management information]

Just to remind us … an excerpt from NANOG



```
INBOX - Alpine
File  Edit  Go  Message  Send  Config  Help
  ALPINE 2.00   MESSAGE TEXT            Folder: INBOX            Message

Date: Sun, 19 Jun 2022 03:13:32 -0700
From: Ronald F. Guilmette <rfg@tristatelogic.com>
To: nanog@nanog.org
Subject: Scanning the Internet for Vulnerabilities

I would like to solicit the opinions of network operators on the practic
of scanning all of, or large chunks of the internet for known vulnerabil

In earlier times, this was generally viewed as being distinctly anti-soc
behavior, but perhaps attitudes have changed relative to earlier eras.
I would thus like to know how people feel about it now, in 2022.


Regards,
rfg


P.S.  Just to be clear, I personally have neither any desire nor any int
to undertake such activity myself, nor am I in communiacation with any p
or parties that have such an intent or desire.  I cannot however say tha
am unaware of any parties that may currently be involved in such activit

   [ Note: This message contains email list management information ]
```



```
BIN - Alpine
File  Edit  Go  Message  Send  Config  Help
  ALPINE 2.00   ZOOMED MESSAGE INDEX          <mail@fu> BIN        Msg 33,303 of 43,703 NEW +

X N 33303 Jun 19 Jorge Amodio               (11K) Re: Scanning the Internet for Vulnerabilities
X N 33304 Jun 19 Dovid Bender               (11K) Re: Scanning the Internet for Vulnerabilities
X N 33305 Jun 19 Ronald F. Guilmette       (8035) Re: Scanning the Internet for Vulnerabilities
X N 33306 Jun 19 Dovid Bender               (11K) Re: Scanning the Internet for Vulnerabilities
X N 33307 Jun 19 David Guo via NANOG        (24K) RE: Scanning the Internet for Vulnerabilities
X   33320 Jun 19 Forrest Christian (List    (11K) Re: Scanning the Internet for Vulnerabilities
X N 33321 Jun 19 Forrest Christian (List    (12K) Re: Scanning the Internet for Vulnerabilities
X N 33322 Jun 19 Randy Bush                (8092) Re: Scanning the Internet for Vulnerabilities
X N 33323 Jun 19 Mark Seiden                (16K) Re: Scanning the Internet for Vulnerabilities
X N 33325 Jun 19 Mark Seiden                (17K) Re: Scanning the Internet for Vulnerabilities
X N 33327 Jun 19 Amreesh Phokeer            (10K) Re: Scanning the Internet for Vulnerabilities
X N 33344 Jun 19 J. Hellenthal via NANOG    (11K) Re: Scanning the Internet for Vulnerabilities
X N 33348 Jun 20 Mel Beckman                (22K) Re: Scanning the Internet for Vulnerabilities
X   33354 Jun 19 Ronald F. Guilmette       (8574) Re: Scanning the Internet for Vulnerabilities
X N 33355 Jun 19 Ronald F. Guilmette       (7872) Re: Scanning the Internet for Vulnerabilities
X N 33357 Jun 19 goemon--- via NANOG       (8205) Re: Scanning the Internet for Vulnerabilities
X N 33373 Jun 19 Owen DeLong via NANOG     (9587) Re: Scanning the Internet for Vulnerabilities
X N 33374 Jun 19 Owen DeLong via NANOG      (12K) Re: Scanning the Internet for Vulnerabilities
X N 33411 Jun 20 J. Hellenthal via NANOG    (12K) Re: Scanning the Internet for Vulnerabilities
X N 33415 Jun 20 J. Hellenthal via NANOG    (12K) Re: Scanning the Internet for Vulnerabilities
X N 33417 Jun 20 Carsten Bormann           (8786) Re: Scanning the Internet for Vulnerabilities
X N 33421 Jun 20 J. Hellenthal via NANOG   (9609) Re: Scanning the Internet for Vulnerabilities
X N 33426 Jun 20 Carsten Bormann           (8537) Re: Scanning the Internet for Vulnerabilities
X   33429 Jun 20 John Kristoff             (8821) Re: Scanning the Internet for Vulnerabilities
X N 33430 Jun 20 Mel Beckman                (18K) Re: Scanning the Internet for Vulnerabilities
X N 33438 Jun 20 J. Hellenthal via NANOG    (11K) Re: Scanning the Internet for Vulnerabilities
X N 33446 Jun 20 Michael Butler via NANOG   (11K) Re: Scanning the Internet for Vulnerabilities
X N 33448 Jun 20 J. Hellenthal via NANOG    (11K) Re: Scanning the Internet for Vulnerabilities
X   33462 Jun 20 goemon--- via NANOG       (8777) Re: Scanning the Internet for Vulnerabilities
X   33475 Jun 20 Randy Bush                (7962) Re: Scanning the Internet for Vulnerabilities
X   33477 Jun 20 Matthew Craig              (22K) Re: Scanning the Internet for Vulnerabilities
X N 33479 Jun 20 Mel Beckman                (17K) Re: Scanning the Internet for Vulnerabilities
X N 33480 Jun 20 nanog08@mulligan.org      (9462) Re: Scanning the Internet for Vulnerabilities
X N 33482 Jun 20 Carsten Bormann           (9720) Re: Scanning the Internet for Vulnerabilities
X N 33484 Jun 20 Mel Beckman                (20K) Re: Scanning the Internet for Vulnerabilities
X   33486 Jun 20 Carsten Bormann            (11K) Re: Scanning the Internet for Vulnerabilities
X   33488 Jun 20 bzs@theworld.com          (8801) Re: Scanning the Internet for Vulnerabilities
X N 33490 Jun 20 Robert L Mathews          (9954) Re: Scanning the Internet for Vulnerabilities
X   33493 Jun 20 J. Hellenthal via NANOG   (9933) Re: Scanning the Internet for Vulnerabilities
X N 33496 Jun 21 Matt Palmer               (8461) Re: Scanning the Internet for Vulnerabilities
X N 33498 Jun 20 Joe Maimon                (8404) Re: Scanning the Internet for Vulnerabilities
X   33500 Jun 20 Randy Bush                (7885) Re: Scanning the Internet for Vulnerabilities
X   33501 Jun 20 Randy Bush                (7771) Re: Scanning the Internet for Vulnerabilities
X N 33532 Jun 21 Fernando Gont             (9446) Re: Scanning the Internet for Vulnerabilities
X   33538 Jun 20 Ronald F. Guilmette       (8179) Re: Scanning the Internet for Vulnerabilities
X N 33548 Jun 21 Fernando Gont              (10K) Re: Scanning the Internet for Vulnerabilities
X N 33552 Jun 21 Ronald F. Guilmette       (8348) Re: Scanning the Internet for Vulnerabilities
X N 33587 Jun 21 Daniel Seagraves          (8618) Re: Scanning the Internet for Vulnerabilities
X N 33685 Jun 21 bzs@theworld.com          (9725) Re: Scanning the Internet for Vulnerabilities
X N 33686 Jun 22 bzs@theworld.com          (9419) Re: Scanning the Internet for Vulnerabilities
X N 33755 Jun 22 John Curran                (20K) Re: Scanning the Internet for Vulnerabilities
X   33812 Jun 22 Fernando Gont              (11K) Re: Scanning the Internet for Vulnerabilities
X N 33813 Jun 22 bzs@theworld.com           (14K) Re: Scanning the Internet for Vulnerabilities
X N 33816 Jun 22 John Curran                (31K) Re: Scanning the Internet for Vulnerabilities
X N 38134 Jul 23 Abraham Y. Chen            (14K) Re: Scanning the Internet for Vulnerabilities Re:
X   38167 Jul 24 John Curran                (34K) Re: Scanning the Internet for Vulnerabilities Re:
                                    [First Index Page]
? Help          < FldrList     P PrevMsg     - PrevPage   D Delete     R Reply
O OTHER CMDS    > [ViewMsg]    N NextMsg   Spc NextPage   U Undelete   F Forward
```

What do we want to achieve?

Identifying servers of specific hypergiants
Identifying off-net servers
Identifying L7 load balancers

Why is this interesting for RIPE?

Inter-domain replication between caches
Unexpected traffic of peers

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.

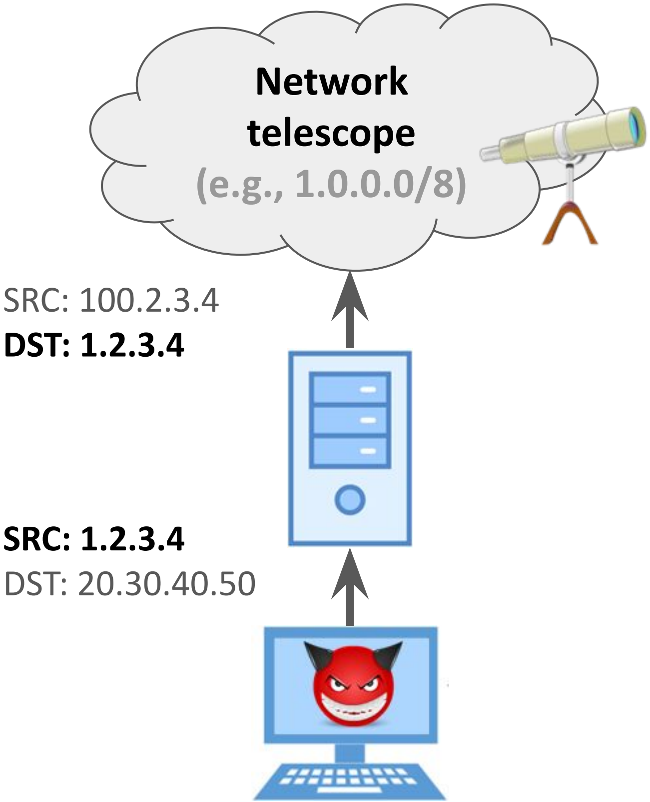
Analyze QUIC backscatter traffic.

Why QUIC?

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

Exposes additional information
(compared to UDP and TCP).

What is backscatter?



Backscatter is response traffic to IP packets with incorrect source IP address.

The source IP address is often randomly generated.

Why does random IP spoofing occur?

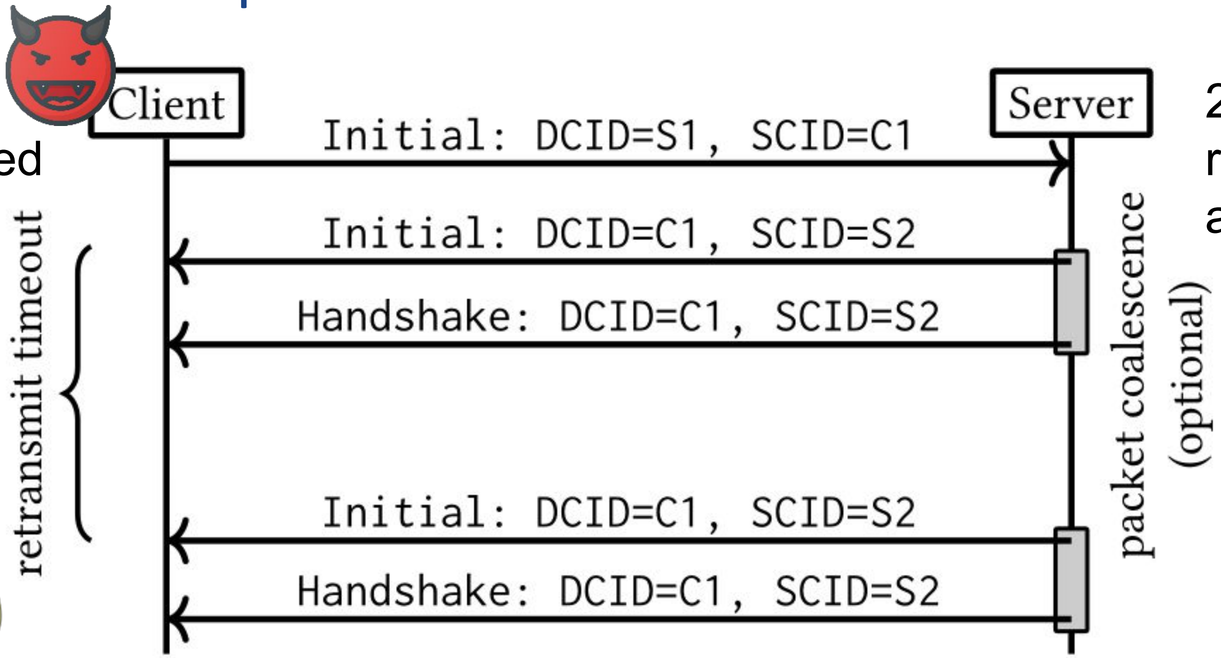
DDoS attacks leveraging state exhaustion

How is backscatter collected?

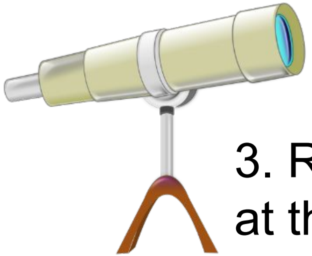
Network telescopes, address space waiting for incoming traffic

Measurement setup

1. Attacker sends spoofed packet.



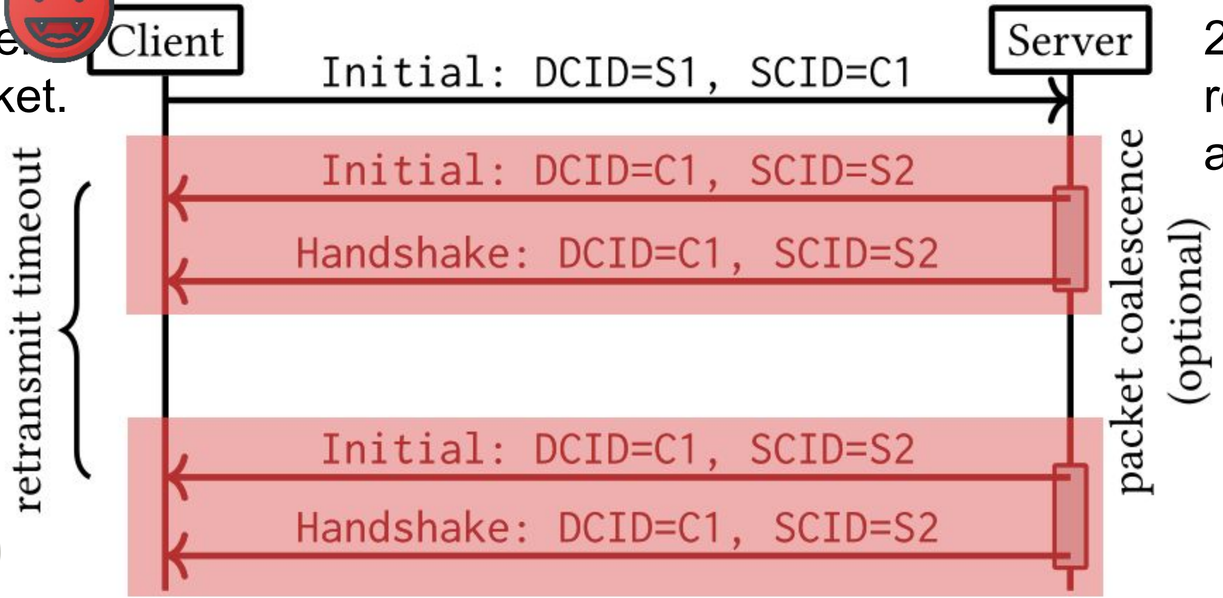
2. Server sends reply to spoofed address.



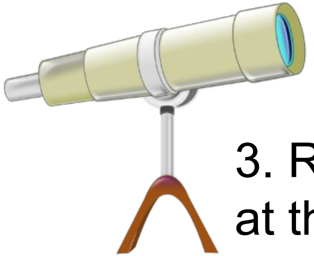
3. Reception of response traffic at the network telescope.

Measurement setup

1. Attacker se
spoofed packet.



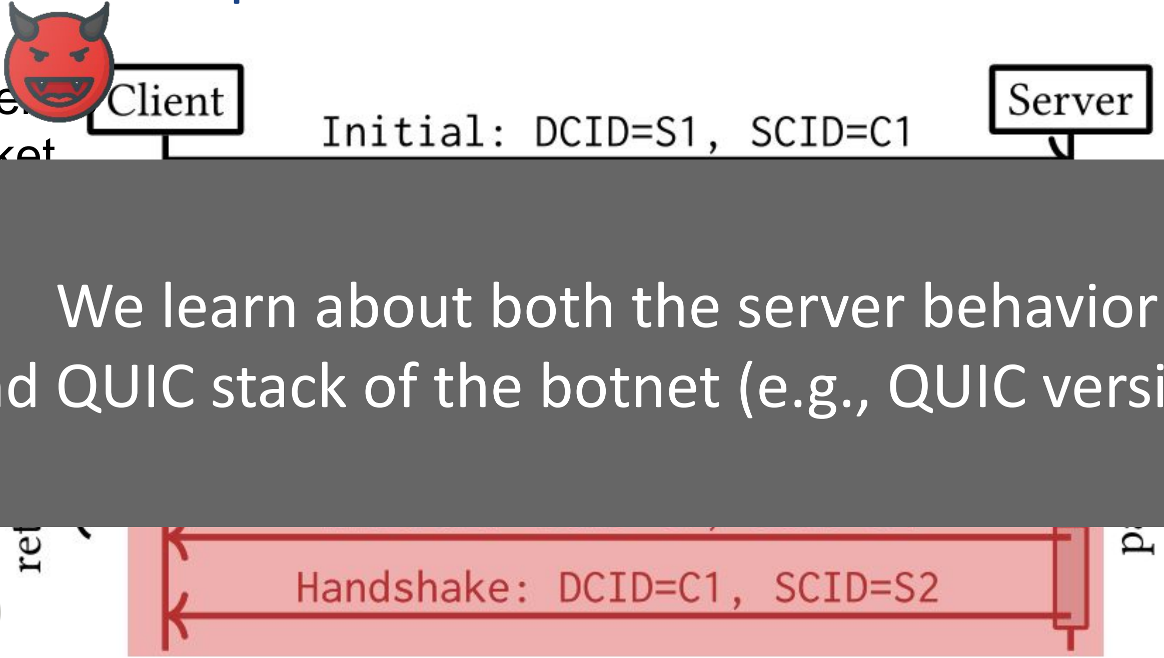
2. Server sends reply to spoofed address.



3. Reception of response traffic at the network telescope.

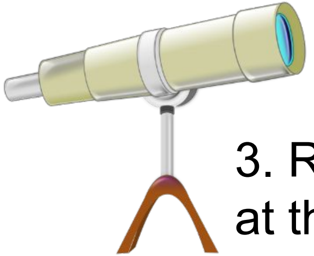
Measurement setup

1. Attacker se
spoofed packet



2. Server sends
reply to spoofed

We learn about both the server behavior
and QUIC stack of the botnet (e.g., QUIC version).



3. Reception of response traffic
at the network telescope.

Measurement setup

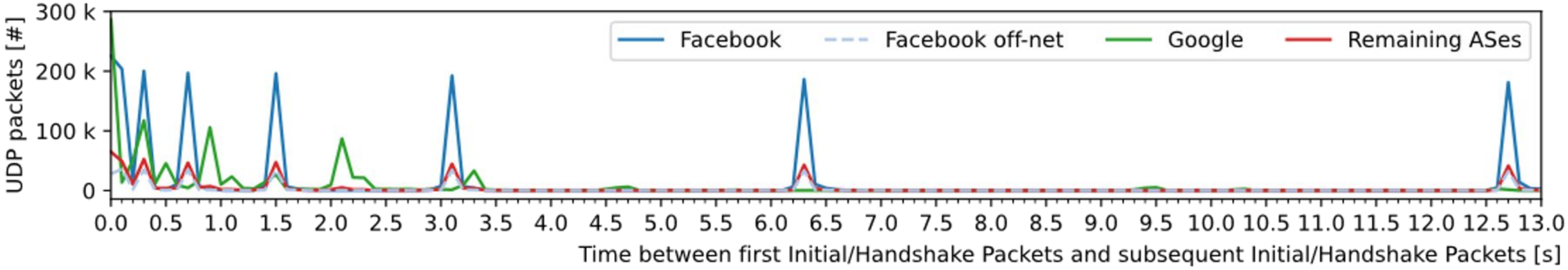
Passive measurements using the CAIDA /9 IPv4 network telescope

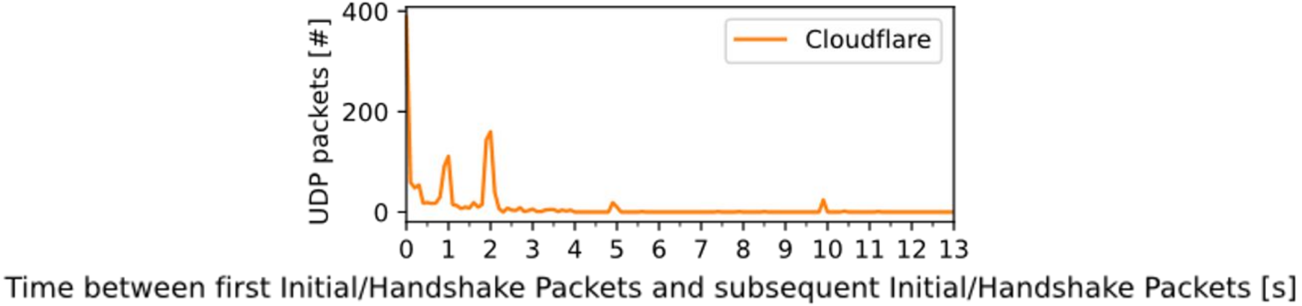
- January 1-31, 2022

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

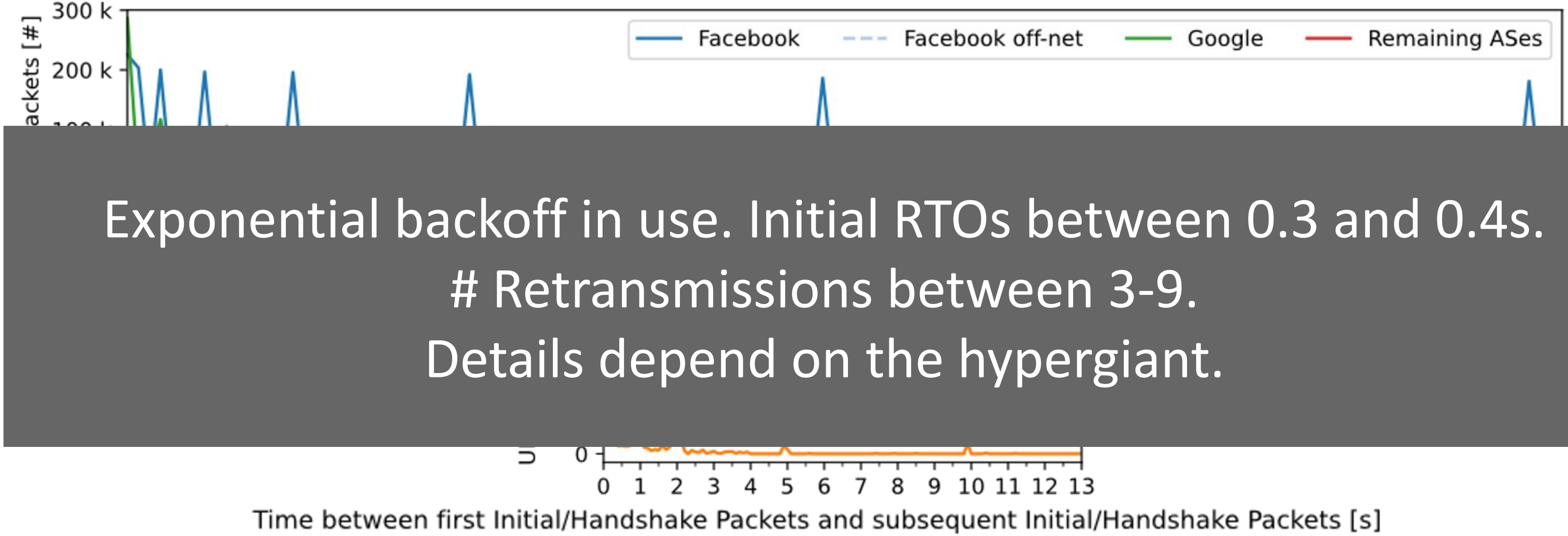
- QUIC scans
- TLS scans
- DNS scans

Inter-arrival times of Initial/Handshakes packets not answered





Inter-arrival times of Initial/Handshakes packets not answered



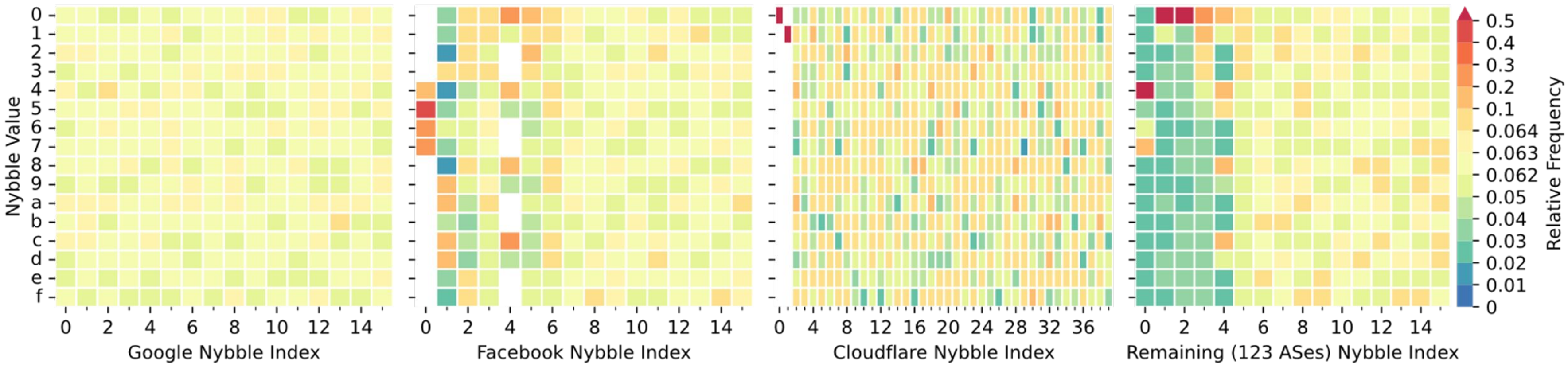
Exponential backoff in use. Initial RTOs between 0.3 and 0.4s.
Retransmissions between 3-9.
Details depend on the hypergiant.

Structure of QUIC Server Connection IDs (SCIDs)

XXXXXXX…XXXXXXXXX max. length 20 Byte

(half Byte, Nybble) 0…f

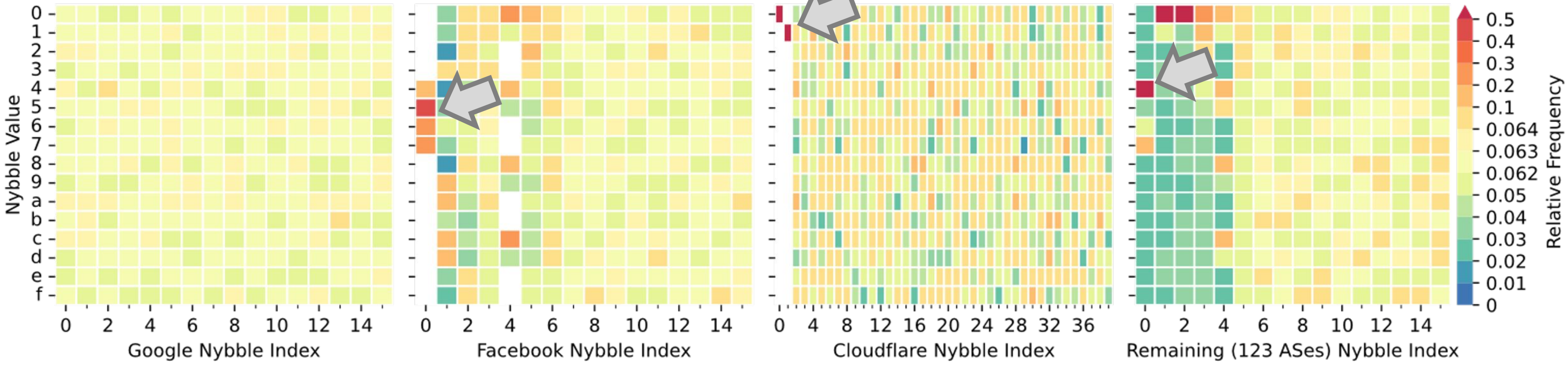
Structure of QUIC Server Connection IDs (SCIDs)



XXXXXXX…XXXXXXXXX max. length 20 Byte

(half Byte, Nybble) 0…f

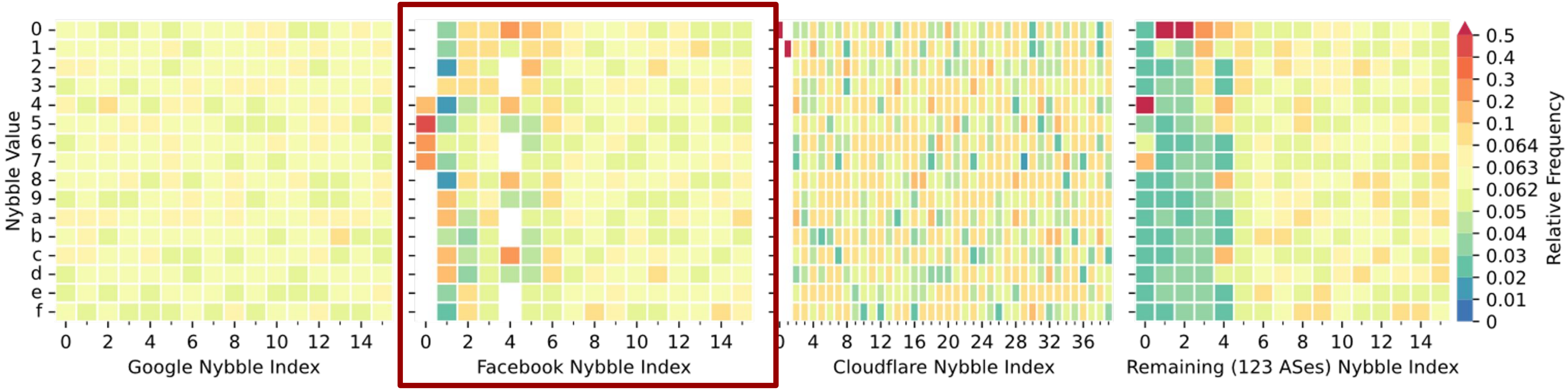
Structure of QUIC Server Connection IDs (SCIDs)



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

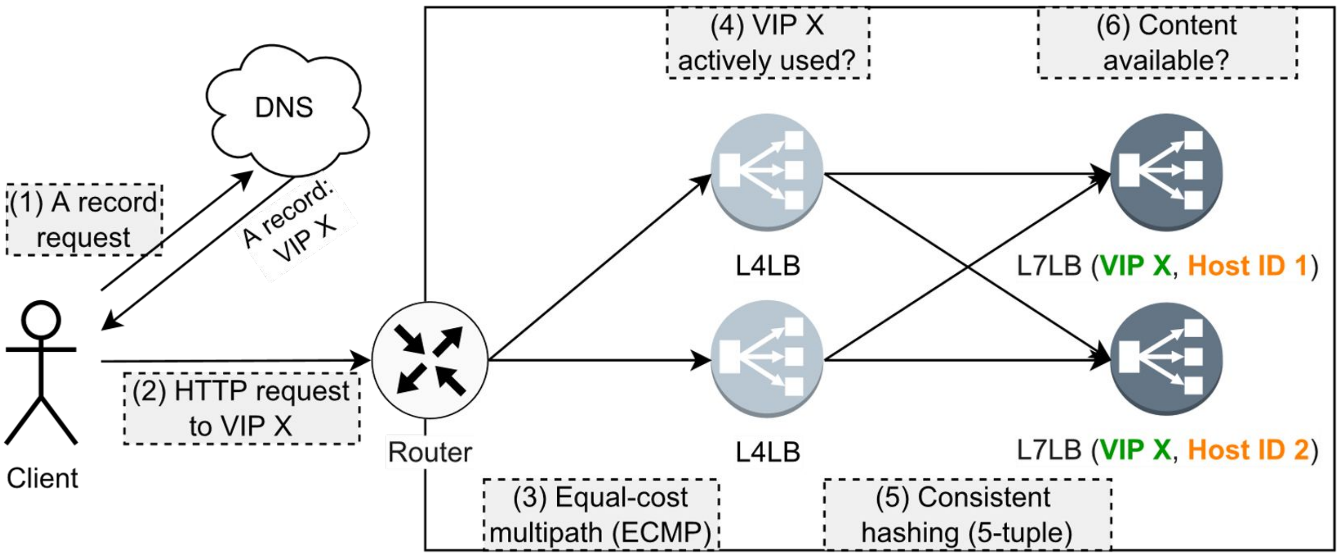
Facebook's SCID Structure according to their QUIC Implementation mvfst.

Detecting Facebook off-net servers

Verified by SAN in TLS certificates

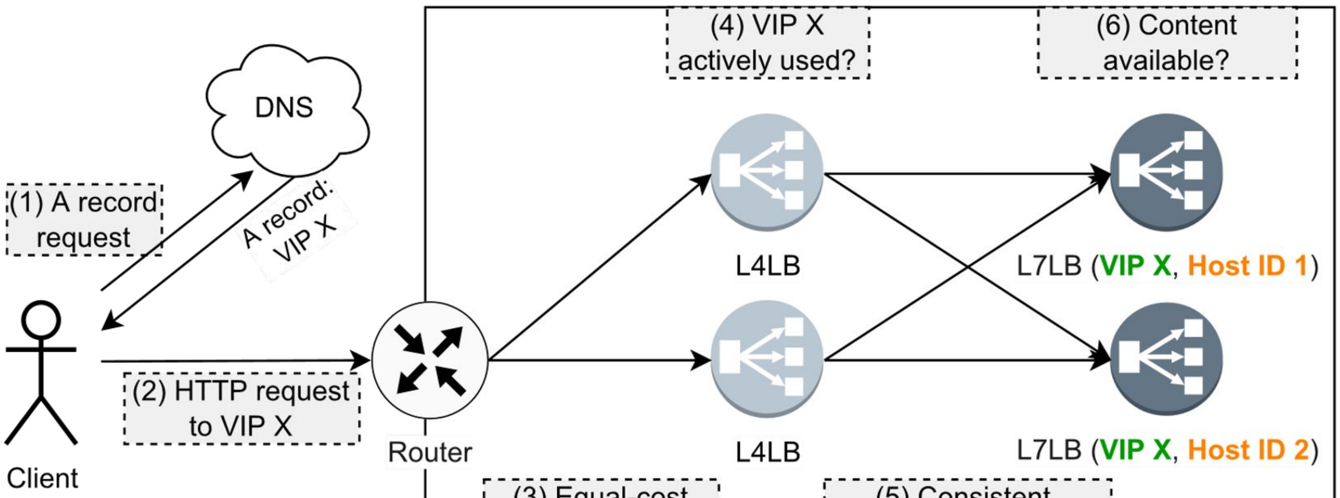
Classificator	TPR	FPR	TNR	FNR	Precision	Recall
Inter-Arrival Time (IAT)	0.772	0.268	0.732	0.228	0.645	0.772
SCID, IAT	0.772	0.046	0.954	0.228	0.914	0.772
Packet Length	0.997	0.328	0.672	0.003	0.657	0.997
Coalescence	1.000	0.931	0.069	0.000	0.403	1.000
SCID	**1.000**	**0.193**	**0.807**	**0.000**	**0.765**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

We collect the Server Connection IDs:

- 37k different Host IDs contained
- 19% are contained in the passive measurement data

The relation between VIPs and host IDs:

If one Host IDs is served from multiple VIPs they are assigned to the same cluster.

Exploring frontend clusters

We collect the Server Connection IDs:

- 37k different Host IDs contained
- 19% are contained in the passive

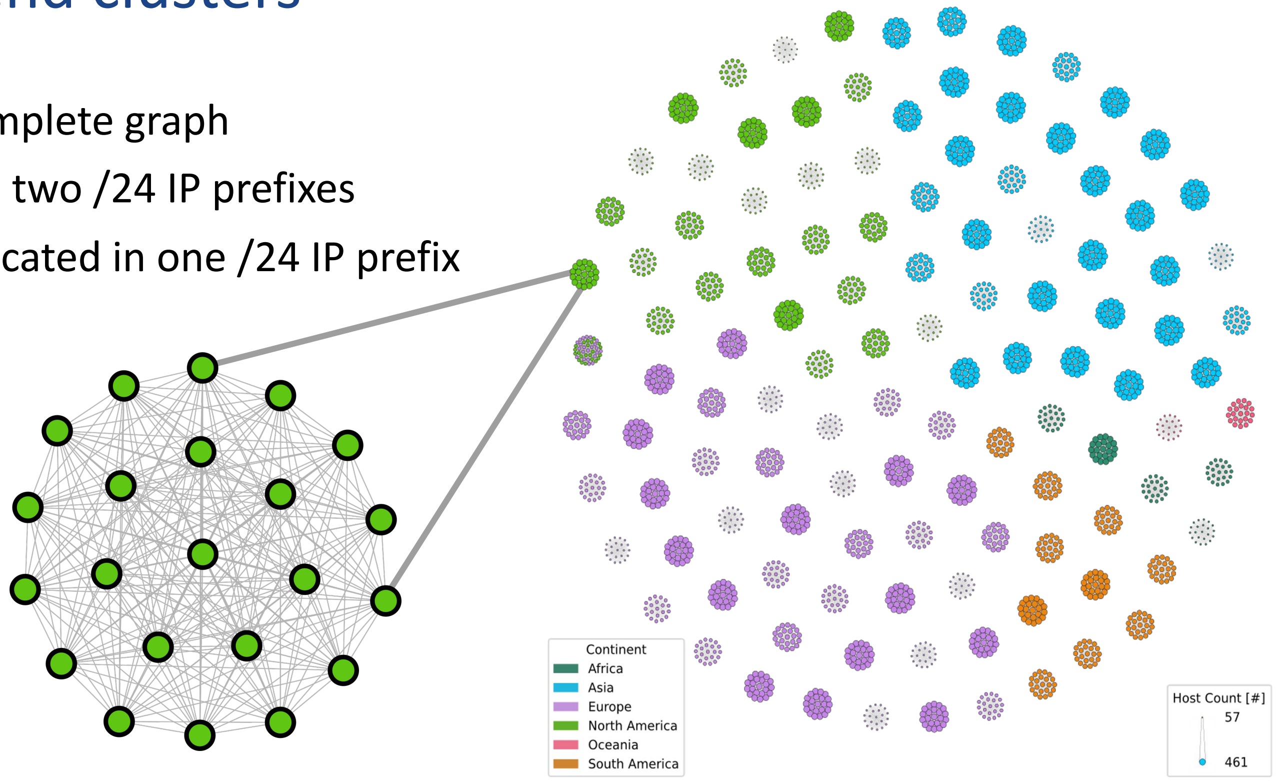
We detect 112 clusters using 22 VIPs and
3 clusters using 21, 20, and 44 VIPs.

Exploring frontend clusters

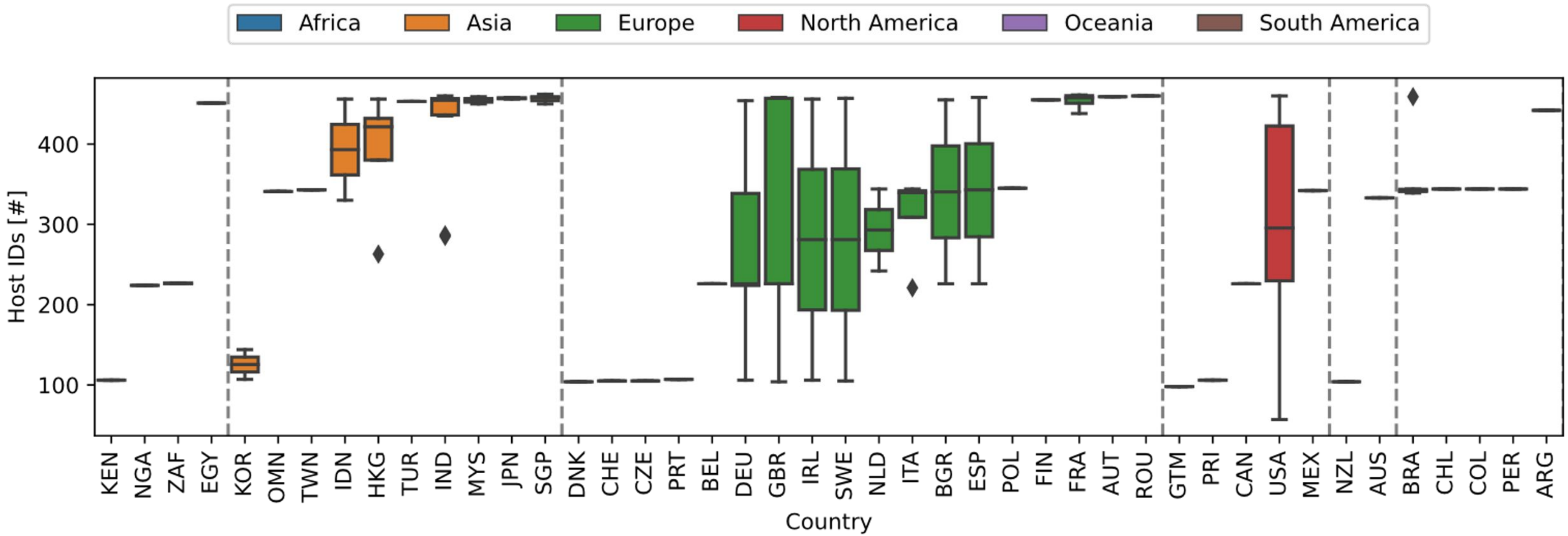
Each cluster forms a complete graph

One cluster is located in two /24 IP prefixes

All remaining clusters located in one /24 IP prefix

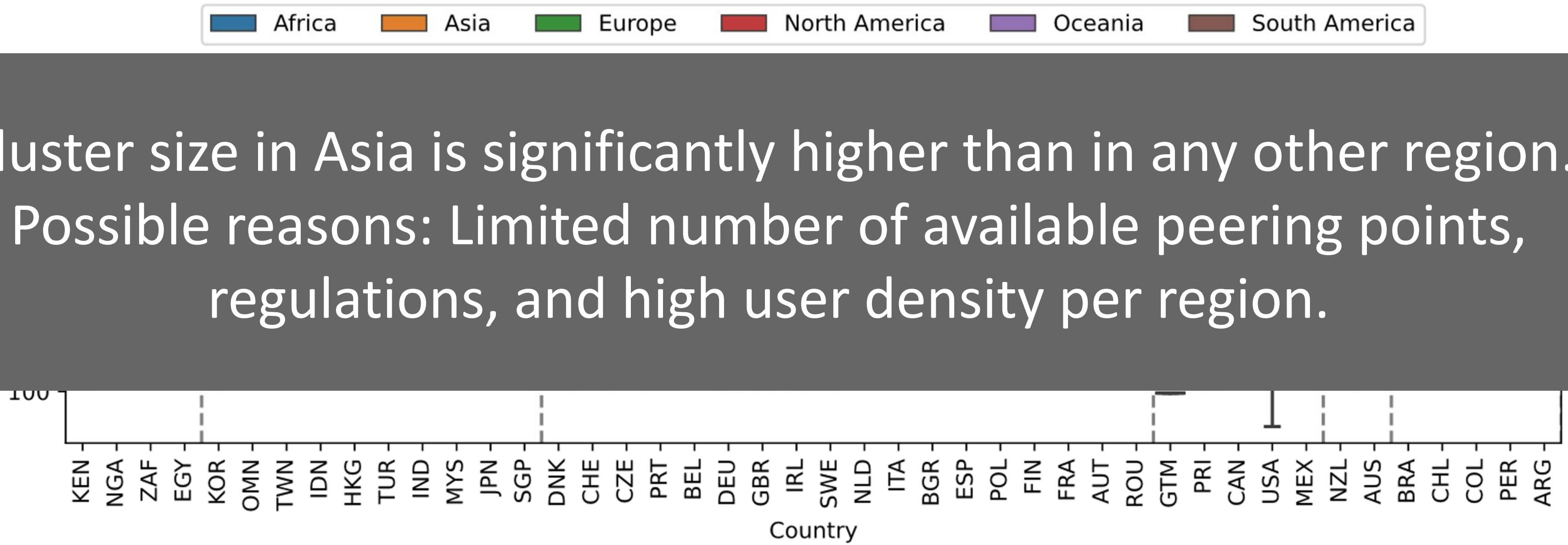


Facebook cluster sizes per country



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

Facebook cluster sizes per country



Cluster size in Asia is significantly higher than in any other region. Possible reasons: Limited number of available peering points, regulations, and high user density per region.

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

https://arxiv.org/pdf/2209.00965

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:

(1) We discuss the potential and need of information encoding in QUIC Connection IDs in large load balancer deployment scenarios. (§ 2)

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

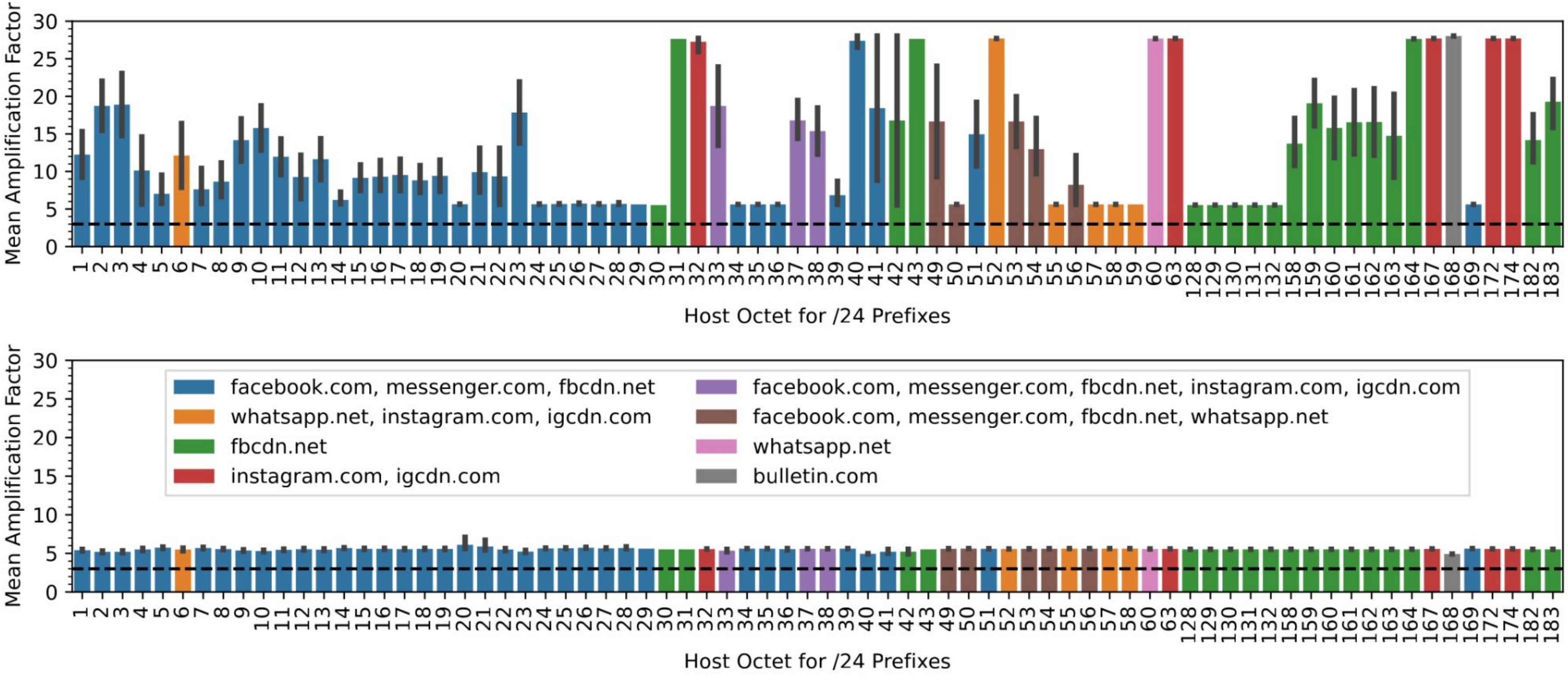
Connection setup. A common QUIC 1-RTT handshake is depicted in Figure 1. All QUIC sessions start with an `Initial` sent by a

Backup

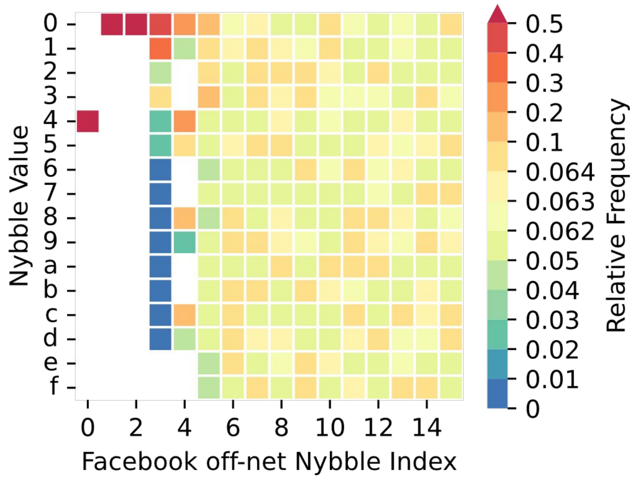
SCID structure of Facebook off-net servers

	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

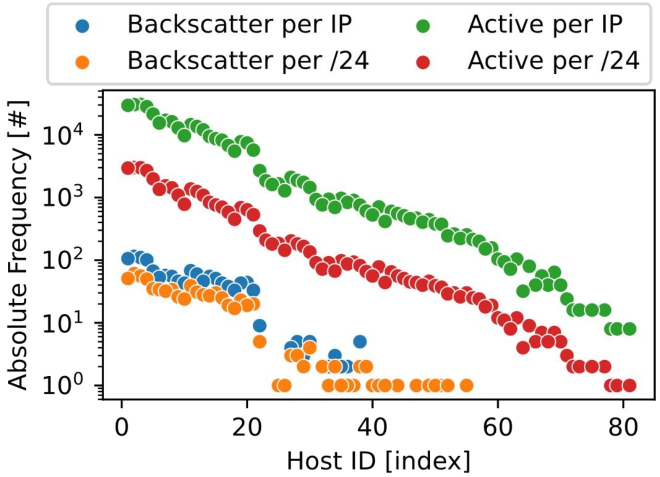


SCID structure of Facebook off-net servers



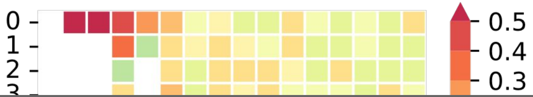
Heatmap of SCIDs of Facebook Off-net Deployments in 2022 Backscatter Traffic.

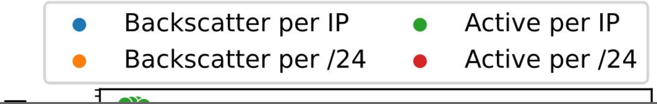
Facebook off-net servers use host IDs < 83.



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
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.

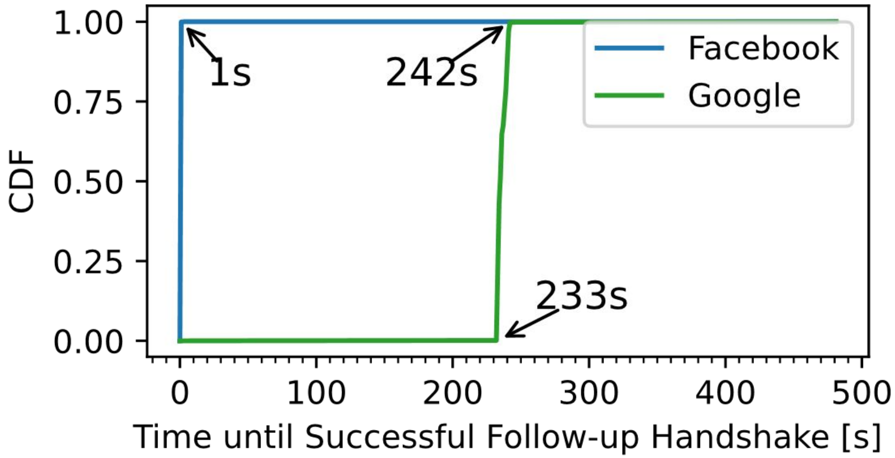
Facebook and Google use different load balancing methods

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.