Waiting for QUIC: Passive Measurements to Understand QUIC Deployments

JONAS MÜCKE, TU Dresden, Germany
MARCIN NAWROCKI, NETSCOUT, USA
RAPHAEL HIESGEN, HAW Hamburg, Germany
PATRICK SATTLER, Technical University of Munich, Germany
JOHANNES ZIRNGIBL, Max Planck Institute for Informatics, Germany
GEORG CARLE, Technical University of Munich, Germany
JAN LUXEMBURK, FIT CTU & CESNET, Czech Republic
THOMAS C. SCHMIDT, HAW Hamburg, Germany
MATTHIAS WÄHLISCH, TU Dresden, Germany

QUIC experiences a rapid adoption since its standardization in 2021, and hypergiants configure their infrastructure to optimize for QUIC performance. In this paper, we introduce a passive measurement method to study both the progressive rollout and individual hypergiant configurations during the last five years. By analyzing backscatter traffic of the UCSD network telescope, we are able to make the following observations. First, Meta, Google, and Cloudflare configure significantly different maximal retransmission numbers and timeouts. Second, we can identify different off-net deployments of hypergiants, using packet features, such as QUIC connection IDs, packet coalescence, and packet lengths. Third, we observe changing hypergiant deployment configurations during our different measurement periods. Fourth, connection IDs can allow further insights into load balancer deployments, such as the number of servers. We bolster our results using two orthogonal measurements: passive recording of QUIC flows and active probing.

CCS Concepts: • **Networks → Network measurement**; **Transport protocols**; **Logical / virtual topologies**.

Additional Key Words and Phrases: QUIC, hypergiant infrastructure, deployment analysis

ACM Reference Format:
Jonas Mücke, Marcin Nawrocki, Raphael Hiesgen, Patrick Sattler, Johannes Zirngibl, Georg Carle, Jan Luxemburk, Thomas C. Schmidt, and Matthias Wählisch. 2025. Waiting for QUIC: Passive Measurements to Understand QUIC Deployments. *Proc. ACM Netw.* 3, CoNEXT4, Article 41 (December 2025), 26 pages. https://doi.org/10.1145/3768988

1 Introduction

Measurement techniques for revealing the setups of large service providers, *i.e.,* hypergiants, are a long-standing research challenge [6, 23, 28, 39]. Detailed knowledge of hypergiant infrastructures is often considered business sensitive and may raise security concerns. Therefore, such knowledge is frequently hidden from the public. Nevertheless, insights from infrastructure deployments and

Authors' Contact Information: Jonas Mücke, TU Dresden, Dresden, Germany; Marcin Nawrocki, NETSCOUT, Westford, MA, USA; Raphael Hiesgen, HAW Hamburg, Hamburg, Germany; Patrick Sattler, Technical University of Munich, Munich, Germany; Johannes Zirngibl, Max Planck Institute for Informatics, Saarbrücken, Germany; Georg Carle, Technical University of Munich, Munich, Germany; Jan Luxemburk, FIT CTU & CESNET, Prague, Czech Republic; Thomas C. Schmidt, HAW Hamburg, Hamburg, Germany; Matthias Wählisch, TU Dresden, Dresden, Germany.

ACM 2834-5509/2025/12-ART41
https://doi.org/10.1145/3768988

Table 1. Information on QUIC deployments inferred from passive backscatter traffic in 2025.

		Hypergiant							
		Akamai	Amazon	Apple	Cloudflare	Fastly	Google	Meta	Microsoft
First backscatter visible		2023	2022	2022	2021	2023	2021	2021	2022
Features	Coalescence	✓	✓	✓	✓	✓	✓	✘	✓
	Structured SCIDs	✓	✓	✓	✓	✓	✓	✓	✓
	Retry observed	✘	✘	✘	✓	✘	✘	✘	✓
	L7 load balancers	n/a	n/a	n/a	n/a	n/a	n/a	✓	n/a
	SCID length	20 B	20 B	20 B	20 B	17 B	8 B	8 B	14/20 B
	Initial RTO	1 s	0.3 s	1 s	1 s	0.2 s	0.3 s	0.1 s	1 s
	Mean # retransmissions	2.1	4.0	2.7	1.5	6.0	3.4	7.5	1.3

protocol configurations of hypergiants may help improve the experience of Internet users by fine-tuning existing deployments and guiding the development of new protocols.

QUIC is adopted by hypergiants since 2021 [42, 52, 67], contributes a significant share of today's Internet traffic [11, 36], and continues to spread widely with HTTP/3 [5]. QUIC has been designed to improve performance [14, 57, 64] and to maximize privacy by disguising meta-information [62].

Prior research studied the deployment of QUIC using active measurements (*e.g.,* [52, 67]) or the performance and interoperability of QUIC in testbeds [33, 44, 47, 55]. Results are thus limited to lab conditions or raise operational and legal concerns due to interference with real world deployments [19, 25, 41, 53]. In this paper, we track QUIC configurations of hypergiants during the last five years using Internet backscatter traffic from the UCSD network telescope, a passive data source. Our method is non-intrusive and does not require beforehand information on deployments. We identify individual QUIC configurations of Akamai, Amazon, Apple, Cloudflare, Fastly, Google, Meta, and Microsoft and gain new insights into the load balancer infrastructure of Meta, summarized in Table 1. Furthermore, we observe the rollout process of new load balancer configurations at Meta and Google. Our findings confirm that even when faced with metadata-hiding protocols (*e.g.,* QUIC) analyzing traffic from network telescopes offer expressive views on the protocol ecosystem.

In detail, our contributions are as follows.

(1) We introduce a new method based on passive measurements to learn about hypergiant deployments and QUIC configurations, and present results from 2021-2025 (§ 3).
(2) We reveal local system stack configurations of QUIC clients and servers relevant to improve performance, such as the usage of packet coalescence and retransmission behaviors (§ 4).
(3) We are the first to systematically analyze QUIC connection IDs, their structure and implications in real-world deployments (§ 5).
(4) We make benign use of QUIC attack traffic to detect off-net servers and show how information encoded in Connection IDs can be used to fingerprint hypergiant deployments (§ 5).
(5) We quantify and track the number of layer 7 load balancers of a single hypergiant throughout our measurement period, a previously hidden property (§ 5).
(6) We validate our results with controlled scanning campaigns and passive flow captures (§ 6).

Our method avoids active scanning and relies on unsolicited malicious QUIC traffic captured passively (*e.g.,* responses to requests via source address spoofing). This allows for the observation of attacks and defense strategies (*e.g.,* QUIC Retry packets) without interacting with attackers or infrastructure operators, therefore, our method does not trigger alarms by Intrusion Detection Systems. Our approach captures data from a large number of deployments, which are topologically and geographically distributed. It is difficult to ethically achieve comparable coverage with active scanning. Many operators have a clear opinion on unsolicited traffic, such as "I would still consider an uninvited scan of my network antisocial" [15]. ISPs may even forbid port scanning in their Acceptable Use Policies, such as Xfinity (Comcast): “Unauthorized port scanning is strictly prohibited” [65]. In contrast to flow measurements, our method does not require access to usually protected data. We do not argue that active or passive flow measurements are bad but that it is worth analyzing alternative passive options to understand hypergiant deployments based on QUIC. To validate our results we transfer the method to flow records and perform active measurements when data is statistically sparse. We expect QUIC backscatter to persist similar to TCP backscatter, which has been observable for more than 25 years [27], and even to increase in the future since spoofed traffic remains an ever-increasing challenge on the Internet [8, 26].

The remainder of this paper is structured as follows. We describe the problem space in § 2. § 3 introduces our measurement methods and the resulting data corpus. We report on the configurations of QUIC stacks found in the wild in § 4 and analyze the particular use patterns of QUIC Connections IDs and their implications for off-net servers and load balancers in § 5. We confirm the validity of our method by orthogonal measurements in § 6. We review related work in § 7, discuss our findings in § 8, and conclude with an outlook in § 9.

2 Background

This section recalls background about QUIC and discusses implications of QUIC for common hypergiant deployments.

2.1 QUIC Overview

QUIC implements a reliable, encrypted transport based on UDP. A key improvement is low-latency connection establishment by combining the transport and TLS handshake into a single round-trip. This requires the inclusion of information usually exchanged in separate TCP and TLS handshakes in the first round-trip. Since QUIC is implemented in user-space [44], several implementations with different default configurations are deployed, depending on application needs.

Connection setup. Figure 1 depicts the QUIC 1-RTT handshake. All QUIC connections start with an *Initial* packet sent by a client, which includes the TLS ClientHello [32]. A server replies with an *Initial* (incl. the TLS ServerHello) and a *Handshake* packet (incl. certificate and encrypted extensions), either combined in a single (*packet coalescence*) or split into two separate UDP datagrams, *e.g.,* to improve the initial RTT estimate [47]. The server will resend *Initial* and *Handshake* packets when the retransmission timeout (RTO) expires due to missing acknowledgments and if the maximum number of retransmissions has not been reached.

QUIC hides metadata by encrypting type specific bits, packet number length, and the packet number. Nevertheless, *Initial* and *Handshake* packets characterize the QUIC network stacks with cleartext version information, connection IDs, retry token, and the packet length (see Appendix E).

Connection IDs. Clients and servers establish connection identifiers (CIDs) during the handshake, to assign packets to connections. CIDs enable multiplexing QUIC connections over the same 5-tuple and ensure that connections survive changes in addresses or ports, *e.g.,* due to client migration. QUIC distinguishes between source and destination IDs (SCID, DCID), seen from the sending endpoint.

Since the client cannot guess the CID a server wants to use, it uses a temporary value (S1 in Figure 1), which can be replaced by the server (ID S2). SCIDs and DCIDs are unencrypted. This enables exposure to middleboxes (*e.g.,* load balancers) and enables observing CIDs in passive measurements without knowing TLS secrets.

Similar to TCP SYN cookies [20], where the TCP segment number encodes network and transport layer information to defer state allocation at servers, CIDs can be used to encode information in packets sent by clients. Whenever the client returns with this information, it can be used to admit connection requests or inform loadbalancing decisions [4, 12].

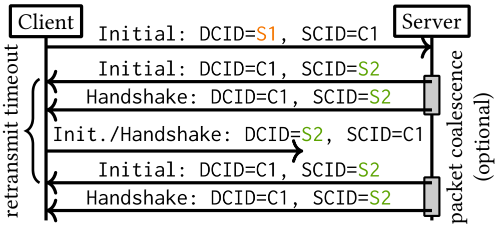

Client
Server
Initial: DCID=S1, SCID=C1
Initial: DCID=C1, SCID=S2
Handshake: DCID=C1, SCID=S2
Init./Handshake: DCID=S2, SCID=C1
Initial: DCID=C1, SCID=S2
Handshake: DCID=C1, SCID=S2
retransmit timeout
packet coalescence (optional)

Fig. 1. Connection establishment using QUIC. Client connection ID (C1) is consistently used during the connection establishment but the initial server ID (S1) can be replaced by the server with ID (S2).

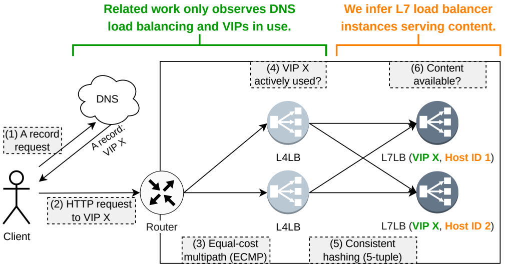

Related work only observes DNS load balancing and VIPs in use.
We infer L7 load balancer instances serving content.
(4) VIP X actively used?
(6) Content available?
DNS
(1) A record request
A record: VIP X
Client
(2) HTTP request to VIP X
Router
L4LB
L4LB
L7LB (VIP X, Host ID 1)
L7LB (VIP X, Host ID 2)
(3) Equal-cost multipath (ECMP)
(5) Consistent hashing (5-tuple)

Fig. 2. Illustration of a Meta /24 load balancer frontend cluster. Our method enables L7LB quantification.

2.2 QUIC in Hypergiant Deployment

Common load balancer deployments. To steer client requests to a close point of presence (PoP), many hypergiants use the DNS (see Figure 2) or anycast [38, 58]. Related virtual IP addresses (VIP) are assigned to multiple logical instances, similiar to Network Address Translation (NAT) and Carrier Grade NAT [51]. Often VIPs belonging to a mid-size network (*e.g.,* /24) form a frontend cluster. The network belongs either to the hypergiant autonomous system (*on-net deployment*) or to a third party provider (*off-net deployment*) [23, 28].

When a client initiates an application handshake with a VIP, the handshake message is forwarded to a layer 4 load balancer (L4LB) based on equal-cost multipath. The L4LB applies consistent hashing of the 5-tuple (*i.e.,* source and destination addresses and ports as well as transport protocol type) to tunnel the packet to a layer 7 load balancer (L7LB) via IP encapsulation. The L7LB completes the handshake with the client and forwards the application layer request. The number of VIPs is a poor indicator for the size of deployments [22], thus we focus on enumerating L7LBs behind VIPs.

QUIC-aware load balancing. QUIC challenges common load balancing deployments for two reasons. First, QUIC allows for client migration, but many load balancers rely on the 5-tuple [21, 59]. Any client that changes the source IP address while maintaining the connection invalidates this 5-tuple, leading to a different mapping of clients to L7LB. Instead, stateless QUIC-aware load balancers [18] forward traffic based on information encoded in connection IDs.

Second, QUIC consumes and retires connection IDs (*e.g.,* during client migration). Since new connection IDs are negotiated in encrypted packets, they must be generated by the termination point of the connection (*i.e.,* the L7LB). Consequently, the transition from an old to a new connection ID is hidden from a QUIC-aware L4LB, preventing it from relaying packets consistently.

To tackle this limitation, L7LBs have two options. Either they share their active CIDs with the L4LBs, which introduces synchronization overhead between load balancers. Or the CIDs issued by the QUIC endpoint encode the destination L7LB. The client then uses those CIDs to contact the server, and the LBs will forward the traffic to the encoded L7LB. The encoded information can reveal information about the server infrastructure but does not conflict with client privacy. In § 5 and § 6, we use this information to explore load balancer infrastructure.

3 Measurement Method and Setup

We analyze five datasets of QUIC Internet background radiation (IBR) from the UCSD IPv4 network telescope, each covers one month of traffic from 2021 to 2025. QUIC flow records from a European National Research and Education Network (NREN) and active measurements allow us to verify our observations. We consider two perspectives, (*i*) QUIC stack configurations and (*ii*) larger infrastructure deployments of distributed QUIC servers.

3.1 Method

Basic idea. We leverage data from a network telescope as passive data source. Network telescopes capture scans (*e.g.,* by QUIC clients) and backscatter traffic (*i.e.,* replies from servers to spoofed addresses of the telescope). We extract information available in QUIC long header packets (*i.e.,* source and destination IP addresses and ports, the QUIC DCID and SCID, packet length(s), packet type(s)), and the reception time at the telescope. Since its early days, QUIC shows significant IBR [48].

We mark all telescope packets with source port `UDP/443` as QUIC backscatter and all packets with destination port `UDP/443` as QUIC requests (*i.e.,* scans). We remove false positives based on the packet payload using the Wireshark QUIC and Google QUIC dissectors, as proposed in prior work [48], and remove data from acknowledged scanning projects [13]. Acknowledged scanners utilize non-existing QUIC version numbers [52] to trigger version negotiation behavior. Removing these scanners prevents bias in our QUIC version analysis. This leaves us with less popular (*i.e.,* undocumented or unknown) and malicious scanners (*e.g.,* bots).

QUIC stack configurations. We identify hypergiant on-net deployments by mapping source IP addresses to autonomous systems (ASes) and related Regional Internet Registry (RIR) data. Grouping multiple QUIC packets into QUIC connections, allows inference of QUIC stack configurations, *i.e.,* retransmissions, enabled QUIC features, attack mitigations and characteristic packet features.

Detection of off-net server deployments. Zirngibl *et al.* [67] revealed a fairly homogeneous infrastructure deployment of CDNs across networks. We use this property for identifying patterns of QUIC traffic of on-net deployments and compare with traffic coming from non-hypergiant ASes. We group similar observations into sets of off-net candidates. QUIC traffic features change within our observation period. We indicate this and adapt our off-net detection algorithms. For verification, we compare subject alternative names in TLS certificates of on-net deployments with those of off-net deployments. For this purpose we establish QUIC connections with the off-net candidates.

Identifying load balancers. We identify specific load balancer instances by using information encoded in the QUIC connection ID. Hypergiants structure this ID to include a unique host ID, which represents the actual L7LB. We treat the connection ID structure of Meta as ground truth because it is exposed in the Meta QUIC implementation [31].

Verification using passive flow data. We compare our observations in IBR with QUIC flow records, randomly sampled (sampling rate 1:99) from inter domain links of the European NREN Czech Education and Scientific NETwork (CESNET). The flow records are bidirectional (*i.e.,* client requests and server responses), while IBR is always unidirectional.

We remove incomplete and invalid flows (*i.e.,* no TLS ClientHello parsed or missing QUIC transport parameters) exported by ipfixprobe [9]. To protect user privacy, IP addresses are exempt from this dataset. Nevertheless, the dataset contains the AS number, and a /24 prefix identifier of the contacted QUIC server. QUIC allows clients to migrate to a new address, but mandates new connection IDs in this case. Neither a flow exporter nor post-processing can correctly link multiple 5-tuples to the same connection since new CIDs are exchanged in encrypted packets. Thus, a QUIC flow represents a lower bound in terms of transferred data sizes and packets.

Each flow contains the aggregated information extracted from long header packets. Additionally, for the first 30 packets, QUIC packet type(s), UDP packet length, inter-arrival time, and the packet direction (*i.e.,* to server or to client) is collected.

Verification using active measurements. When verifying data, we use data from third-party scans [67] and self-conducted active probings. The third-party measurement campaign combines

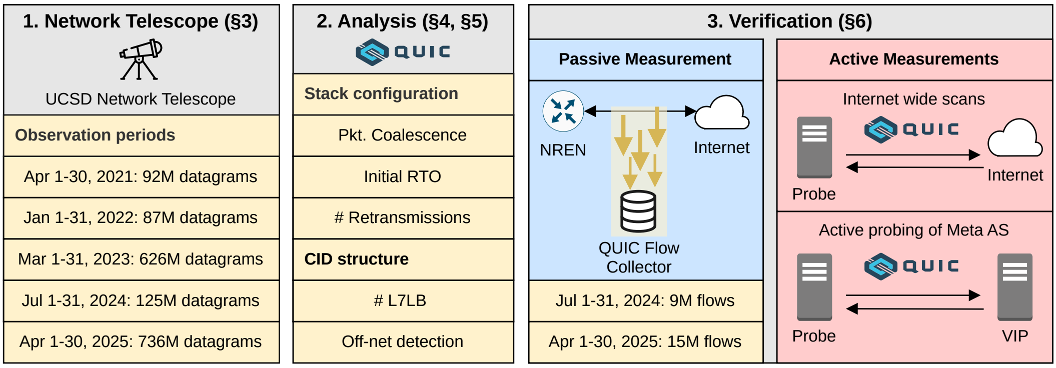


Fig. 3. Overview of our measurement setup and data corpus.

stateful and stateless scans of the complete IPv4 space to reveal QUIC server deployments, the supported QUIC versions, and CDN off-net deployments. We utilize this data to identify off-net deployments where passive data is sparse. The method of Zirngibl *et al.* [67] of analyzing off-net deployments using QUIC transport parameters and HTTP headers remains valid to this date. However, we find that the values of the transport parameters change within our observation period and adapt the detection accordingly. Since 2023, we verify whether the QUIC implementations match between on- and off-net deployments of hypergiants [68]. It is noteworthy that this data quantifies VIP addresses, not L7LB instances.

We conduct active measurements to enumerate the number of layer 7 load balancers at Meta. We connect multiple times to known QUIC servers (represented by a VIP, detected by our telescope or Zirngibl *et al.* [67]) while varying the source port of our outgoing packets. On established connections, we log the connection IDs (representing L7LBs). To reduce potential conflicts of active measurements, we limit the amount of active scanning, which might lead to time delays between the (passive) observation period and active scanning. We perform probing from a single origin within a university network. We carefully monitor our probe for any signs of blocklisting.

3.2 Data Corpus

We utilize the UCSD Network Telescope [7], an IPv4 darknet operated by CAIDA within a /9 and /10 prefix, to track QUIC IBR from 2021 to 2025. The telescope size varies by up to 9% in this period (*cf.*, Appendix C). Storage constraints limit analysis to one month of backscatter per year.

Figure 3 provides an overview of our measurement setup, observations, and verification. Due to agreements with the data providers we cannot publish telescope data. To facilitate further research, we publish the QUIC flows sampled from CESNET and our analysis.

QUIC backscatter traffic increased from 1M packets in 2021 to 10.8M packets in 2023 (10×) and remains stable in 2024 and 2025 with 9.1M and 10.6M packets. This aligns with the relative development of traffic received by the network telescope.

Table 2 shows the number of IP addresses and L7LBs in each monthly dataset. In March 2023, at the largest expanse of our backscatter dataset, we observe traffic from 350 Meta, 2,769 Google, and 359 Cloudflare on-net server VIP addresses. This covers 7.8% (Meta), 1.3% (Google), and 0.2% (Cloudflare) of all VIPs that allow for QUIC connections at that time (*cf.*, Zirngibl *et al.* [67]). In subsequent years, the number of VIPs from Cloudflare and Google in backscatter is lower than in 2023, while the set of QUIC capable VIPs increases within that time [67]. For Meta, both the number of QUIC capable VIPs and the VIPs in backscatter increase. In 2025, 8.7% (637) of their QUIC capable VIPs are present in backscatter.

Table 2. Number of IP addresses, and L7LBs contained in backscatter. We observe a strong increase in the number of VIPs, and contained L7LB-IDs. The number of L7LBs is determined by the number of unique host IDs per cluster in each /24 subnet.

	Most Backscatter Observations				Subsequently subsumed as *Others*					
	VIPs from Source Network [#]			L7LBs [#]	VIPs from Source Network [#]					
Year	Cloudflare	Google	Meta	Meta	Akamai	Amazon	Apple	Fastly	Microsoft	Others
2021	33	1,790	167	4,273	-	1	-	-	-	604
2022	78	1,655	246	7,145	11	2	2	-	14	677
2023	359	2,769	350	12,048	258	115	33	19	51	1,623
2024	151	1,681	514	20,744	431	40	335	20	41	1,112
2025	250	2,042	637	22,527	396	124	331	51	61	1,290

Traffic from other hypergiants is present, but the number VIPs and QUIC connections (see Appendix F) in backscatter is low, *e.g.,* Amazon traffic consists of a single QUIC connection in 2021. Cloudflare, Google, and Meta contribute significantly more traffic than Apple, the next largest source, with 35,620 QUIC connections (5.9 %) in 2025. Subsequently, we subsume traffic from Akamai, Amazon, Apple, Fastly, Microsoft, and all other ASes as *Others*. The share of those providers is too small to consider them separately in aggregated statistics.

The sanitized flow records contain 449 Meta, 7,217 Google, and 5,425 Cloudflare on-net server VIP addresses in 2024, and increase to 698, 7,478, and 8,093 on-net VIPs in 2025. Except for VIPs observed from Meta in 2024, the flow records contain more VIPs from Cloudflare, Google, and Meta than the backscatter. In 2025, we observe 378k flows from Cloudflare, 9.4M flows from Google, and 2.5M flows from Meta. This is between 4× and 58× of the QUIC connections in backscatter. We conclude, that even geographically limited vantage points can provide significant insight into hypergiant deployments. Appendix F includes statistics of other large content providers.

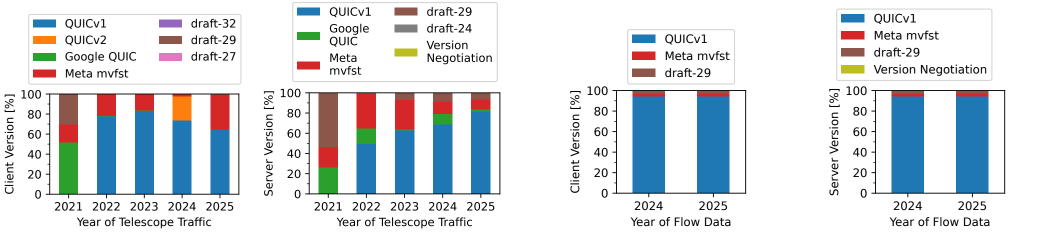
3.3 Limitations

Backscatter traffic depends on attacker and target behavior. Backscatter traffic is response traffic to packets with spoofed source addresses. Consequently, observations of network telescopes are limited (*i*) by the choice of targets of spoofers, (*ii*) by the mitigation efforts of the targets, and (*iii*) by the choice of spoofed addresses and network telescope address space, *i.e.,* direct path attacks are never recorded by network telescopes. Due to missing ground-truth, we cannot assess to which extent this impacts telescope data. However, we observe indicators of the above limitations: (*i*) the majority of QUIC traffic is received by only a few subnets [43], and (*ii*) the amount of targeted IP addresses fluctuates even at similar overall reception of QUIC traffic by the telescope. The telescope observes significantly less VIPs compared to the active measurements [67]. Despite backscatter originating from geographically distributed servers, the coverage in a region is not complete.

We are unable to dissect how mitigation efforts affect our dataset. Hypergiants present in one year of our dataset are contained in subsequent years. This only reveals that mitigation efforts, if present, do not suppress all backscatter traffic. Nevertheless, the amount of traffic and VIP addresses from *e.g.,* Akamai and Cloudflare seems peculiarly low. We can only speculate on attack mitigations or knowledge of telescope address space of those providers.

We are not aware of attackers polluting network telescope datasets by crafting packets resembling backscatter. However, telescopes would collect such forged packets.

Flow records reveal global deployment configurations, and are limited to the most recent years. Geolocating servers in flow records, we find that 87% (2024) to 71% (2025) of flow records communicate with servers in the United States of America. Flows to servers in Europe only account



(a) Clients in Telescope (b) Servers in Telescope (c) Clients in Flow Data (d) Servers in Flow Data

Fig. 4. QUIC versions of clients and servers in one month of telescope traffic in 2021 to 2025, and flow records off the same months in 2024 and 2025. Colors included in the legend but invisible in the bars contribute <0.1 % of the traffic. After standardization of QUIC in May 2021, QUICv1 is rapidly adopted in 2022.

for 11% (2024) to 26% (2025). Due to homogeneity of hypergiant QUIC deployments [67], this does not limit the validity of using the flow records for verification purposes.

While the longitudinal telescope data covers 5 years, our validation data from CESNET covers only the last two years. Because QUIC is relatively new protocol, the flow exporter used at CESNET supports detailed QUIC flow monitoring since April 2024.

QUIC exposes additional information, but analysis is limited to the handshake. By condensing the transport and TLS handshake in a single round trip and enabling features through additional headers, QUIC exposes more information to the telescope. Different from TCP, the first client flight includes the TLS ClientHello, which can be analyzed at the telescope. In response to 0-RTT packets from clients, QUIC servers can even send data with the first server flight. However, encryption limits observations to information from *Initial* packets and cleartext packet headers (see Appendix E).

4 QUIC Stack Configurations

The recent development of QUIC spread different QUIC versions, implementations, and complex protocol mechanics allow for a wide range of configurations. Together, this can serve as a fingerprint of the current content provider infrastructure. In this section, we analyze outstanding characteristics observable from our backscatter measurements.

Within our measurement period, we observe both consistent and changing configurations. While the latter aligns with the rapid change of QUIC deployments, variations within one measurement period might hint towards heterogeneous configurations, limited amount of data, or variation in attacker behavior. We indicate such findings but cannot dissect possible biases.

4.1 QUIC Versions

Whenever the version offered by a client in the *Initial* is incompatible with the server, the server reacts with a *Version Negotiation* packet indicating all supported versions. The client then starts a new QUIC connection with a supported version. If the version offered by a client is compatible with the server, the server accepts the client-chosen version, or performs compatible version negotiation by transforming the *Initial* to a mutually supported version. To allow for compatible version negotiation clients indicate supported versions in a QUIC transport parameter [32, 54].

IBR reveals that client (*i.e.,* scans) and server implementations (backscatter) adopt QUICv1 after standardization in May 2021. QUIC IETF Draft-29 and custom versions of Google and Meta are present to this date, but clients and servers strongly gravitate towards QUICv1 (see Figure 4). In contrast to active scans [50, 52, 67] with non-existing QUIC versions, backscatter traffic reveals the version that client *and* server agreed upon, *i.e.,* the version that is used in a connection and not only offered. To this extend, we learn about the versions used by attackers. QUICv2, standardized in May

Table 3. QUIC packet types visible in backscatter traffic from 2021-2025. Columns sum to 100 %.

	Relative number of packets from source network per year [%]																			
	Cloudflare					Google					Meta					Others				
QUIC Packet Type	'21	'22	'23	'24	'25	'21	'22	'23	'24	'25	'21	'22	'23	'24	'25	'21	'22	'23	'24	'25
Initial	42	56	54	49	45	34	23	7	9	35	65	48	47	43	47	69	46	33	36	43
Handshake	28	41	43	42	44	21	24	26	34	33	35	52	53	57	53	29	43	41	40	41
0-RTT	-	-	-	-	-	2	<1	<1	-	-	-	-	-	-	-	1	<1	<1	<1	-
Retry	-	-	-	-	3	-	-	-	-	-	-	-	-	-	-	<1	<1	<1	<1	<1
Version Negotiation	-	-	-	<1	-	-	-	-	-	-	-	-	-	-	-	-	3	<1	3	1
Colaesced Packets																				
Initial+Initial	10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	<1	-	<1	-	-
Initial+Handshake	10	3	3	8	9	44	53	67	57	32	-	-	-	-	-	1	9	26	20	14
Handshake+Handshake	10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	<1	-	-	-	-

2023, and discussed since Nov. 2021 [46], is used only in 2024 in a scanning campaign originating from a Chinese NREN. To this date, QUICv2 is not present in backscatter traffic.

In 2025, we observe a significant increase of Client packets using of Meta mvfst versions (34.6 %). Those requests target a single IP address with repeating octets in the telescope. The requests originate from ASes in Asia, with Malaysian and Indonesian ASes contributing the largest share. We are unable to dissect the root cause for those packets.

We observe version negotiation packets from up to 19 VIPs with Google QUIC versions but outside any Google AS and a single VIP in the Cloudflare AS in 2024. Since version negotiation packets do not allocate state at servers, it is plausible that attackers rely on compatible versions.

4.2 Coalescing Packets

QUIC allows to coalesce multiple QUIC packets into one UDP datagram. All hypergiants except Meta use this feature (see Table 3). For Microsoft, we observe it since 2024 (not shown). Cloudflare shows a significantly lower share of coalesced packets than Google and Fastly. This originates from their deployment model. The load balancer responds with a coalesced packet if it is in hold of the certificate, otherwise the acknowledgment of the client *Initial* is send in a separate packet, thereby trading precise RTT-estimates for overhead [47, 49]. We observe similarly low shares of coalesced packets from Akamai, Apple, and Microsoft (not shown).

4.3 Packet Lengths

Hypergiants use distinct packet length formations. Figure 5 shows the five most frequently received QUIC packet lengths in each year and combinations due to packet coalescence. Major providers (indicated by colors) show distinct distributions of packet length formations, which indicates this as a characterizing feature, *i.e.,* 90.7% of packets from Meta in 2025 are 1232 B long. Cloudflare, Google and Meta use identical packet lengths in multiple years but with varying persistence *i.e.,* Google uses 1250 B QUIC packets since 2022, Meta uses 1232 B since 2021, and Cloudflare uses the same packet lengths only in 2024 and 2025. We find that the Meta QUIC implementation by default uses 1232 B for sending QUIC packets [30]. In 2024 and 2025, Cloudflare QUIC packet lengths are much smaller compared to other hypergiants. This may originate from instant ACKs [47]. Cloudflare pads those packets on the UDP level and not with QUIC padding frames.

Changes of the most frequently used packet length(s) during our measurement period, have been linked to certificate replacement [40], may point to changing deployments or different libraries used by clients interacting with specific services (*e.g.,* YouTube vs Instagram application).

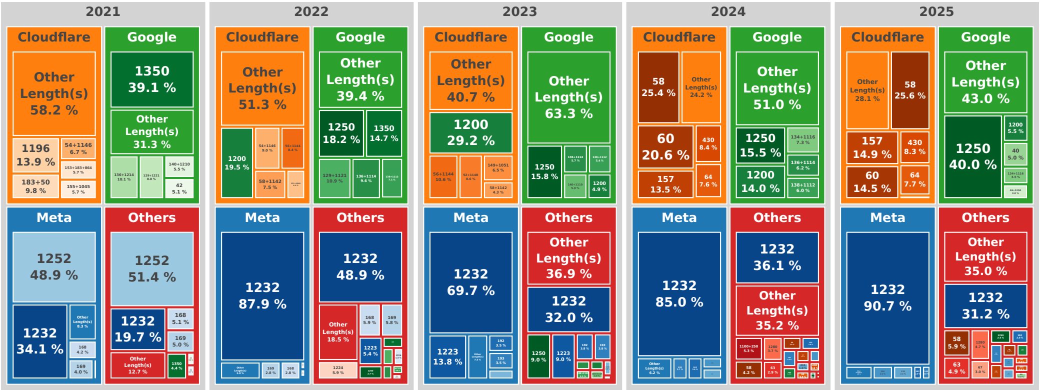


Fig. 5. QUIC packet lengths [B] per hypergiant in 2021-2025. The size of each rectangle and second line represent the proportion of the packet length indicated in the first line. Colors of hypergiant packet lengths in non-hypergiant ASes (*Others*) point to off-net deployments.

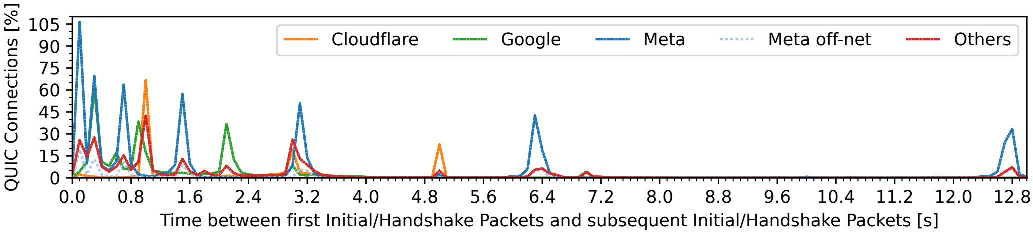


Fig. 6. Retransmission configurations of QUIC servers, visible in backscatter when a server replies to a spoofed source IP address of the telescope. The retransmission configurations of Google and Meta servers are consistent throughout our observation period (shown here for 2025). All hypergiants use exponential backoff.

Although the share of length(s) of *Others* increased since 2023, large shares of traffic from *Others* use the same packet length as Google and Meta. The increase originates from the number of VIPs, and QUIC connection of other hypergiants are significantly increasing (see Appendix F). A significant share of those packet length(s) are contributed by Apple, Fastly, and Akamai in 2025.

4.4 Retransmission Behavior

A server resends both *Initial* and *Handshake* packets if no acknowledgment is received within the retransmission timeout (RTO). Unlike TCP, QUIC does not reuse packet numbers (sequence numbers in TCP), which challenges detection of retransmissions in backscatter. We detect a retransmission if an *Initial* is received after a *Handshake*. Figure 6 depicts the time gaps between the first received packet and subsequently detected retransmissions within the same connection—resent by QUIC servers in reply to spoofed traffic. Peaks indicate common configurations of when and how frequently these messages were re-sent. At 0.1 ms, we observe more retransmissions than QUIC connections from Meta. We find that these connections, do two resends at ~0.1 ms.

The first retransmission happens after 0.1 s in 94% (Meta), 0.3 s in 51% (Google), and 1.0 s in 67% (Cloudflare) of QUIC connections in 2025. Except for Cloudflare (0.3 s in 31% of the connection in 2022) this observation is stable throughout our entire measurement period, but the share of the most frequent retransmission varies. Possible reasons for varying configurations per hypergiant range from more heterogeneous sets of services (Google), the low amount of data we obtained in a given year (Cloudflare), or variations in attacker behavior. We observe exponential backoff from Cloudflare, Meta, and Google servers. The maximum number of resends differs between hypergiants (see Figure 7), showing that servers require different amounts of resources to keep connection states. Overall, we detect the shortest resend timeouts and most retransmissions from Meta. This indicates that Meta reacts faster to packet loss and expects shorter delays between clients and servers than Google and Cloudflare. In comparison, Google and Cloudflare reserve less resources to cope with faulty connections. This leads to a reduced vulnerability by QUIC flood attacks that build-up state.

The high number of retransmissions from Meta surprises, since previous work reports amplification for Meta reduced in October 2022 [49]. To validate the backscatter observation we send the same *Initial* of size 1252 B seven times. This elicits ten retransmissions at intervals matching the telescope observations. Sending more *Initials* does not increase the number of retransmissions, while sending fewer reduces the retransmissions. We conclude that timing of server retransmissions is not steered by clients. Nevertheless, clients add to the byte-budget of the server with each resend.

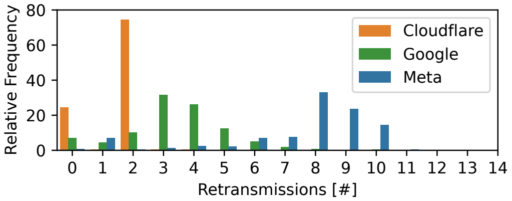


Fig. 7. Number of retransmissions in 2025. We observe significantly more retransmission from Meta.

4.5 Denial of Service Mitigation with Retry Packets

DoS mitigation with *Retry* packets is rarely used. QUIC *Retry* packets enable QUIC servers to verify the client address by mandating the client to reconnect with a Retry Token given by the server. Those *Retries* can be effective against QUIC floods but add an RTT [48]. We observe this defense strategy rarely deployed (see Table 3). For example, Cloudflare started in 2025, where 3% of QUIC packets from Cloudflare are *Retries*). This indicates that deployments favor low latency connection setups over DoS mitigation.

5 What can Connection IDs tell?

To reduce tracking vectors, QUIC connection IDs must not contain information that allows correlation of multiple CIDs to one QUIC connection [32]. When deliberately chosen, their unencrypted transfer allows loadbalancers to forward packets to the same L7LB even across changing 5-tuples. In this section, we analyze the CIDs chosen by clients and servers, use their structure to detect off-net deployments, and infer the number of L7LBs behind VIPs.

5.1 Structure of Client CIDs

Client CIDs are frequently zero-length or set at random. QUIC servers use the SCID set by the client in the first flight as the DCID in subsequent packets. Until 2023, 99% of the DCIDs in long header packets in backscatter were zero length. In this case clients must not use the same source port and address for multiple QUIC connections. The proportion of zero length DCIDs reduced to 71% in 2024 and in 88% in 2025. This originates from Cloudflare, Apple, and Akamai backscatter, which send an 8 byte DCID in 95-100% of QUIC long header packets. We find that client CIDs follow the random distribution - not revealing any information.

5.2 Structure of Server CIDs

To ensure high entropy despite information encoding and to prevent collisions between multiple connections, we found server CIDs between 8 and 20 bytes are preferred by hypergiants over shorter or zero-length CIDs. SCIDs (*cf.,* S2 in Figure 1) can leak data if hypergiants encode information in them. Such encoding distorts the uniform distribution of specific values in the SCID. Figure 8

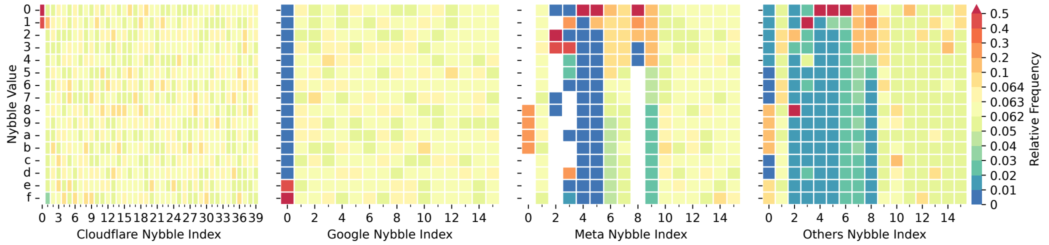


Fig. 8. Relative frequencies of SCIDs values in backscatter from 2024. A non-uniform distribution per column indicates information encoding (*e.g.,* Google). *Others* contains patterns similar to those of Meta and Google.

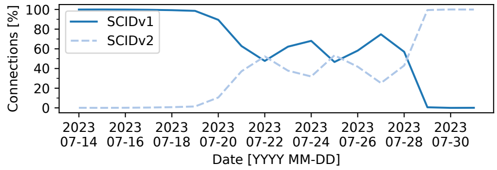


Fig. 9. Encoded SCID version in Meta QUIC connections in backscatter. Between July, 19 and July, 29 2023 Meta migrates to SCIDv2.

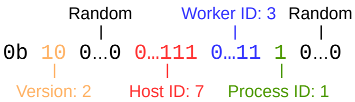


Fig. 10. Meta encodes version, host, worker and process IDs into the CID. See Appendix D for details on all versions of CID encoding.

visualizes the frequencies of SCID nybble values as monitored in the backscatter traffic. Frequencies that diverge from the random distribution, *i.e.,* expected value $\frac{1}{16} = 0.0625$ (light-yellow and -green) show SCIDs encoding specific information.

We observe that Akamai, Amazon, Apple, Google, Fastly, Meta, Microsoft and Cloudflare repeatedly use reoccurring values at specific positions within the SCID, indicating information encoding. From Akamai, Amazon, Apple, and Cloudflare we observe 20 B SCIDs, Fastly uses 17 B SCIDs, Google and Meta use 8 B SCIDs, and Microsoft uses 14 B and 20 B SCIDs.

Cloudflare SCIDs. Since the beginning of our measurements Cloudflare uses 20 B connection IDs and a fixed first byte `0x01`. Our large-scale active measurement data set confirms their preference for this fixed first byte and, in a prior post [35], Cloudflare acknowledged the benefits of encoding information in QUIC IDs. The encoded information is not described in their open source QUIC implementation *quiche* [10] or by the IETF draft for Generating Routable QUIC Connection IDs [18], though, as the first byte would indicate a connection ID length of 1 or include random bits.

Google SCIDs. In 2021 and 2022, the SCIDs from Google follow a random distribution. In 2023 and 2024, the loadbalancer configuration changes. We observe SCIDs starting with `0b11` in 99% (2023) and `0b111` in 99.9% (2024) of long header packets. By manually connecting to Google servers, we find that Google echoes the first 8 bytes of the DCID provided by clients and overrides the first bits since adopting information encoding. This means that our backscatter exposes what clients send to Google, and clients should send random connection IDs (§ 2). Echoing large parts of the CID weakens *Initial* protection, but does not impact the setup of the encrypted connection.

The Google QUIC implementation QUICHE supports the IETF Draft for Generating Routable QUIC Connection IDs [18, 24]. `0b11` and `0b111` indicate loadbalancers to forward according to the 4-tuple instead of information from the CID. IETF Draft-18 [17] changed from two bit encoding to three bit encoding. Surprisingly, Google closely tracks the development of the draft, while we do not detect indications for CID based loadbalancing.

Meta SCIDs. The Meta QUIC implementation *mvfst* [31] allows for encoding details about hosts, workers, processes, and the version of this encoding within the SCID (see Figure 10). Given higher densities for some values in the first five bytes we conclude that Meta currently encodes information.

Closely tracking the reception of QUIC backscatter from Meta allows us to observe the migration to a new loadbalancer configuration, from SCID version 1 to version 2, in July 2023 (see Figure 9). The migration took 10 days, and follow-up active measurements confirm the changes.

Before this migration, host IDs denominated individual L7LBs. Thereafter, Meta uses the same host IDs in different clusters. We detail our method of cluster inference in § 6.2. According to it, the number of L7LBs in backscatter increases from 4,273 in 2021 to 22,527 in 2025. Compared to active measurements, backscatter observes the largest proportion of Meta host IDs in 2023 with 29%.

5.3 Passive Detection of Off-net Servers

All hypergiants use information encoding in connection IDs. Entirely passive, we are able to determine that all hypergiants observed in backscatter use information encoding in CIDs. We now use the aforementioned distinct patterns to detect off-net deployments. Methods from prior work, using subject alternative names from certificates [23] and transport parameters [67] cannot be transferred to backscatter, since this information is exchanged in encrypted *Handshake* packets.

Evaluation method. We analyze all QUIC servers emitting backscatter that are deployed in non-hypergiant ASes. To evaluate our approach, we need to associate off-net IP addresses with hypergiant services. We actively connect to candidates and inspect X.509 certificates. We label IP addresses as off-net deployment if the subject alternative names include any of the domains *facebook.com, instagram.com, fbcdn.net, whatsapp.com, whatsapp.net* for Meta and *google.com, googlevideo.com, doubleclick.net, edgestatic.com, gstatic.com, blogger.com, googleapis.com, gvt*[*0-9*]*.com* for Google. We only present data since 2022, since we collect certificate data since then.

We classify an IP address (*i.e.,* off-net candidate) operated by a given hypergiant if all SCIDs issued by that IP address conform to the passively detected pattern, *i.e.,* `0x01` for Cloudflare, `0b01` for *Google SCIDv1*, `0b111` for *Google SCIDv2*, `0b01` for *Meta SCIDv1*, and `0b10` for *Meta SCIDv2*.

Evaluation results. Subsequently, we show results from 2023 because backscatter contains the most off-net candidates in this year. For Google, this predicts 556 candidates. Conservatively, we mark 100 of those as false positives as they do not allow for QUIC connections, *i.e.,* 456 are true positives. 1516 are predicted negative—only 7 are false negatives (TPR 0.98, FPR 0.06). For Meta, we identify 727 candidates. 651 of those indeed belong to Meta. 22 do not allow QUIC connections, we consider them as false positives and 54 are falsely predicted as Meta off-net deployments. No false negatives are predicted (TPR 1.0, FPR 0.05). We find six off-net candidates for Cloudflare, but none allow for QUIC connections, preventing us from fetching certificates to determine ground truth.

Table 4. F_1-score of SCID classifiers. Low F_1-scores for Google in 2022 originate from not using information encoding. Meta off-net classifiers consider more bits than Google classifiers and achieve better scores.

	F_1-score			
Classifier	2022	2023	2024	2025
Meta Off-net SCIDv1	0.98	0.98	-	-
Meta Off-net SCIDv2	-	-	0.98	0.99
Google SCIDv1	0.17	0.89	0.79	0.77
Google SCIDv2	0.12	0.38	0.8	0.78

Improving SCID-based detection. We find that Meta off-net servers use low numbers for host IDs in 2023. Active probing of 45k Meta off-net VIPs found by Zirngibl *et al.* [67] confirms this. Consequently, requiring the first 9 bits of the host ID to be zero confirms the 651 off-net deployments and reduces the false positive rate (TPR 1.0, FPR 0.02).

Besides SCID structure, we perform off-net detection based on retransmission intervals, packet lengths, and coalescence, as well as combinations of them. We detail those in Appendix G.

SCID classifiers are effective in all years with structured connection IDs. The applicability of our method is not limited to a single year of observations. Hence, we show the F_1-score to evaluate predictive performance in Figure 4. The F_1-score combines precision and recall into a single value. The maximum attainable value is 1, F_1-scores larger than 0.9 are generally considered very good. We achieve very good performance using the Meta off-net classifiers, while performance of the Google SCID classifiers is worse due to using fewer bits. VIPs with few connections are misclassified as Google if their SCID matches the Google SCID structure by chance. This increases the number of false positives and negatively impacts the precision. Notably, very low F1 scores for Google in 2022 originate from not using information encoding in this year.

6 Verification

In this section we (*i*) assess the correctness of observations in backscatter with QUIC flow records from CESNET, (*ii*) enhance the information extracted from backscatter to assess completeness of backscatter observations and gain detailed insight into L7LB deployments at Meta.

6.1 Verification with Flow Records

QUIC versions. Each flow stores the version indicated by the client in the Client *Initial*, and the version the client and server agreed on. The majority of clients and servers use QUICv1, while a small share of the connections is performed with a custom version of Meta. A QUIC draft version and version negotiations messages are also present, but without a significant share (see Figure 4). Flow records confirm backscatter observations.

Coalescing packets. Flows store flags for QUIC packet types in the first 30 UDP datagrams. This reveals coalescence of different packet types, but coalescence of the same packet type (*e.g.,* two Initial packets) remains hidden. At present, coalescence of different packet types is the most common (*cf.,* Table 3).

Figure 11 shows 10% (Cloudflare) and 18-10% (Google) of datagrams from servers contain coalesced packets. This amounts to less coalesced packets in flow records compared to backscatter for Google (*i.e.,* up to 39% less) and similar proportions for Cloudflare. Lower observation of packet coalescence is expected for flow records because reactive clients do not necessitate the server to resend information from the QUIC handshake, which is composed of Initial and Handshake packets.

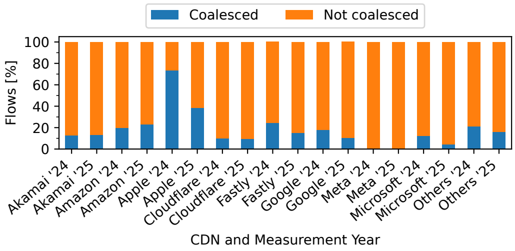


Fig. 11. Share of packet coalescence in server packets. Meta servers do not use packet coalescence, while packet coalescence is used by all other CDNs.

Packet lengths. Flows collect the UDP payload length for the first 30 packets. The most common size of server packets is ≥ 1200 B from Google and Meta. The sizes match with the sizes observed in backscatter, although other packet length are also present *e.g.,* 1357 B. For Cloudflare the flow record and telescope observations deviate. The most common packet length from Cloudflare servers is 1200 B, followed by small packets (<52B). Small packets are similar but not identical to packet sizes in backscatter. 1200 B is present in backscatter, but at a very low share.

Retransmission behavior. Backscatter contains server reactions to unresponsive clients. To track such behavior in flow records, we select flows that consist of a single packet sent by a client and at least three packets sent by the server. Those records occur when clients abandon connections after sending a request, *e.g.,* users close a web browser before the QUIC connection is established. Retransmission in flow records match the retransmissions observed in backscatter (not shown).

Client CIDs. Zero-length connection IDs are the most common CID lengths (75% flow records), which similarly appears in backscatter traffic of those years. 16 B and 3 B connection IDs are the next most common CID lengths. Testing browser connections to QUIC servers, we find that Chrome and Safari use 0 B SCIDs while Firefox uses 3 B SCIDs.

Server CIDs. The flows reveal the same information encoding as backscatter (*cf.,* Figure 8), *i.e.,* confirms structured CIDs of Akamai, Amazon, Apple, Cloudflare, Google, Meta and Microsoft servers. However, we find significant shares of other CID length in flows from Akamai, Amazon, and Microsoft. Only one connection ID length from Amazon shows no signs of information encoding.

Meta CIDs. Counting unique host IDs per /24 prefix, we detect 10,038 L7LBs in 2024 and 25,534 in 2025—between 3% and 7% of the total number or L7LBs in active probing of Meta on-net servers (see § 6.2). The SNI in each flow record, allows us to use the same domains to find Meta off-net servers as described in § 5.3. This confirms 1,003 off-net VIPs. Even our geographically limited vantage point reveals a significant number of L7LBs and off-net deployments.

Except for packet length(s) observations, flow records confirm insights from backscatter.

6.2 Verification of Convergence: Active Measurements to Improve Statistics

Using active scans towards hypergiants between 2022 and 2025, we now assess the robustness of backscatter observations and extend our understanding of hypergiant QUIC deployments. While we want to infer how much backscatter is needed to gain statistical convergence, we concentrate on Meta since only Meta confirms their encoding of L7LB host IDs in their SCIDs (ground truth). To this end, we scan all QUIC VIPs active at the time (up to 7,355 VIPs in 2025) in the Meta AS 32934. For each VIP, we complete 20k handshakes while successively decreasing the client port.

Meta clusters 2022-2023. We group VIPs into clusters, when the same host ID is used by multiple VIPs in the same /24 prefix. We confirm the derived structure of clusters using reverse DNS, the IATA airport code encoded in DNS PTR records of all cluster VIPs is identical, *i.e.,* clusters are limited to a single /24. Using this method reveals a set of *large clusters* (≥ 20 VIPs) and *small clusters* (≤ 10 VIPs). In 2022, we find 114 clusters composed of 22 VIPs and 1 cluster composed of 20 and 21 VIPs in 2022 (total: 116). In 2023, the number of *large clusters* and VIPs within each cluster increases. We now find 120 clusters with 27 VIPs and 1 cluster with 22 and 28 VIPs in 2023 (total: 122). Within a single cluster we observe up to 461 different host IDs, *i.e.,* L7LBs. Domains hosted by the *large clusters* cover the full range of Meta services.

Similar to large clusters, the number of *small* clusters and size of clusters increases from 43 clusters with 8 VIPs, 2 clusters with 7 VIPs and 1 clusters with 6 VIPs in 2022 (total: 46) to 55 clusters with 10 VIPs, 5 clusters with 9 VIPs, and 19 clusters with 1 VIP in 2023 (total: 79). The clusters with 1 VIP have up to 225 L7LBs, while all other *small clusters* are composed of up to 16 host IDs. When collecting certificates from the VIPs, we found all of the cluster with more than 1 VIP serve *fbcdn.net* except for a single VIP in each cluster to serve *whatsapp.net.*

After rollout of a new loadbalancer configuration host IDs are repeated in multiple clusters. In 2022 and beginning 2023, host IDs were often globally unique. After the rollout of *Meta SCIDv2* in July 2023, we observe same host IDs now in different clusters. We confirm this by active measurements in the beginning and end of July 2023. In early July, 3% of the VIPs used *Meta SCIDv2* and 41,016 host IDs per /24 prefix (31,733 unique host IDs) are detected. On 31 July, 95% of all Meta on-net VIPs use *Meta SCIDv2* and the number of L7LBs is 41,277 (4,193 unique host IDs). The migration did not change the number of L7LBs, but host IDs are now repeated.

Meta clusters 2024-2025. In 2024, we see significantly higher numbers of smaller clusters, *i.e.,* 1151 (2024), and 1532 (2025). No clusters with more than 10 VIPs exists. We observe 65 clusters with 10 VIPs in 2024 and 82 in 2025; 120 (2024) and 116 (2025) with 9 VIPs. Furthermore, the share of clusters with 2-8 VIPs contributes 428 clusters in 2024 and 788 in 2025. There also are clusters

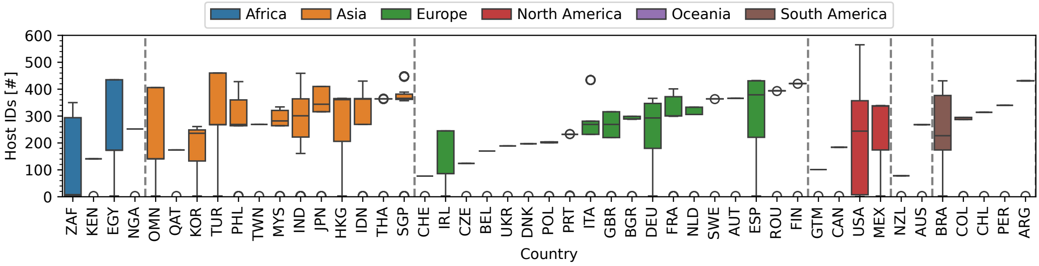


Fig. 12. Meta L7LBs aggregated by country in 2025. PoPs in the Asian region utilize more L7LBs (# host IDs).

with a single VIP (538 in 2024 and 546 in 2025). The largest cluster is composed of 466 host IDs in 2024 and 1360 in 2025. Collected certificates do not provide distinct sets of domains, which would explain the new structure. PTR records reveal, clusters with 1 VIP tend to point to WhatsApp, while 9-10 VIPs point to the whole range of Meta services. We also see significantly more PTR records containing *edge*, which may indicate an increased number of edge locations.

Meta on-net clusters are large in Asia. Figure 12 depicts the number of L7LBs per cluster in different countries. Until migration to *Meta SCIDv2*, *small clusters* were only located in North America (46) and Africa (15). We find that the median number of L7LBs in *large* clusters is significantly higher in Asia than on any other continent (414 *vs.* 344.5 in Europe and 353.5 in North America in 2023). In 2024, the clusters change significantly. In 2025, 580 clusters are located in Asia—significantly more than North America (395) and Europe (347). The median number of L7LBs per cluster in Asia (317) continues to be larger than in Europe (269), North America (184) and South America (295).

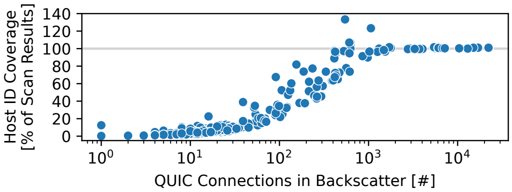


Fig. 13. Number of host IDs in Meta clusters visible in backscatter relative to hosts IDs in active measurements. With increasing number of QUIC connections a larger percentage of L7LBs is discovered. For 21 clusters, we observe more host IDs in backscatter than during our active measurements, *i.e.,* >100%.

We conjecture three reasons for this: (*i*) the number of available peering points in Asia is limited, (*ii*) political instabilities in specific regions and regulations limit additional data centers, and (*iii*) high population and hence user densities per region. All three reasons may guide Meta to not increase the number of PoPs but to provide more L7LB instances per PoP.

95% of host IDs are discovered within 1,000 connections. On average, we detect ~95% of all host IDs after 1k handshakes per VIP. Also, 99% of the VIPs aggregate the same number of host IDs (deviations ≤ 5) within each cluster. We conclude that it suffices to observe backscatter or probe one VIP to determine the number of all L7LBs of one cluster.

Coverage of backscatter. We compare the number of host IDs in backscatter with our active measurements. Figure 13 depicts that more QUIC connections lead to larger coverage of the number of host IDs. While in 2023, 2,366 QUIC connections allowed detection of 93% of all host IDs in a cluster, we achieve 100% coverage with only 545 QUIC connection in 2025 following the new structure. This is related to the on median lower number of L7LBs per cluster. For 21 clusters, we observe more host IDs in backscatter than in our active measurements. This may be due to cluster reconfiguration during the observation period or delays between the passive observation and active probings, in which instances may be replaced or clusters extended.

7 Related Work

QUIC Deployments. Prior studies measure QUIC hypergiant deployment by counting VIPs through active measurements [52, 67], evaluate spread of early QUIC versions in MAWI traffic data, netflow from a European Tier-1 ISP, and a European IXP [52] or evaluate implementation behavior, interoperability, and performance of public test instances and sandboxed deployments [33, 44, 47, 55, 60, 68]. In contrast to prior research, our work uses less intrusive passive measurements (QUIC backscatter traffic) and is more accurate by counting L7 loadbalancers instead of VIPs. We extend telescope usage from quantifying protocol attacks to understand QUIC configurations in the wild by leveraging standardized protocol mechanics and traffic solicited by malicious actors.

Stack Configurations. Related work [34, 45] focuses on the detection and classification of TCP anomalies not on the detection of individual stack configurations. They use testbeds or traffic traces from real networks but not darknet traffic provided by telescopes. To the best of our knowledge, no analysis of stack configurations (*e.g.,* inference of retransmission timeouts and number of retransmissions) neither on QUIC nor TCP has been performed. We consider this future work.

Inferring Loadbalancers. Measurement methods that identify loadbalancers or larger CDN infrastructure use traceroute-like tools (*e.g.,* [3, 63]) or build on IP addresses made available via the DNS (*e.g.,* [1, 16]). Similarly, machine-learning techniques for traffic classification of TLS and QUIC [2, 40, 56, 61, 66] and TLS fingerprinting using handshake variations [29] are used to derive services and hypergiant deployments. The same services deployed as off-nets can thus reveal off-net footprints. While active probing, DNS, and traffic classification can quantify the number of VIPs they do not identify L7LBs. [37] count L7LBs through successful QUIC connections until a failure occurs when reusing connection IDs. Since loadbalancing does not guarantee all L7LBs receive a connection before forwarding to the same L7LB again, the number of L7LBs is only a lower bound. We use information encoded in connection IDs to unveil the number of L7LBs and structure of frontend clusters at Meta, a previously hidden property.

8 Discussion

In this work, we extracted various deployment details from meta-information encoded in QUIC headers. Although QUIC was designed to hide meta-information, we find that efficient loadbalancing and client migration require additional information encoding. This does not leak details about clients but about the server infrastructure of hypergiants.

Why use passive measurements? With our analysis we show that (*i*) even small amounts of backscatter reveal QUIC stack configurations of hypergiants, and (*ii*) enumeration of loadbalancer instances is possible at high fidelity, with less than 1,000 spoofed QUIC connections in backscatter.

Any observation from passive measurements is reproducible with active measurements, but active measurements require prior knowledge on potential targets, cost additional network traffic, and might trigger intrusion detection. Telescope measurements can help to improve active measurements as they shed light on real world QUIC traffic. Given the variety of existing QUIC libraries and their potential configurations, reproducing realistic behavior for active measurements can be challenging. Insights from passive data should be used to guide future scan campaigns.

Is passive data biased? Network telescopes capture scans and responses to spoofed traffic. This does not provide insights into Web clients but into QUIC servers and software of malicious actors. First, it exposes QUIC clients that are used for benign or malicious scans. Second, additionally to measuring QUIC features current server deployments offer, we measure what clients and servers agree on when they communicate. Spoofed traffic is primarily triggered by malicious activities (*e.g.,* botnets), which allows us to indirectly derive insights into software deployed in such environments.

Our telescope does not receive traffic from all PoPs and VIPs, but we were able to show that a single VIP is enough to unveil a substantial part of L7LBs for a given PoP. Given that homogeneous configurations across PoPs are common because they ease maintenance, we argue that not observing all PoPs is not a short-coming. Additionally, our findings are confirmed through active measurements, analysis of flow records, and by prior work [67]. Based on our complementary active measurements, analysis of flow data, and discussions with the operator community, we are confident that insights resulting from backscatter traffic are valid.

Will deployment of structured connection IDs increase? Can we apply the same methods to other deployments? Structured CIDs may serve as a fingerprint of specific hypergiants. We find that Google migrated to such connection IDs in 2023, and we observe distinct information encoding from Akamai, Amazon, Apple, Cloudflare, Fastly, Meta, and Microsoft in backscatter. We argue that the usage will increase over time since they simplify fine-grained provider controlled routing, but standardization might limit the cardinality of unique identification properties and as such our method. Advanced QUIC features such as client migration even require additional data encoded in such IDs to prevent overhead from synchronizing connection state.

Our detection of off-net deployments is applicable to other deployments and measurement methods such as flow records without ground-truth knowledge from open source implementations, while detection of layer 7 loadbalancers is limited to Meta, since only Meta uses a cleartext encoding.

What are the implications from knowing the number of loadbalancers? Encoding the destination loadbalancer into a connection ID enables clients to steer traffic to specific loadbalancer instances. This is unwanted behavior because attackers could direct traffic to a single loadbalancer, bypassing single point of failure mitigation. Although the number of loadbalancer instances does not reveal the underlying capacity and compute power, knowing the distribution in a geographic region or size of a single cluster can be used to estimate the load necessary to overload that PoP. This information is not only valuable for attackers but also for competitors, allowing them to (*i*) anticipate business opportunities and local competition, (*ii*) the importance of a region, and (*iii*) improving their own infrastructure.

9 Conclusion and Outlook

In this paper, we showed how passive backscatter traffic recorded at the Internet edge can be utilized to closely monitor the rollout of QUIC and in particular reveal deployed configurations of local QUIC stacks and larger distribution infrastructures. We explicitly quantify layer 7 loadbalancers, which were previously disguised behind virtual IP addresses.

Since passive measurements are completely uncontrolled, *i.e.,* based on data triggered by third parties, we used alternative means—captured flow data and active scans—to validate our findings.

Future Trends based on Common Protocols. Scans reveal that hypergiants operate between 1.6× and 2.1× (Cloudflare, Meta), up to 7.5× (Google) as many TLS/TCP endpoints by number of VIPs than QUIC, but the number of QUIC endpoints is rapidly increasing. We observe 2.5× (Meta) to 3× (Google) the amount of QUIC endpoints in 2025 than in 2021. The Cloudflare deployment increased by 17% during this time. A continued rise in endpoints suggests a potential for higher victim counts in the future. Understanding the reasons for not seeing significant backscatter from all QUIC-enabled hypergiants (*e.g.,* fewer attacks or filtering) will be part of our future work. We argue that the proposed approach will gain precision with increasing QUIC deployment.

Acknowledgments

This work was partly supported by the Federal Ministry of Research, Technology and Space (BMFTR) within the project AI.Auto-Immune (16KIS2333 and 16KIS2332K).

References

[1] Bernhard Ager, Wolfgang Mühlbauer, Georgios Smaragdakis, and Steve Uhlig. 2011. Web Content Cartography. In *Proc. of ACM IMC*. ACM, New York, NY, USA, 585–600.

[2] Iman Akbari, Mohammad A. Salahuddin, Leni Ven, Noura Limam, Raouf Boutaba, Bertrand Mathieu, Stephanie Moteau, and Stephane Tuffin. 2021. A Look Behind the Curtain: Traffic Classification in an Increasingly Encrypted Web. *Proc. ACM Meas. Anal. Comput. Syst.* 5, 1, Article 04 (feb 2021), 26 pages. https://doi.org/10.1145/3447382

[3] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. 2006. Avoiding traceroute anomalies with Paris traceroute. In *Proc. of ACM IMC* (Rio de Janeriro, Brazil). ACM, New York, NY, USA, 153–158.

[4] Tom Barbette, Chen Tang, Haoran Yao, Dejan Kostic, Gerald Q. Maguire Jr., Panagiotis Papadimitratos, and Marco Chiesa. 2020. A High-Speed Load-Balancer Design with Guaranteed Per-Connection-Consistency. In *17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2020, Santa Clara, CA, USA, February 25-27, 2020*, Ranjita Bhagwan and George Porter (Eds.). USENIX Association, 667–683. https://www.usenix.org/conference/nsdi20/presentation/barbette

[5] M. Bishop. 2022. *HTTP/3.* RFC 9114. IETF. https://doi.org/10.17487/RFC9114

[6] Timm Böttger, Felix Cuadrado, Gareth Tyson, Ignacio Castro, and Steve Uhlig. 2018. Open Connect Everywhere: A Glimpse at the Internet Ecosystem through the Lens of the Netflix CDN. *SIGCOMM Comput. Commun. Rev.* 48, 1 (apr 2018), 28–34. https://doi.org/10.1145/3211852.3211857

[7] CAIDA. 2012. The UCSD Network Telescope. Website. https://www.caida.org/projects/network_telescope/ Last Access: May 2025.

[8] CAIDA. 2022. Spoofer Project. https://www.caida.org/projects/spoofer/ Last Access: June 2025.

[9] CESNET. 2019. ipfixprobe – IPFIX flow exporter with DPDK support. GitHub Repository. https://github.com/CESNET/ipfixprobe Last Access: September 2025.

[10] Cloudflare. 2018. QUICHE. Github Repository. https://github.com/cloudflare/quiche Last Access: Jun 2025.

[11] Cloudflare. 2025. Cloudflare Radar: Adoption & Usage. https://radar.cloudflare.com/adoption-and-usage?dateRange=52w Last Access: June 2025.

[12] Cloudflare. 2025. UDP Graceful Restart Marshal. GitHub Repository. https://github.com/cloudflare/udpgrm/tree/main Last Access: June 2025.

[13] Michael Collins. 2021. Acknowledged Scanners. GitLab Repository. https://gitlab.com/mcollins_at_isi/acknowledged_scanners Last Access: May 2025.

[14] Yong Cui, Tianxiang Li, Cong Liu, Xingwei Wang, and Mirja K ühlewind. 2017. Innovating Transport with QUIC: Design Approaches and Research Challenges. *IEEE Internet Comput.* 21, 2 (2017), 72–76. https://doi.org/10.1109/MIC.2017.44

[15] Owen DeLong. 2022. Re: Scanning the Internet for Vulnerabilities. https://seclists.org/nanog/2022/Jun/269 Last Access: June 2025.

[16] Trinh Viet Doan, Roland van Rijswijk-Deij, Oliver Hohlfeld, and Vaibhav Bajpai. 2022. An Empirical View on Consolidation of the Web. *ACM Trans. Internet Technol.* 22, 3 (feb 2022), 30 pages. https://doi.org/10.1145/3503158

[17] Martin Duke, Nick Banks, and Christian Huitema. 2024. *QUIC-LB: Generating Routable QUIC Connection IDs.* Internet-Draft – work in progress 18. IETF. https://datatracker.ietf.org/doc/html/draft-ietf-quic-load-balancers-18

[18] Martin Duke, Nick Banks, and Christian Huitema. 2025. *QUIC-LB: Generating Routable QUIC Connection IDs.* Internet-Draft – work in progress 21. IETF. https://datatracker.ietf.org/doc/html/draft-ietf-quic-load-balancers-21

[19] Zakir Durumeric, David Adrian, Phillip Stephens, Eric Wustrow, and J. Alex Halderman. 2024. Ten Years of ZMap. In *Proc. of ACM IMC*. ACM, New York, NY, USA, 139–148. https://doi.org/10.1145/3646547.3689012

[20] W. Eddy. 2007. *TCP SYN Flooding Attacks and Common Mitigations.* RFC 4987. IETF. https://doi.org/10.17487/RFC4987

[21] Daniel E. Eisenbud, Cheng Yi, Carlo Contavalli, Cody Smith, Roman Kononov, Eric Mann-Hielscher, Ardas Cilingiroglu, Bin Cheyney, Wentao Shang, and Jinnah Dylan Hosein. 2016. Maglev: A Fast and Reliable Software Network Load Balancer. In *Proc. of USENIX NSDI*. USENIX Association, Santa Clara, CA, 523–535. https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eisenbud

[22] Marwan Fayed, Lorenz Bauer, Vasileios Giotsas, Sami Kerola, Marek Majkowski, Pavel Odintsov, Jakub Sitnicki, Taejoong Chung, Dave Levin, Alan Mislove, Christopher A. Wood, and Nick Sullivan. 2021. The Ties That Un-Bind: Decoupling IP from Web Services and Sockets for Robust Addressing Agility at CDN-Scale. In *Proc. of ACM SIGCOMM*. ACM, New York, NY, USA, 433–446. https://doi.org/10.1145/3452296.3472922

[23] Petros Gigis, Matt Calder, Lefteris Manassakis, George Nomikos, Vasileios Kotronis, Xenofontas Dimitropoulos, Ethan Katz-Bassett, and Georgios Smaragdakis. 2021. Seven Years in the Life of Hypergiants' Off-Nets. In *Proc. of ACM SIGCOMM*. ACM, New York, NY, USA, 516–533. https://doi.org/10.1145/3452296.3472928

[24] Google. 2021. QUICHE. GitHub Repository. https://github.com/google/quiche Last Access: Jul 2025.

[25] Ronald F. Guilmette et al. 2022. Scanning the Internet for Vulnerabilities. https://seclists.org/nanog/2022/Jun/250 Last Access: June 2025.

[26] Raphael Hiesgen, Marcin Nawrocki, Marinho Barcellos, Daniel Kopp, Oliver Hohlfeld, Echo Chan, Roland Dobbins, Christian Doerr, Christian Rossow, Daniel R. Thomas, Mattijs Jonker, Ricky Mok, Xiapu Luo, John Kristoff, Thomas C. Schmidt, Matthias Wählisch, and KC Claffy. 2024. The Age of DDoScovery: An Empirical Comparison of Industry and Academic DDoS Assessments. In *Proc. of ACM Internet Measurement Conference (IMC)*. ACM, New York, 259–279. https://doi.org/10.1145/3646547.3688451

[27] Raphael Hiesgen, Marcin Nawrocki, Alistair King, Alberto Dainotti, Thomas C. Schmidt, and Matthias Wählisch. 2022. Spoki: Unveiling a New Wave of Scanners through a Reactive Network Telescope. In *Proc. of USENIX Security*. USENIX Association, Berkeley, CA, USA, 431–448. https://www.usenix.org/conference/usenixsecurity22/presentation/hiesgen

[28] Fahad Hilal, Patrick Sattler, Kevin Vermeulen, and Oliver Gasser. 2024. A First Look At IPv6 Hypergiant Infrastructure. *Proc. ACM Netw.* 2, CoNEXT2 (June 2024), 11:1–11:25. https://doi.org/10.1145/3656300

[29] Martin Husák, Milan Cermák, Tomás Jirsík, and Pavel Celeda. 2016. HTTPS traffic analysis and client identification using passive SSL/TLS fingerprinting. *EURASIP J. Inf. Secur.* 2016 (2016), 6. https://doi.org/10.1186/s13635-016-0030-7

[30] Facebook Incubator. 2019. mvfst. GitHub Repository. https://github.com/facebookincubator/mvfst Last Access: June 2025.

[31] Facebook Incubator. 2019. mvfst – encodeConnectionId function. GitHub Repository. https://github.com/facebookincubator/mvfst/blob/9603981b74a7d28004331e6fd6dbf2882ad2c291/quic/codec/DefaultConnectionIdAlgo.cpp#L323 Last Access: June 2025.

[32] J. Iyengar and M. Thomson. 2021. *QUIC: A UDP-Based Multiplexed and Secure Transport*. RFC 9000. IETF. https://doi.org/10.17487/RFC9000

[33] Benedikt Jaeger, Johannes Zirngibl, Marcel Kempf, Kevin Ploch, and Georg Carle. 2023. QUIC on the Highway:Evaluating Performance on High-rate Links. In *In Proc. of IFIP Networking Conference (IFIP Networking)*. IFIP, Barcelona, Spain.

[34] Wolfgang John and Sven Tafvelin. 2007. Analysis of internet backbone traffic and header anomalies observed. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement* (San Diego, California, USA) *(IMC '07)*. Association for Computing Machinery, New York, NY, USA, 111–116. https://doi.org/10.1145/1298306.1298321

[35] Nick Jones. 2018. Get a head start with QUIC. Blog. https://blog.cloudflare.com/head-start-with-quic/ Last Access: May 2025.

[36] Matt Joras. 2024. Re: Why isn't QUIC growing? https://mailarchive.ietf.org/arch/msg/quic/SqMCCWSyVeI46Exu4I-MCAAgg_w/ Last Access: June 2025.

[37] Liliana Kistenmacher, Anum Talpur, and Mathias Fischer. 2025. QUIC-Aware Load Balancing: Attacks and Mitigations. In *55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2025, Naples, Italy, June 23-26, 2025*. IEEE, 524–536. https://doi.org/10.1109/DSN64029.2025.00056

[38] Thomas Koch, Ethan Katz-Bassett, John Heidemann, Matt Calder, Calvin Ardi, and Ke Li. 2021. Anycast In Context: A Tale of Two Systems. In *Proc. of ACM SIGCOMM*. ACM, New York, NY, USA, 398–417. https://doi.org/10.1145/3452296.3472891

[39] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. 2010. Internet Inter-Domain Traffic. *ACM Sigcomm Computer Communication Review* 40, 4 (Aug 2010), 75–86. https://doi.org/10.1145/1851275.1851194

[40] Jan Luxemburk, Karel Hynek, and Tomáš Čejka. 2023. Encrypted traffic classification: the QUIC case. In *Proc. of Network Traffic Measurement and Analysis Conference (TMA)* (Naples, Italy). IFIP, Naples, Italy.

[41] Gordon Fyodor Lyon. 2009. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Nmap Project, Chapter Legal Issues.

[42] Diego Madariaga, Lucas Torrealba, Javier Madariaga, Javiera Berm údez, and Javier Bustos-Jiménez. 2020. Analyzing the Adoption of QUIC From a Mobile Development Perspective. In *Proc. of the Workshop on the Evolution, Performance, and Interoperability of QUIC* (Virtual Event, USA) *(EPIQ '20)*. ACM, New York, NY, USA, 35–41. https://doi.org/10.1145/3405796.3405830

[43] Alexander Männel, Jonas Mücke, kc Claffy, Max Gao, Ricky K. P. Mok, Marcin Nawrocki, Thomas C. Schmidt, and Matthias Wählisch. 2025. Lessons Learned from Operating a Large Network Telescope. In *Proc. of ACM Special Interest Group on Data Communication (SIGCOMM)*. ACM, New York, 826–841. https://doi.org/10.1145/3718958.3754347

[44] Robin Marx, Joris Herbots, Wim Lamotte, and Peter Quax. 2020. Same Standards, Different Decisions: A Study of QUIC and HTTP/3 Implementation Diversity. In *Proceedings of the 2020 Workshop on the Evolution, Performance, and Interoperability of QUIC, EPIQ@SIGCOMM 2020, Virtual Event, USA, August 10-14, 2020*, Jörg Ott and Lars Eggert (Eds.). ACM, New York, NY, USA, 14–20. https://doi.org/10.1145/3405796.3405828

[45] Marco Mellia, Michela Meo, Luca Muscariello, and Dario Rossi. 2008. Passive analysis of TCP anomalies. *Comput. Networks* 52, 14 (2008), 2663–2676. https://doi.org/10.1016/J.COMNET.2008.05.010

[46] A. Minaburo and L. Toutain. 2023. *A YANG Data Model for Static Context Header Compression (SCHC)*. RFC 9363. IETF. https://doi.org/10.17487/RFC9363

[47] Jonas Mücke, Marcin Nawrocki, Raphael Hiesgen, Thomas C. Schmidt, and Matthias Wählisch. 2024. ReACKed QUICer: Measuring the Performance of Instant Acknowledgments in QUIC Handshakes. In *Proc. of ACM Internet Measurement Conference (IMC)*. ACM, New York, 389–400. https://doi.org/10.1145/3646547.3689022

[48] Marcin Nawrocki, Raphael Hiesgen, Thomas C. Schmidt, and Matthias Wählisch. 2021. QUICsand: Quantifying QUIC Reconnaissance Scans and DoS Flooding Events. In *Proc. of ACM Internet Measurement Conference (IMC)*. ACM, New York, 283–291. https://doi.org/10.1145/3487552.3487840

[49] Marcin Nawrocki, Pouyan Fotouhi Tehrani, Raphael Hiesgen, Jonas Mücke, Thomas C. Schmidt, and Matthias Wählisch. 2022. On the Interplay between TLS Certificates and QUIC Performance. In *Proc. of 18th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. ACM, New York, NY, USA, 204–213. https://dl.acm.org/doi/10.1145/3555050.3569123

[50] Maxime Piraux, Quentin De Coninck, and Olivier Bonaventure. 2018. Observing the Evolution of QUIC Implementations. In *Proc. of the Workshop on the Evolution, Performance, and Interoperability of QUIC* (Heraklion, Greece) *(EPIQ'18)*. ACM, New York, NY, USA, 8–14. https://doi.org/10.1145/3284850.3284852

[51] Philipp Richter, Florian Wohlfart, Narseo Vallina-Rodriguez, Mark Allman, Randy Bush, Anja Feldmann, Christian Kreibich, Nicholas Weaver, and Vern Paxson. 2016. A Multi-Perspective Analysis of Carrier-Grade NAT Deployment. In *Proc. of ACM IMC*. ACM, New York, NY, USA, 215–229. https://doi.org/10.1145/2987443.2987474

[52] Jan Rüth, Ingmar Poese, Christoph Dietzel, and Oliver Hohlfeld. 2018. A First Look at QUIC in the Wild. In *Passive and Active Measurement (LNCS, Vol. 10771)*. Springer Nature, Switzerland, 255–268.

[53] Morteza Safaei Pour, Christelle Nader, Kurt Friday, and Elias Bou-Harb. 2023. A Comprehensive Survey of Recent Internet Measurement Techniques for Cyber Security. *Computers & Security* 128 (2023), 103123. https://doi.org/10.1016/j.cose.2023.103123

[54] D. Schinazi and E. Rescorla. 2023. *Compatible Version Negotiation for QUIC*. RFC 9368. IETF. https://doi.org/10.17487/RFC9368

[55] Marten Seemann and Jana Iyengar. 2020. Automating QUIC Interoperability Testing. In *Proceedings of the 2020 Workshop on the Evolution, Performance, and Interoperability of QUIC, EPIQ@SIGCOMM 2020, Virtual Event, USA, August 10-14, 2020*, Jörg Ott and Lars Eggert (Eds.). ACM, New York, NY, USA, 8–13. https://doi.org/10.1145/3405796.3405826

[56] Wazen M. Shbair, Thibault Cholez, Jerome Francois, and Isabelle Chrisment. 2016. A multi-level framework to identify HTTPS services. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. IFIP, Istanbul, Turkey, 240–248. https://doi.org/10.1109/NOMS.2016.7502818

[57] Tanya Shreedhar, Rohit Panda, Sergey Podanev, and Vaibhav Bajpai. 2021. Evaluating QUIC Performance over Web, Cloud Storage and Video Workloads. *IEEE Transactions on Network and Service Management* (2021), 16 pages. https://doi.org/10.1109/TNSM.2021.3134562

[58] Raffaele Sommese, Leandro Bertholdo, Gautam Akiwate, Mattijs Jonker, Roland van Rijswijk-Deij, Alberto Dainotti, KC Claffy, and Anna Sperotto. 2020. MAnycast2: Using Anycast to Measure Anycast. In *Proc. of ACM IMC*. ACM, New York, NY, USA, 456–463. https://doi.org/10.1145/3419394.3423646

[59] Terin Stock. 2020. High Availability Load Balancers with Maglev. https://blog.cloudflare.com/high-availability-load-balancers-with-maglev/

[60] Kashyap Thimmaraju and Björn Scheuermann. 2021. Count Me If You Can: Enumerating QUIC Servers Behind Load Balancers. In *Proc. of ECEASST NetSys*. ECEASST, Dortmund, Germany, 5 pages. http://dx.doi.org/10.14279/tuj.eceasst.80.1172.1077

[61] Van Tong, Hai Anh Tran, Sami Souihi, and Abdelhamid Mellouk. 2018. A Novel QUIC Traffic Classifier Based on Convolutional Neural Networks. In *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Abu Dhabi, UAE, 1–6. https://doi.org/10.1109/GLOCOM.2018.8647128

[62] Piet De Vaere, Tobias Bühler, Mirja Kühlewind, and Brian Trammell. 2018. Three Bits Suffice: Explicit Support for Passive Measurement of Internet Latency in QUIC and TCP. In *Proc. of ACM IMC*. ACM, New York, NY, USA, 22–28. https://dl.acm.org/citation.cfm?id=3278535

[63] Kevin Vermeulen, Justin P. Rohrer, Robert Beverly, Olivier Fourmaux, and Timur Friedman. 2020. Diamond-Miner: Comprehensive Discovery of the Internet's Topology Diamonds. In *Proc. of USENIX NSDI*. USENIX Association, USA, 479–494.

[64] Konrad Wolsing, Jan Rüth, Klaus Wehrle, and Oliver Hohlfeld. 2019. A Performance Perspective on Web Optimized Protocol Stacks: TCP+TLS+HTTP/2 vs. QUIC. In *Proc. of the Applied Networking Research Workshop (ANRW '19)*. ACM, New York, NY, USA, 1–7. https://doi.org/10.1145/3340301.3341123

[65] Xfinity. 2021. Acceptable Use Policy for Xfinity Internet. https://www.xfinity.com/corporate/customers/policies/highspeedinternetaup Last Access: May 2025.

[66] Lixuan Yang, Alessandro Finamore, Feng Jun, and Dario Rossi. 2021. Deep Learning and Zero-Day Traffic Classification: Lessons Learned From a Commercial-Grade Dataset. *IEEE Transactions on Network and Service Management* 18, 4 (2021), 4103–4118. https://doi.org/10.1109/TNSM.2021.3122940

[67] Johannes Zirngibl, Philippe Buschmann, Patrick Sattler, Benedikt Jaeger, Juliane Aulbach, and Georg Carle. 2021. It's over 9000: Analyzing Early QUIC Deployments with the Standardization on the Horizon. In *Proc. of ACM IMC*. ACM, New York, NY, USA, 261–275. https://doi.org/10.1145/3487552.3487826

[68] Johannes Zirngibl, Florian Gebauer, Patrick Sattler, Markus Sosnowski, and Georg Carle. 2024. QUIC Hunter: Finding QUIC Deployments and Identifying Server Libraries Across the Internet. In *Passive and Active Measurement: 25th International Conference, PAM 2024, Virtual Event, March 11–13, 2024, Proceedings, Part II*. Springer-Verlag, Berlin, Heidelberg, 273–290. https://doi.org/10.1007/978-3-031-56252-5_13

A Ethics

Our proposed method aims for less intrusive measurements.

Backscatter from Network Telescope. We introduce passive measurements capturing unsolicited traffic via a network telescope. The telescope may receive sensitive information, *e.g.*, backscatter from victims of source spoofed attacks and compromised hosts. Unanonymized data is only processed in a controlled environment, and we only share aggregated statistics and our analysis.

NREN Flow Records. The privacy of NREN users is utmost important to us. All work based on NREN data was conducted with great care. We want to emphasize that the flow data we received did not include full client IP addresses but was already stripped. Therefore, it is not possible to trace the identity of the data subjects. Moreover, the dataset consists solely of flow records; no PCAP files were included, and we have never had access to payload data, apart from metadata available in QUIC handshakes. The NREN legitimately obtained this data on the basis of a legitimate interest (i.e., not consent), among other things, for the purpose of ensuring the further development of services provided to the scientific and research community (which is also why the data has been provided to us).

Active Measurements. All active measurements conducted in this study complement or verify our results using common techniques and adhere to best practices. To reduce potential conflicts, we limit the amount of active scanning.

B Artifacts

Due to agreements with the data providers we cannot publish telescope data. To facilitate further research, we publish the QUIC flows sampled from CESNET and our analysis.

Data set: QUIC flow records, active measurements of Meta on-net deployments

Run-time environment: Python, Jupyter Notebooks

How much disk space is required? ~100 GB

How much memory is required? 256 GB

Publicly available? Yes

Archived? Yes.
Analysis: https://doi.org/10.5281/zenodo.17232715
CESNET flow data: https://doi.org/10.5281/zenodo.17249078

C Changes in the UCSD Network Telescope

The address space of the UCSD network telescope, located in a /9 and /10 IPv4 prefix, changes due to legitimate use by the owners of the address space [43]. Figure 14 shows the size of the UCSD network telescope decreases throughout our measurement period and highlights our telescope observation periods. At the end of our observation period, the telescope is ~6% smaller than in our first observation period. In 2023, ~2 % more addresses are part of the telescope compared to 2021. The maximum difference equals the size of a ~/20 subnet. Due to missing ground-truth it is impossible to determine how that influences telescope traffic. Figure 15 shows the dominance of a few target /24 subnets in backscatter. Within each dataset, a single /24 subnet contributes >43% of the backscatter traffic. They contain repeated octets. Not all subnets with large shares of traffic in one dataset are part of the telescope throughout our measurement period. However, the two most active subnets are present throughout our observation period. We are unable to dissect how this affects the overall dataset and amount of QUIC backscatter received.

The UCSD network telescope can experience dataloss [43]. Our dataset it affected by this, but we are unable to determine to which extend more QUIC backscatter could have been received. However, our method is based on the volume of received QUIC packets, not requiring reception of specific backscatter.

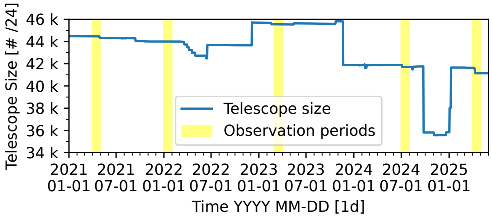


Fig. 14. Size of the UCSD network telescope and observation periods. The size of the UCSD network telescope decreases within our measurement period.

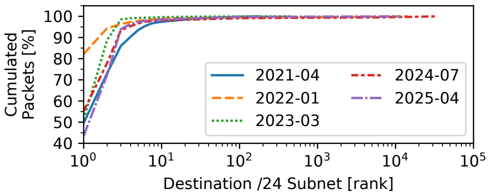


Fig. 15. Cumulated reception of QUIC backscatter packets by /24 telescope subnet. The majority of QUIC backscatter is received by only a few subnets.

D Details on Meta Connection IDs

The Meta QUIC implementation mvfst, encodes information in CIDs. The meaning of the encoding is provided in their public source code [31]. Table 5 presents details about the encoding schema.

Table 5. Meta SCID structure includes clear text host IDs and reduces randomness to less than half of the SCID length.

	Bits of the SCID				
SCID Version	Version	Host ID	Worker ID	Process ID	Remaining random bits
1	0-1	2-17	18-25	26	27-63
2	0-1	8-31	32-39	40	2-7,41-63
3	0-1	8-39	40-47	48	2-7,49-63

E Information encoding in the QUIC header

QUIC tries to hide metadata but *Initial* and *Handshake* messages carry data about the deployed QUIC stack. This includes in cleartext the QUIC version a client offers and a server accepts, the destination and source connection IDs, the retry token and the packet Length information. Only type-specific nits, packet number length, and the packet number are protected by header protections keys. The respective keys are derived using a version specific salt and the client DCID set by the client in its first flight for *Initial* packets and using asymmetric cryptography for all other headers. As such the header protection keys cannot be calculated in backscatter, because we lack the DCID from the client *Initial* or key material.

Table 6 lists all QUIC header fields, whether they are protected, and other limitations for successful extraction.

F Measurement statistics

Sanitization removes up to 99.9% of packets, which mainly consist of research scans targeting the complete telescope; those scans have little significance to our analysis, since research infrastructure is well-documented and does not have to be inferred.

Backscatter. Table 7 lists the number of connections in backscatter in the respective year. We determine QUIC connections by unique SCID, DCID, source and destination IP addresses, and destination port.

Table 6. QUIC metadata in long (Initial, Handshake, Version Negotiation, 0-RTT, Retry) and short header packets (1-RTT).

	Protected in	
Field	Long Header	Short Header
Header Form	✘	✘
Fixed Bit	✘	✘
Long Packet Type	✘	-
QUIC Type-Specific Bits (Reserved Bits, Spin Bit, Key Phase)	✓	✓
Version	✘	-
DCID Length	✘	-
DCID	✘	✘[†]
SCID Length	✘	-
SCID	✘	-
Token Length	✘	-
Token	✘	-
Length	✘	-
Packet Number Length	✓	-
Packet Number	✓	✓

[†] Without knowledge of the DCID length due to its variable length, the DCID cannot be extracted from short header packets.

Table 7. Number of QUIC connections in backscatter. We observe a strong increase in the number of QUIC connections.

	QUIC connections from Source Networks[#]								
Year	Cloudflare	Google	Meta	Akamai	Amazon	Apple	Fastly	Microsoft	Others
2021	39	62,403	9,552	-	1	-	-	-	9,290
2022	170	113,879	63,615	15	2	2	-	102	47,030
2023	1,126	314,219	257,002	5,173	3,356	108	276	2,422	133,083
2024	163,117	147,936	162,088	13,787	4,485	27,958	7,478	208	160,835
2025	68,830	160,909	231,880	14,984	3,435	35,620	9,597	707	73,645

NREN Flow Data. Table 8 shows the number of IP addresses of large content providers in flow records from a European NREN in 2024 and 2025. Table 9 shows the number of flows of large content providers in the same dataset.

Table 8. Number of IP addresses in flow records from European NREN in 2024 and 2025.

	IP addresses from source network [#]								
Year	Akamai	Amazon	Apple	Cloudflare	Fastly	Google	Meta	Microsoft	Other ASes
2024	1,568	1,927	116	5,425	132	7,217	449	464	1,363
2025	3,012	1,088	115	8,093	184	7,478	698	1,389	2,673

G Off-net Classifiers

Aside from SCID structure, on-net deployments differ in configured retransmission intervals, packet lengths, and enabled packet coalescence. Subsequently, we detail these classifiers and show their

Table 9. Number of flows in flow records from European NREN in 2024 and 2025.

	Flows from Source Network [#]								
Year	Akamai	Amazon	Apple	Cloudflare	Fastly	Google	Meta	Microsoft	Others ASes
2024	80,081	37,115	1,603	176,460	79,280	7,470,552	606,561	115,381	46,339
2025	418,531	55,642	3,663	378,468	299,016	9,352,327	2,535,108	284,153	110,270

performance in backscatter. Table 10 presents true positive rate, false positive rate *etc.* of identifying Google and Meta off-net servers. We do not show classifier performance for Cloudflare, since none of the potential off-net candidates allowed for QUIC connections at the time of our measurement. The bad performance of the Retransmission Interval classifier for Google originates from the less consistent configuration, *i.e.,* not all connections have 3 retransmissions.

Retransmission Intervals. The Retransmission Interval classifiers calculate the time since the reception of the first packet of a connection. We then use the determined value of the Initial RTO and perform binary exponential backoff on this value. If the calculated value is larger than 0.4s, we tolerate packets within ±0.1s of that value. Otherwise, the set of timeouts overlaps with the timeouts used by other hypergiants, which increases the false positive rate. We classify an IP address to be operated by a given hypergiant if at least 3 of the retransmissions from distinct backoff intervals are observed in any flow from that IP address.

Packet Length(s). The packet length classifier considers an IP address to be operated by a given hypergiant if any of the Top 5 packet length(s) (see Figure 5) is contained in any packet from that address.

Packet Coalescence. An IP address is detected as coalescing if a coalesced packet originates from that address.

Table 10. Performance of classifying off-net Google and Meta servers based on backscatter traffic with the largest set of off-net candidates (2023). IP addresses were assigned to content providers using subject alternative names in certificates collected with QScanner.

Classifier	TPR	FPR	TNR	FNR	Precision	Recall	F-score
Meta							
Retransmission Intervals	0.857	0.266	0.734	0.143	0.596	0.857	0.703
Retransmission Intervals & SCIDv1	0.857	0.023	0.977	0.143	0.944	0.857	0.899
Retransmission Intervals & SCIDv1 & No Coalescence	0.857	0.021	0.979	0.143	0.949	0.857	0.901
Packet Length	0.995	0.117	0.883	0.005	0.796	0.995	0.885
Packet Length & SCIDv1 & No Coalescence	0.995	0.033	0.967	0.005	0.932	0.995	0.963
No Coalescence	1.0	0.136	0.864	0.0	0.771	1.0	0.871
SCIDv1	1.0	0.053	0.947	0.0	0.895	1.0	0.945
SCIDv1 & No Coalescence	1.0	0.044	0.956	0.0	0.912	1.0	0.954
Off-net SCIDv1	1.0	0.015	0.985	0.0	0.967	1.0	0.983
Google							
Retransmission Intervals & SCIDv1	0.374	0.006	0.994	0.626	0.951	0.374	0.536
Retransmission Intervals & SCIDv1 & Coalescence	0.374	0.006	0.994	0.626	0.951	0.374	0.536
Retransmission Intervals	0.384	0.217	0.783	0.616	0.338	0.384	0.36
Packet Length & SCIDv1 & Coalescence	0.896	0.024	0.976	0.104	0.916	0.896	0.906
Packet Length	0.909	0.181	0.819	0.091	0.59	0.909	0.716
SCIDv1 & Coalescence	0.961	0.05	0.95	0.039	0.848	0.961	0.901
Coalescence	0.972	0.345	0.655	0.028	0.448	0.972	0.613
SCIDv1	0.985	0.062	0.938	0.015	0.82	0.985	0.895

Received June 2025; accepted September 2025