Towards a Behavioral and Privacy Analysis of ECS for IPv6 DNS Resolvers

Leyao Nie, Lin He*, Guanglei Song, Hao Gao, Chenglong Li*, Zhiliang Wang*, Jiahai Yang*
Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University, Beijing, China
Quan Cheng Laboratory, Jinan, Shandong, China
*Zhongguancun Laboratory, Beijing, China

***Abstract*—The Domain Name System (DNS) is critical to Internet communications. EDNS Client Subnet (ECS), a DNS extension, allows recursive resolvers to include client subnet information in DNS queries to improve CDN end-user mapping, extending the visibility of client information to a broader range. Major content delivery network (CDN) vendors, content providers (CP), and public DNS service providers (PDNS) are accelerating their IPv6 infrastructure development. With the increasing deployment of IPv6-enabled services and DNS being the most foundational system of the Internet, it becomes important to analyze the behavioral and privacy status of IPv6 resolvers. However, there is a lack of research on ECS for IPv6 DNS resolvers.**

In this paper, we study the ECS deployment and compliance status of IPv6 resolvers. Our measurement shows that 11.12% IPv6 open resolvers implement ECS. We discuss abnormal non-compliant scenarios that exist in both IPv6 and IPv4 that raise privacy and performance issues. Additionally, we measured if the sacrifice of clients' privacy can enhance IPv6 CDN performance. We find that in some cases ECS helps end-user mapping but with an unnecessary privacy loss. And even worse, the exposure of client address information can sometimes backfire, which deserves attention from both Internet users and PDNSes.

***Index Terms*—ECS, IPv6, DNS**

I. Introduction

The Domain Name System (DNS), which resolves domain names into IP addresses, is essential for Internet communications. Large content delivery networks (CDN) and content providers (CP) use DNS to map end-users to geographically optimal edge servers [1]. Traditionally, CDNs and CPs use the local DNS resolver's IP address as proximity of the end-user to select an edge server for the end-user. This is often effective when the local resolver is provided by ISP because the end-user is usually close to the ISP local resolver. However, in recent years, the proliferation of public DNS services (e.g., Google Public DNS) has invalidated that local resolvers are close to end-users. CDNs and CPs cannot simply use the address of public DNS resolvers (PDNSes) to assign edge servers, because the PDNS is usually far away from the end-user and can degrade the end-user experience [2], [3].

To address this issue, the Internet Engineering Task Force (IETF) has standardized an extension to DNS called EDNS Client Subnet (ECS) in 2016 [4]. While traditional DNS queries do not include the IP address of the query sender, this extension allows a recursive resolver to include the client's IP subnet into the DNS query and forward it to the authoritative nameserver. Authoritative nameservers can use client IP address information to infer client location and select edge servers close to end-users.

Although there have been numerous studies on ECS [1], [5]–[8], there are still two unresolved issues. *First, the state of ECS deployment and resolution behavior in IPv6 is still unclear.* Nowadays, major content providers such as YouTube, Netflix, and Apple are accelerating their IPv4/IPv6 dual-stack deployment and have established on a large scale [9]–[11]. The CDN and DNS infrastructures on which these CPs rely heavily are also migrating to IPv6 [12], [13]. IPv6 DNS resolvers gradually adopt ECS to enhance CDN end-user mapping performance. However, while there have been efforts on ECS in IPv4, there is a lack of research discussing ECS options in IPv6. *Second, in terms of IP privacy, it is still unknown whether the finer the client subnet used by ECS can really bring greater CDN performance improvement.* While the public is increasingly concerned about cybersecurity issues, the ECS option carries client subnets in DNS requests, exposing the geographic location privacy and IP privacy of clients. In the ECS option, the SOURCE PREFIX-LENGTH field defines how much client IP information is to be attached to the DNS query, specifically, the leftmost number of bits of IP address. In general, a longer prefix length brings better CDN end-user mapping [6] but also exposes more client privacy. The choice of ECS prefix length is a tradeoff between privacy and CDN performance benefit. Yet, we do not know whether service providers can cope with fine-grained client subnet information. If they lack this capability, fine-grained client subnet will unnecessarily expose more client privacy.

In this paper, we study the landscape of ECS in IPv6 and discuss its privacy issue. We measured the deployment status of ECS among IPv6 resolvers to provide a better view of ECS in IPv6 (Section IV). We perform the ECS compliance measurement to reveal the behavior of ECS-enabled resolvers in IPv6 (Section IV-D). To understand service providers' capability of utilizing ECS, we measure the IPv6 CDN performance benefit after deploying ECS, in the hope of helping end-users and PDNSes make tradeoffs between ECS privacy and performance improvement in IPv6 (Section V). To summarize, we make the following contributions:

- We find over 14K IPv6 open resolvers and 36K IPv6 egress resolvers using three different methods. We measure their ECS-deployment rates, finding 11.12% ingress resolvers and 7.83% egress resolvers implement ECS.
- We measure the compliance of open resolvers in both

978-3-903176-51-5 © 2022 IFIP

IPv6 and IPv4 against ECS according to RFC 7871 [4]. In IPv6, we observe the non-compliant behavior of rewriting the ECS option. Almost all (96.03%) of the ECS options are rewritten by intermediate resolvers and 94.64% of the IPv6 client subnets are changed to IPv4 subnets, which goes against the original purpose of ECS.
- We measure the performance improvement of ECS-enabled content delivery service from a third-party perspective in IPv6. We find that a client cannot get the corresponding CDN performance improvement in return for IP privacy sacrifice incurred by a longer ECS prefix. We hope this could enlighten end-users and PDNSes about the tradeoff.
- We continuously probe IPv6 open resolvers and share the address dataset with interested researchers on request.

II. Background and Related Work

A. Background

With ECS, the client's IP subnet is forwarded to the authoritative nameserver in the form of a network prefix by all ECS-enabled recursive resolvers. Authoritative nameserves can infer the approximate geographic location of a client based on a portion of the client IP address and assign it to the nearest edge server (end-user mapping) for more accurate and faster content delivery [1], [7], [14], [15]. ECS has been rapidly adopted by major Internet companies such as Google to distribute user requests to their edge servers better and improve the end-user experience.

The recursive resolver sets the client IP into the ADDRESS field and then uses its maximum cacheable prefix length value to set the SOURCE PREFIX-LENGTH. For privacy reasons, full-length IP addresses are rarely used to customize DNS queries. RFC 7871 [4] recommends that the prefix length should not exceed 56 in IPv6 and 24 in IPv4. In some cases, this option can also be initialized by the stub resolver (client itself) and then sent to the recursive resolver.

B. Open Resolver Discovery

Open recursive DNS resolvers provide DNS resolution services for the entire Internet. One type of them is public DNS (PDNS), such as Google's 2001:4860:4860::8888. PDNSes are usually provided by cloud computing vendors, and they are powerful, have complete hardware facilities and large storage capacity. Another type of open resolvers is recursive resolvers that are misconfigured to provide the DNS resolution for the public, which can be a security risk [16]. With the increasing deployment of IPv6-enabled services [9], it becomes important to analyze the behavioral and privacy status of IPv6 resolvers. The IPv6 address space is so much larger than the IPv4 address space that it's impossible to scan the entire address space [17]–[19]. Hendriks et al. [20] proposed a method to find 1,038 IPv4/IPv6 dual-stacked open resolvers, assuming that most IPv6 open resolvers are migrated from the IPv4 network. AI-Dalky et al. [6] studied 145 non-whitelisted IPv6 ingress resolvers that may have deployed ECS obtained in the passive DNS log. We combine 3 methods to find 14,599 IPv6 resolvers (Section III) for ECS deployment and behavior analysis.

C. Security Concerns for ECS-enabled Resolvers

Since DNS messages are often transmitted in clear text, ECS allows anyone on the DNS resolution path to read a portion of the client IP address, which stimulates a discussion of privacy security issues with ECS. Kintis et al. [8] discussed the privacy leakage and security issues posed by ECS, implementing selective cache poisoning attacks against specific subnets/zones. Kountouras et al. [5] shows that most ECS-enabled domains appear to exacerbate existing privacy problems related to DNS without any benefit to the end-user, and the users' anonymity may be jeopardised.

Existing related works mainly focus on IPv4 measurement, and there is a lack of research revealing the ECS landscape in IPv6. To give a more comprehensive view of ECS. we measure the deployment and compliance status of both IPv6 and IPv4 resolvers (Section IV).

D. ECS-enabled CDN Performance Measurement

As for ECS-enabled CDN performance improvement, Hounsel et al. [21] mentioned that the cost incurred by ECS cache misses could negate the benefits of directing a user to a local server via ECS. Sanchez et al. [2] used Dasu to measure the HTTP latencies to connect assigned edge servers with ECS disabled and enabled, observing time savings with ECS enabled. Chen et al. [1] analyzed passive data from the perspective of ISPs, showing that ECS end-user mapping provides significant performance benefits for clients who use PDNSes, including a decrease in mapping distance, RTT, content download time, and the time-to-first-byte. However, we can only obtain measurement results for IPv4 from previous works.

To better inform IPv6 end-users about ECS, we measure the performance improvement of ECS-enabled content delivery service from a third-party perspective in IPv6 and discuss the correlation between the sacrifice of client privacy and performance improvement (Section V). We find that ECS can incur unnecessary privacy loss.

III. Discovering IPv6 Resolvers

In this section, we used three methods—IPv4 to IPv6, IPv6 Hitlist and IPv6 Address Generating Algorithm, discovering a total of 14,599 IPv6 open resolvers (ingress resolvers) and 36,237 IPv6 egress resolvers.

A. Methodology

1) IPv4 to IPv6: We verify if an IPv4 resolver is related to an IPv6 resolver, based on Hendriks et al. [20] proposed method. We carried out this measurement on December 22, 2021. First, we use ZMap to send DNS queries for the entire IPv4 address space. We collect DNS response packets that match the open resolver feature (RCODE!=5, QR=1, RA=1). We discover 1,679,851 IPv4 open resolvers in this phase.

We then verify if these IPv4 resolvers are IPv6 resolvers. To this end, we register a domain and control the corresponding

Table I
THREE TYPES OF RESOLVERS.

Protocol	Method	Resolver categories		
		Ingress	Egress	Independent
IPv4	ZMap	1,679,851	94,466	48,495
IPv6	IPv4 to IPv6	9,230	35,822	8,880
	IPv6 hitlist	478	643	16
	IP Generating	4,902	2,153	7
	Total (unique IP)	14,599	36367	8892

authoritative nameserver. Specifically, we set up four hosts. One is the sender that acts as a client to initiate DNS queries, and the others are authoritative nameservers NS1, NS4, NS6. NS1 processes "ourdomain.xyz" related queries and divides queries to NS4 or NS6 according to subdomain label (i.e., "v4only." or "v6only"); NS4 collects and processes queries for "v4only." subdomain; NS6 does that for "v6only.". We configure the resource records (RRs) in this way to distinguish the IPv4 and IPv6 connectivity of target resolvers [20]. NS6 is only reachable via IPv6 and, as such, to resolve the v6only label, the egress needs to be able to connect to N6 via IPv6. To conduct the subsequent analysis, we hardcoded the ingress IP in the DNS query. The form of query name is "$IP_{ingress}.Timestamp.Random.v6only.ourdomain.xyz$".

2) IPv6 Hitlist: We verify if a host in the IPv6 hitlist is an IPv6 open resolver. We verify the existing IPv6 addresses in the IPv6 hitlist [22]. The hitlist comes from a longitudinal active measurement study over four months, with more than 1.3B IPv6 addresses distributed in 45.2K prefixes announced by BGP. The author used a pattern-based algorithm to probe the prefixes announced by BGP, overcoming the problems of uneven address distribution and low active rate. And the verification process is the same as III-A1.

3) IPv6 Address Generating Algorithm: The third method is using active IPv6 address probing techniques based on seed addresses to generate candidate addresses and then verify if they provide public DNS services. We use efficient target generation algorithms including 6Gen [23], 6Tree [24], DET [25], and AddrMiner [26], which learn the characteristics of the IPv6 seed addresses and generate candidate addresses according to the characteristics. We feed the IPv6 addresses of the open resolvers obtained by the first two methods as seed addresses to the above algorithms and get the generated addresses. Then we use ZMap to scan UDP port 53 of the generated addresses and get hosts that open UDP port 53. Lastly, we scan the hosts that open UDP port 53 to verify if they are open resolvers. Meanwhile, we append the obtained addresses to the seed address pool and do iterative scanning.

B. Analysis

1) Resolver Classification: Our resolver dataset contains all hosts we collected from packets we received at our authoritative servers, including back-end servers that directly interact with our servers which are transparent to our front-end scanning vantage points.

We count the number of 3 types of resolvers in our dataset—ingress, egress, and independent resolver—based on their position on the DNS query resolution path.

Table II
TOP 10 ASES WITH MOST IPV6 RESOLVER.

Open resolver/Ingress			Egress			Independent		
ASN	Num.	Pct.	ASN	Num.	Pct.	ASN	Num.	Pct.
133111	5220	35.77%	3462	2803	7.71%	3462	1009	11.35%
3462	1012	6.93%	15169	2744	7.55%	4837	391	4.40%
4837	406	2.78%	28573	766	2.11%	4766	235	2.64%
4766	236	1.62%	4837	742	2.04%	16276	212	2.38%
51167	224	1.53%	7922	738	2.03%	24940	184	2.07%
16276	215	1.47%	13335	735	2.02%	51167	164	1.84%
24940	186	1.27%	7018	633	1.74%	9318	119	1.34%
7922	137	0.94%	16276	592	1.63%	7922	119	1.34%
9318	120	0.82%	7552	522	1.44%	4134	105	1.18%
4134	112	0.77%	36692	518	1.42%	8560	103	1.16%

Ingress resolvers (Open resolvers) interact directly with the client. The ingress can provide public DNS service to Internet users, so it's also called open resolvers. **Egress resolvers** interact directly with the authoritative nameserver. The source IP in the DNS query that arrives at the authoritative is classified as egress. **Independent resolvers** are both an ingress resolver and an egress resolver at the same time.

As shown in Table I, in total, we found 14,599 IPv6 open resolvers with 3 methods and 1,679,851 IPv4 open resolvers by ZMap scanning. In IPv6, 60.91% of them are independent resolvers that can carry out recursive resolution independently; in IPv4, 2.89% of them are independent resolvers. So in our dataset, more than half of IPv6 ingress resolvers directly do the DNS recursive resolution itself and do not belong to a complex DNS resolver cluster.

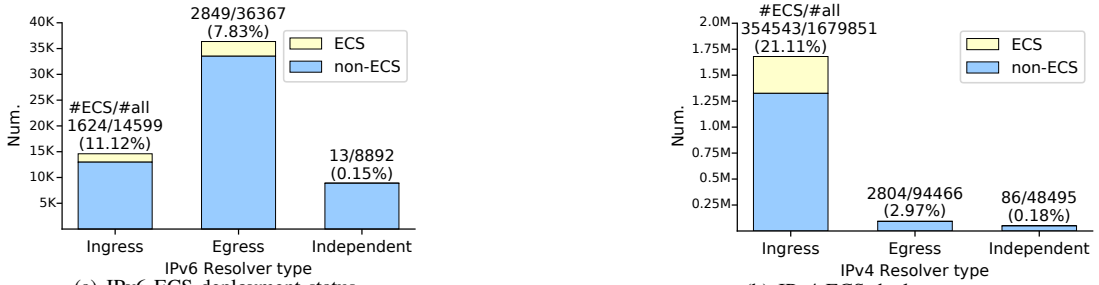
2) Resolver Distribution: We obtained the Autonomous System Number (ASN), network prefix, and country of discovered IPv6 resolvers from MaxMind GeoIP2 database [27]. We count the number of resolvers per AS and per country. We observe open resolvers coming from 1827 ASes, independent open resolvers from 1777 ASes, egress resolvers from 4751 ASes. Table II shows top 10 ASes. And we observe open resolvers from 122 countries, independent open resolvers from 120 countries, and egress resolvers from 171 countries (top countries include China, USA, Germany, Korea, Russia).

IV. ECS DEPLOYMENT AND COMPLIANCE MEASUREMENT

To understand the ECS landscape of IPv6 open resolvers and their behavior, we measured the ECS deployment rate of the discovered open resolvers and further measured the compliance status for those resolvers that support ECS. An open resolver is considered having **deployed** ECS (ECS-enabled) when it responses (ingress) or forwards (egress) queries with the ECS option. The compliance of an ECS resolver consists of two aspects in this paper.

Prefix compliance: the length of the prefix in an ECS option added by an open resolver for a client MUST NOT exceed the recommended value in RFC 7871 (IPv6: /56; IPv4: /24).

Forwarding compliance: (1) if an open resolver receives an ECS query with prefix length set to 0, it MUST NOT include client address information in queries for that client; (2) if a client initiates the ECS option with its subnet information, all subsequent ECS-enabled resolvers along the resolution path SHOULD forward the ECS option without changing the original information.



(a) IPv6 ECS deployment status (b) IPv4 ECS deployment status

Figure 1. ECS deployment status.

A. Methodology

Overall, we actively send DNS queries for "ourdomain.xyz" from a vantage point to the IPv6 open resolvers, and we collect DNS queries at the authoritative nameserver under our control. We analyzed if DNS queries are ECS-extended and check if queries are compliant (prefix compliance and forwarding compliance).

B. ECS Deployments

We send traditional DNS queries without the ECS option to open resolvers. We then verify if We then verify if they extend the ECS option when forwarding the queries to our authoritative servers.

Figure 1(a) shows the IPv6 open resolver ECS deployment measurements, up to 11.12% of discovered IPv6 open resolvers have deployed ECS. Figure 1(b) shows the IPv4 measurements, the deployment rate is 21.11%, which is higher than in IPv6. Generally, ingress resolvers' deployment rate is higher than egress resolvers'. Also interestingly, there are only a few independent resolvers adopting ECS. Almost all the ECS-enabled ingress resolvers are forwarders, and they forward DNS queries with ECS options to public DNS service clusters such as Google and OpenDNS to obtain tailored responses for clients. Independent recursive resolvers do not rely on other resolvers to perform DNS lookups. Some operators deploy independent recursive resolvers to avoid DNS cache poisoning attacks. So independent resolvers are usually located near a group of end-users who specify them to be the local DNS nameservers. To some extent, they represent end-user's location and it's not as imperative as complex public DNS resolver clusters to deploy ECS.

C. ECS Prefix Compliance

We analyze the open resolver scan data and count the prefix length (PL) they extend in the DNS queries and then classify them as compliant and non-compliant resolvers according to the recommendation of RFC 7871 [4]. The results are shown in Table III.

We found 5,204 non-compliant IPv4 open resolvers. We didn't observe IPv6 open resolvers exposing subnet prefix more than /56 in this scan. There are three possible explanations: (1) IPv6 ECS-enabled resolvers do better in compliance with RFC. (2) We didn't cover all the IPv6 open resolvers. There is still a possibility that undiscovered non-compliant IPv6 resolvers exist. (3) The open resolvers in our dataset are mainly IPv4/IPv6 dual-stacked hosts, and when they serve as forwarding proxies between the clients and egress resolvers, they switch to IPv4 protocol to perform later querying. As shown in the table, 94.64% of IPv6 open resolvers forwarded the ECS options with IPv4 prefixes, so there is some unobservable situation in the IPv6 ECS prefix compliance measurement.

Table III
OPEN RESOLVER PREFIX COMPLIANCE RESULTS.

Type	PL[a] Compliant			PL Non-compliant		
	PL	Num.	Sum	PL	Num.	Sum
IPv6	/24[b]	1,537	1,624 (100%)	≥/25[b]	0	0 (0%)
	/48	21		≥/56	0	
	/56	66				
IPv4	/22	143	417,745 (98.8%)	/25	2,142	5,204 (1.2%)
	/24	416,969		/30	5	
	/48[c]	2		/32	3,057	
	/56[c]	631				

[a]"PL" is short for "prefix length". [b] IPv4 prefix. [c] IPv6 prefix.

D. ECS Forwarding Compliance

When we measure the ECS deployment rate and compliance rate, we collect data from both the client and authoritative nameservers, and by correlation analysis, we can find some abnormal non-compliant behavior scenarios.

Scenario I: In the IPv6 open resolver scan experiment, almost all (96.03%) of the ECS options are rewritten by intermediate resolvers. And almost all (94.64%) of the ECS options are changed from an IPv6 client subnet to an IPv4 client subnet on the DNS resolution path. In the IPv4 experiment, 97.29% of the ECS options are rewritten by intermediate resolvers. We refer to this phenomenon later in this paper as "ECS rewrite" policy. We can infer that there is an IPv4-enabled device on the resolution path in this scenario and the IP of an IPv4-enabled device is used to change the ECS option.

As an example in Figure 2, (1) the stub resolver on the client-side initiates a query with a /56-ECS-option. (2) The ingress resolver is a dual-stacked device and it forwards the query to the egress resolver, but we cannot observe this packet. In theory, the ingress resolver can either forward the query with or without the original ECS option. (3) The egress resolver rewrites the ECS option with the source IP it observes (i.e., the ingress resolver's IP 186.4.154.0/24). It may make the mapping between client and edge server not geographically optimal and lose the original purpose of ECS as a result of the "ECS rewrite" policy. This behavior goes against the motivation of the ECS option, so we consider it as a minor non-compliant behavior in IPv6.

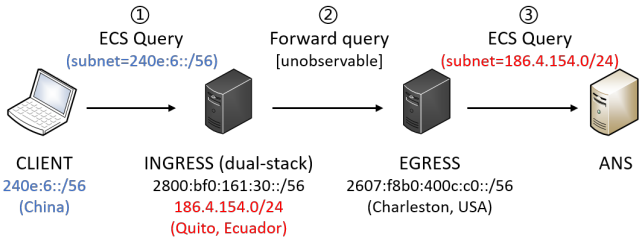


Figure 2. Abnormal scenario I: 96.03% of the ECS options are rewritten by intermediate resolvers.

Scenario II: According to RFC 7871: in the case that the client is not willing to expose privacy (actively carrying the /0 prefix), the intermediate resolver cannot pass information about the client subnet, and if it does, there is a risk of privacy disclosure. In this paper, we simulate the client to send /0 prefix in the ECS option actively and find that some open resolvers non-compliantly forward client IP address in the ECS option. For instance, the client IP is 12.12.12.12, and the client's stub resolver explicitly sends /0-ECS-queries to all open resolvers, yet we receive ECS options containing 12.12.12.12/32 and 12.12.12.0/24 at the authoritative nameserver, partially or completely exposing the source client subnet address, both of which are non-compliant.

We found 14 non-compliant IPv4 open resolvers (out of 1,679,851). 9 of them are from China, 3 from the United States, 2 from South Korea. Though we did the experiment in both IPv6 and IPv4, we didn't observe non-compliant open resolvers in IPv6. We infer this is because our dataset doesn't cover all the existing IPv6 resolvers, so there may be non-compliant resolvers we haven't discovered.

V. IPv6 ECS-CDN Performance vs. IP Privacy

In Section IV-B, we observe that many open resolvers have adopted ECS. ECS is proposed to improve content delivery. Clients expect CDN performance benefit in return for IP exposure. To inform clients about the tradeoff between performance and privacy, we measure the CDN performance improvement of ECS-enabled CDN in IPv6 and discuss the sacrifice of client privacy in the ECS option.

A. IP privacy issue in IPv6

From IPv4 to IPv6, it becomes impossible to scan the entire address space, which in some way protects devices from being discovered. Before adopting ECS, attackers in the public network have limited vision of targets. However, in IPv6, where NAT is deprecated, exposure of partial IP address in the ECS option makes it easier to scan a certain subnet and then exploit the system [28]. Saidi et al. [29] shows that up to a 19% of end-user prefixes in a large ISP can face IPv6 privacy leakage. Devices that leak a longer prefix can be leveraged as a tracking identifier for devices in the same end-user prefix.

B. Measurement

1) Measurement Targets: **a.** Public DNS: Google Public DNS, OpenDNS, AliDNS. **b.** CDN: Among all CDN providers that use ECS to optimize content delivery, Akamai is the typical representative [9]. We selected 9 content sites (e.g., streaming media, news feeds) that have deployed IPv6 Akamai's CDN globally. **c.** Probe Nodes (Vantage Points): We have deployed eight VPSs in Los Angeles, São Paulo, Frankfurt, Moscow, Beijing, Nanjing, Tokyo, and Sydney, spread across different continents.

2) Performance Metric: We use TCP connection time—the time taken to establish a TCP connection between the client and the assigned edge server. Comparing the measurement results of ECS-disabled queries (/0) and ECS-enabled queries will shed light on the influence of ECS on CDN performance.

3) Methodology: Probe nodes actively carry ECS options of different prefix lengths(/0, /32, /48, /56 or none) in DNS queries to obtain CDN edge servers' IPv6 addresses $IP_{edgeserver}$. Then, we log the network metric when probes accessing different assigned $IP_{edgeserver}$.

We repeated the ECS-CDN performance measurement for consecutive 48 hours in January 2022. For each prefix length and each PDNS, we initiate a DNS query to resolve the edge server's IP, and then access this IP 100 times and record TCP connection time each time. Last, we calculate the average for the final analysis to eliminate the effect of network jitter.

C. Analysis

The IPv6 performance measurement results are shown in Table IV. This performance measurement controls multiple variables. The following section will analyze from multiple perspectives: prefix length, public DNS, probe node.

In Table IV, to compare how the ECS prefix length contributes to performance, we set the value of /0 prefix TCP connection latency $time_{/0}$ as the reference value. And the values of the rest of the prefixes (/32, /48, /56, default) $time_{/x}$ are presented in the form of original values plus the relative change $C = \frac{time_{/x} - time_{/0}}{time_{/0}}$ in the parentheses. To give a visual indication of whether or not ECS improves end-user mapping, we use ↑ and ↓ to replace positive and negative signs of C. Note that the $default$ column reflects ECS-CDN performance in real network environment. Also, we emphasize the values that show significant performance degradation with colors.

1) ECS-disabled & ECS-enabled: Values in the "/0" column represent the latency with ECS disabled and values in the "default" column represent ECS-enabled CDN performance improvement. Suboptimal end-user mapping geographically results in performance degradation.

For Cell "GPDNS-SãoPaulo-default", /0-queries mostly resolve to South America, while default-queries mostly resolve to Texas and Mexico, which results in 10.3%↓ performance degradation. Through the logs, we found that all sites had geographically assigned suboptimal edge servers, rather than individual site latency slowing down the overall latency.

2) Multi-prefix-length: Comparing multiple prefix-lengths is the focus of this ECS-CDN performance improvement measurement, we want to know if the longer the network prefix the client exposes, the more the performance improves.

First, we look at the values of performance degradation present in the table individually: "GPDNS-Tokyo-/32", /0-ECS-queries resolve to regions in Japan, /32-ECS-queries

Table IV
IPv6 ECS-CDN PERFORMANCE.

PDNS	Probe	IPv6 TCP connection latency (relative change C)					Benefit B			
		/0	/32	/48	/56	default	/0	/32	/48	/56
GPDNS	Sydney	22.1	2.4 (89.1%↑)[a]	2.3 (89.4%↑)	2.0 (90.9%↑)	2.1 (90.7%↑)	0	0.4456	0.2236	0.1137
	Los Angeles	56.1	17.1 (69.5%↑)	11.7 (79.1%↑)	8.8 (84.4%↑)	**19.3** (65.7%↑)	0	0.3477	0.1978	0.1055
	Beijing	91.4	22.4 (75.5%↑)	22.4 (75.5%↑)	22.6 (75.3%↑)	23.8 (73.9%↑)	0	0.3777	0.1888	0.0941
	Nanjing	136.1	31.4 (76.9%↑)	30.4 (77.7%↑)	30.8 (77.4%↑)	29.7 (78.2%↑)	0	0.3846	0.1942	0.0968
	Frankfurt	9.3	2.2 (76.7%↑)	1.9 (79.2%↑)	2.0 (78.7%↑)	5.9 (36.5%↑)	0	0.3834	0.1981	0.0983
	Tokyo	6.5	31.7 (**386.0%↓**)	4.1 (36.8%↑)	3.6 (45.4%↑)	**4.6** (28.7%↑)	0	-1.9302	0.0919	0.0568
	Moscow	91.3	79.9 (12.5%↑)	83.7 (8.4%↑)	79.5 (13.0%↑)	84.4 (7.6%↑)	0	0.0625	0.0209	0.0162
	São Paulo	125.8	144.8 (**15.2%↓**)	131.7 (4.7%↓)	126.6 (0.6%↓)	138.7 (**10.3%↓**)	0	-0.0758	-0.0118	-0.0008
OpenDNS	Sydney	66.9	2.5 (96.3%↑)	2.1 (96.9%↑)	2.2 (96.8%↑)	4.8 (92.8%↑)	0	0.4815	0.2423	0.1210
	Los Angeles	62.7	7.1 (88.7%↑)	6.6 (89.5%↑)	4.4 (93.0%↑)	**6.8** (89.1%↑)	0	0.4436	0.2238	0.1162
	Beijing	45.1	25.0 (44.6%↑)	23.3 (48.3%↑)	22.5 (50.1%↑)	23.5 (48.0%↑)	0	0.2230	0.1209	0.0626
	Nanjing	69.4	29.3 (57.7%↑)	29.4 (57.6%↑)	29.3 (57.7%↑)	30.9 (55.5%↑)	0	0.2886	0.1439	0.0050
	Frankfurt	82.4	1.8 (97.8%↑)	1.9 (97.7%↑)	5.6 (93.2%↑)	1.8 (97.8%↑)	0	0.4892	0.2443	0.1165
	Tokyo	52.7	4.4 (91.7%↑)	4.8 (90.8%↑)	3.4 (93.6%↑)	**6.2** (88.2%↑)	0	0.4583	0.2271	0.1170
	Moscow	87.4	79.1 (9.5%↑)	79.9 (8.5%↑)	79.4 (9.1%↑)	79.3 (9.3%↑)	0	0.0473	0.0213	0.0114
	São Paulo	260.9	127.0 (51.3%↑)	127.7 (51.1%↑)	136.5 (47.7%↑)	139.0 (46.7%↑)	0	0.2566	0.1276	0.0596
AliDNS	Sydney	1.9	2.5 (31.7%↓)	2.1 (7.3%↓)	2.0 (6.6%↓)	2.2 (12.5%↓)	0	-0.1584	-0.0181	-0.0083
	Los Angeles	180.4	181.0 (0.3%↓)	179.4 (0.6%↑)	179.0 (0.8%↑)	**179.6** (0.5%↑)	0	-0.0016	0.0014	0.0010
	Beijing	42.9	21.9 (49.0%↑)	23.0 (46.5%↑)	23.5 (45.4%↑)	24.0 (44.1%↑)	0	0.2450	0.1162	0.0567
	Nanjing	184.2	31.4 (82.9%↑)	31.6 (82.9%↑)	31.6 (82.9%↑)	31.2 (83.1%↑)	0	0.4147	0.2072	0.1036
	Frankfurt	2.7	1.8 (31.5%↑)	2.1 (23.6%↑)	2.1 (21.3%↑)	1.9 (28.4%↑)	0	0.1575	0.0591	0.0266
	Tokyo	86.5	85.0 (1.7%↑)	78.5 (9.2%↑)	88.2 (2.0%↓)	**87.9** (1.7%↓)	0	0.0083	0.0231	-0.0025
	Moscow	86.6	86.8 (0.3%↓)	87.0 (0.4%↓)	86.4 (0.2%↑)	86.3 (0.3%↑)	0	-0.0013	-0.0010	0.0003
	São Paulo	126	125.8 (0.1%↑)	125.9 (0.1%↑)	125.7 (0.3%↑)	126.3 (0.2%↓)	0	0.0006	0.0002	0.0003

[a]The unit of TCP connection time is milliseconds. The percent value in the parentheses is relative change C.

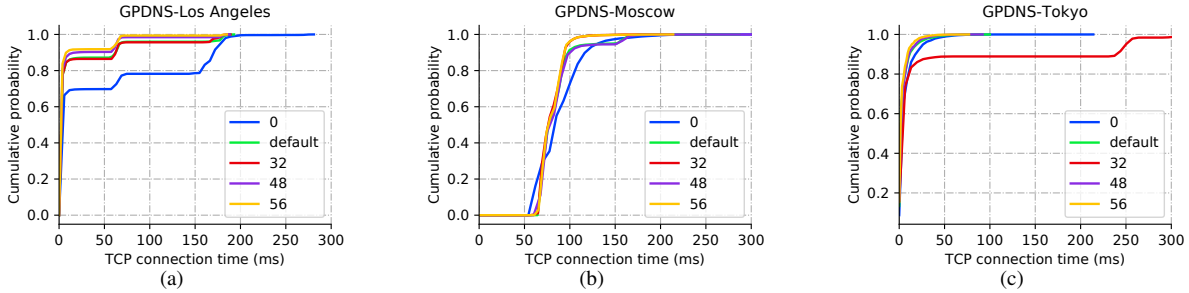


Figure 3. Comparison of the TCP connection time between 3 types of ECS-CDN "benefit". (a) CDF of connection time for "GPDNS-Los Angeles". ECS option brings significant performance improvement but we can see /32 is sufficient, /48 and /56 incur unnecessary privacy loss. (b) CDF of connection time for "GPDNS-Moscow". There is no significant improvement. (c) CDF of connection time for "GPDNS-Tokyo". ECS option backfires.

resolve to India, South America, Europe, and the Middle East, so we observe a nearly 4-fold degradation.

Second, by looking at each row of the table horizontally, we can compare performance improvement of each prefix length (32, /48, /56, default). we propose a metric to measure the benefits from exposure of client subnet (negative values means backfired)—Benefit: $B = (\frac{64-prefixlength}{64}) \times C$, in order to evaluate privacy exposure in exchange for performance improvement. We can observe from the table that there is no positive correlation between ECS prefix lengths and performance benefit, which means the client cannot get the corresponding improvement in return for IP privacy sacrifice.

To better illustrate ECS-CDN "benefit", we select 3 typical cases' CDF of TCP connection time to compare. In the first case "GPDNS-Los Angeles" (Figure 3(a)), the ECS option brought significant performance improvement but we can see /48 and /56 prefixes didn't enhance as much performance as /32 performance. If PDNS adds the /56 prefix without any policy, it will unnecessarily over-disclose the client's IP address information and geolocation information. In the second case "GPDNS-Moscow" (Figure 3(b)), the client exposed privacy for an insignificant improvement. In the third case "GPDNS-Tokyo" (Figure 3(c)), the client actively exposed /32 prefix to the public, only to see unwanted performance degradation (compared with /0 prefix). All the cases discussed above require further optimization on the practice of ECS, such as better choice of prefix length, enabling ECS or not for specific clients, but it will incur huge costs to balance between privacy and performance.

3) Multi-PDNS: Comparing the 3 PDNSes, GPDNS and OpenDNS have little difference in the mapping performance (the "default" column). However, the values for "AliDNS-LosAngeles-default" and "AliDNS-Tokyo-default" are hugely greater than the values for GPDNS and OpenDNS under the same conditions (19.3, 6.8, **179.6**; 4.6, 6.2, **87.9**). We find that GPDNS and OpenDNS both resolve to the corresponding regions, Los Angeles and Tokyo, while AliDNS resolves the queries to Singapore. From the table, we see that AliDNS' ECS option has no significant performance improvement outside of China, possibly because AliDNS uses ECS to improve CDN performance in China, while in other regions it still uses Anycast to assign the nearest DNS cluster to the client. So, due to the different coverage of Ali's infrastructure, AliDNS assigned the probes to the more distant DNS clusters by Anycast. Then, Akamai CDN maps these probes to suboptimal servers seeing only recursive resolver's IP.

4) Multi-probes: The latency of accessing the same website from different probes varies. This reflects the difference in the density of Akamai CDN infrastructure deployment in different regions, latency in Sydney, Los Angeles, Frankfurt, and Tokyo being significantly lower than that in Beijing, Nanjing, Moscow, and São Paulo. It indicates that the former regions have higher CDN infrastructure density and better network performance overall [2].

VI. Management Recommendations

First, operators of a closed resolver should check the configuration to see if it mistakenly provides DNS service for users outside the subnet. Next, operators of an open resolver should update or patch its DNS software according to official guidelines (e.g. BIND community [30]) to avoid non-compliant behavior. If a resolver is ECS-enabled, operators must limit outbound ECS-queries' SOURCE PREFIX-LENGTH to no more than 24/56. Also, they should refer to RFC7871 [4] for detailed definition of RCODE in DNS response to ensure consistency of behavior. Lastly, PDNS and CDN providers could build a measurement platform like Microsoft's Odin [31] to collect first-hand performance data and improve the ECS-CDN user mapping policy mitigating privacy issue.

VII. Conclusion

In this paper, we mainly focus on the behavior of ECS-enabled resolvers in IPv6. We discovered 14,599 IPv6 open resolvers. By analyzing DNS queries on both the client-side and the server-side, we found that 96% of ECS options had been changed due to the "ECS rewrite" policy implemented by intermediate resolvers, which goes against the original purpose of ECS. To discuss the tradeoff between ECS privacy and CDN performance improvement, we measured ECS-enabled CDN performance and found that there is no significant positive correlation between the prefix length and performance benefit. The client cannot get the corresponding improvement in return for IP privacy sacrifice and in some cases, ECS can even downgrade the performance. We hope to inform both users and service providers about the unnecessary ECS privacy loss.

VIII. Acknowledgments

This work is supported by the Beijing Natural Science Foundation under Grant No.4222026, National Key Research and Development Program of China under Grant No.2018YFB1800200, and Tsinghua University Initiative Scientific Research Program under Grant No.2021Z11GHX010. Jiahai Yang is the corresponding author.

References

[1] F. Chen, R. K. Sitaraman, and M. Torres, "End-user mapping: Next generation request routing for content delivery," in *SIGCOMM '15*, 2015.
[2] M. A. Sánchez, J. S. Otto, Z. S. Bischof, D. R. Choffnes, F. E. Bustamante, B. Krishnamurthy, and W. Willinger, "Dasu: Pushing experiments to the internet's edge," in *Proc. of NSDI '13*, 2013.
[3] J. S. Otto, M. A. Sánchez, J. P. Rula, and F. E. Bustamante, "Content delivery and the natural evolution of dns: Remote dns trends, performance issues and alternative solutions," in *Proc. of IMC '12*, 2012.
[4] C. Contavalli, W. van der Gaast, D. C. Lawrence, and W. A. Kumari, "Client Subnet in DNS Queries," RFC 7871, May 2016. [Online]. Available: https://rfc-__editor.org/rfc/rfc7871.txt
[5] A. Kountouras, P. Kintis, A. Avgetidis, T. Papastergiou, C. Lever, M. Polychronakis, and M. Antonakakis, "Understanding the Growth and Security Considerations of ECS," in *Proc. of 2021 NDSS*, 2021.
[6] R. Al-Dalky, M. Rabinovich, and K. Schomp, "A Look at the ECS Behavior of DNS Resolvers," in *Proc. of IMC '19*, 2019.
[7] M. Calder, X. Fan, and L. Zhu, "A Cloud Provider's View of EDNS Client-Subnet Adoption," in *Proc. of TMA '19*, 2019.
[8] P. Kintis, Y. Nadji, D. Dagon, M. Farrell, and M. Antonakakis, "Understanding the Privacy Implications of ECS," in *Proc. of DIMVA '16*, 2016, vol. 9721.
[9] E. Carisimo, C. Selmo, J. I. Alvarez-Hamelin, and A. Dhamdhere, "Studying the evolution of content providers in IPv4 and IPv6 internet cores," *Computer Communications*, vol. 145, 2019.
[10] Y.-C. Chiu, B. Schlinker, A. B. Radhakrishnan, E. Katz-Bassett, and R. Govindan, "Are we one hop away from a better internet?" in *Proc. of IMC '15*, 2015.
[11] M. Nikkhah, R. Guérin, Y. Lee, and R. Woundy, "Assessing IPv6 through web access a measurement study and its findings," in *Proc. of CoNEXT '11*, 2011.
[12] K. Fujiwara, A. Sato, and K. Yoshida, "DNS traffic analysis: Issues of IPv6 and CDN," in *Proc. of SAINT '12*, 2012.
[13] V. Bajpai and J. Schönwälder, "Ipv4 versus ipv6 - who connects faster?" in *Proc. of IFIP Networking '15*, 2015.
[14] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan, "Mapping the expansion of Google's serving infrastructure," in *Proc. of IMC '13*, 2013.
[15] W. B. de Vries, R. Van Rijswijk-Deij, P.-T. de Boer, and A. Pras, "Passive Observations of a Large DNS Service: 2.5 Years in the Life of Google," in *Proc. of TMA'18*, 2018.
[16] J. Park, R. Jang, M. Mohaisen, and D. Mohaisen, "A large-scale behavioral analysis of the open dns resolvers on the internet," *IEEE/ACM Transactions on Networking*, vol. 30, no. 1, 2022.
[17] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by internet-wide scanning," in *Proc. of CCS '15*, New York, NY, USA, 2015.
[18] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast internet-wide scanning and its security applications," in *Proc. of USENIX Security '13*, Washington, D.C., Aug. 2013.
[19] "Theshadowserver foundation: Dns scanning project," Website, 2022, https://scan.shadowserver.org/dns/.
[20] L. Hendriks, R. de Oliveira Schmidt, R. van Rijswijk-Deij, and A. Pras, "On the Potential of IPv6 Open Resolvers for DDoS Attacks," in *PAM '17*, 2017, vol. 10176.
[21] A. Hounsel, K. Borgolte, P. Schmitt, J. Holland, and N. Feamster, "Comparing the Effects of DNS, DoT, and DoH on Web Performance," in *Proc. of WWW '20*, 2020.
[22] G. Song, L. He, Z. Wang, J. Yang, T. Jin, J. Liu, and G. Li, "Towards the construction of global IPv6 hitlist and efficient probing of IPv6 address space," in *Proc. of IWQoS '20*, 2020.
[23] A. Murdock, F. Li, P. Bramsen, Z. Durumeric, and V. Paxson, "Target generation for internet-wide IPv6 scanning," in *Proc. of IMC '17*, 2017.
[24] Z. Liu, Y. Xiong, X. Liu, W. Xie, and P. Zhu, "6Tree: Efficient dynamic discovery of active addresses in the IPv6 address space," *Computer Networks*, vol. 155, 2019.
[25] G. Song, J. Yang, Z. Wang, L. He, J. Lin, L. Pan, C. Duan, and X. Quan, "Det: Enabling efficient probing of ipv6 active addresses," *IEEE/ACM Transactions on Networking*, 2022.
[26] G. Song, J. Yang, L. He, Z. Wang, G. Li, C. Duan, Y. Liu, and Z. Sun, "AddrMiner: A comprehensive global active IPv6 address discovery system," in *Proc. of USENIX ATC '22*, 2022.
[27] "Maxminf geolite2 free geolocation data," Website, 2022, https://dev.maxmind.com/geoip/geolite2-free-geolocation-data.
[28] S. Hao, R. Liu, Z. Weng, D. Chang, C. Bao, and X. Li, "Addressless: A new internet server model to prevent network scanning," *Plos one*, vol. 16, no. 2, 2021.
[29] S. J. Saidi, O. Gasser, and G. Smaragdakis, "One bad apple can spoil your IPv6 privacy," *SIGCOMM Comput. Commun. Rev.*, vol. 52, no. 2, 2022.
[30] "BIND community," Website, 2022, https://kb.isc.org/docs/aa-01310.
[31] M. Calder, R. Gao, M. Schröder, R. Stewart, J. Padhye, R. Mahajan, G. Ananthanarayanan, and E. Katz-Bassett, "Odin: Microsoft's scalable Fault-Tolerant CDN measurement system," in *Proc. of NSDI '18*, 2018.