Passive Observations of a Large DNS Service: 2.5 Years in the Life of Google

Wouter B. de Vries, Roland van Rijswijk-Deij, Pieter-Tjerk de Boer, and Aiko Pras

***Abstract*—In 2009 Google launched its Public DNS service, which has since become the largest DNS service in existence. A common problem with public resolvers is that Content Delivery Networks (CDNs) struggle to map end user origin. The EDNS Client Subnet (ECS) extension allows resolvers to reveal part of a client's IP to authoritative name servers, helping CDNs pinpoint client origin. A side effect of ECS is that authoritative name server operators learn where in its network the public resolver handles queries. We leverage this side effect to study Google Public DNS (GPDNS). We perform a longitudinal analysis over data covering 2.5 years and 3.7 billion queries. Our study focuses on three aspects. First, we show that while GPDNS has PoPs in many countries, traffic is frequently routed out of country. This can reduce performance, and expose DNS requests to state level surveillance. We also show that end users are often served by a suboptimal PoP. Second, we show that end users switch to GPDNS *en masse* when their ISP resolver is unresponsive, and do not switch back. Finally, we also find that many e-mail providers configure GPDNS as resolver on their servers, causing serious privacy concerns due to information leakage.**

***Index Terms*—Computer networks, resilience, performance, privacy, domain name system, network topology.**

I. Introduction

THE DOMAIN Name System (DNS) is an important part of what the Internet is today. It *resolves* domain names to IP addresses. The DNS is a hierarchical system where different *name servers* are responsible for different parts of a domain name. In order to resolve a domain name, a so-called recursive resolver queries each name server responsible for part of a domain in turn, until it has the final answer. A customer of an Internet Service Provider (ISP), typically uses a recursive resolver that is provided by the ISP, and that is usually automatically configured. While most ISPs provide their own recursive resolver for their customers, it is usually not mandatory to use these. Customers can either run their own resolver, or use a third party resolver. Examples of organizations offering such third-party services include OpenDNS, Quad9 and Google. There are many reasons why an end-user might use a different resolver than the one operated by their ISP, such as stability, performance, privacy or to circumvent censorship.

In this paper we focus on one particular public resolver, Google Public DNS (GPDNS), which was launched in 2009. Since its inception, GPDNS has grown to be the largest public resolver in existence, serving hundreds of billions of requests per day [1]. While GPDNS uses only a few public IP addresses, its servers have a global presence. Google uses a technique called *anycast* to ensure traffic to GPDNS is routed to a nearby Point-of-Presence (PoP). This reduces latency for clients.

Now while services like GPDNS have a global presence, they typically do not have PoPs in every country. In fact, at the time of writing, GPDNS had 21 active locations on 5 continents. It turns out that this poses challenges for Content Delivery Networks. CDNs frequently rely on the geo-location of the IP address of recursive resolvers as a proxy for the location of end customers. This information is then used to route requests to content caches near the end customer and to serve local content. The underlying assumption is that DNS queries are typically handled by a resolver 'near' the end customer, e.g., their ISP's resolver. If, however, a public resolver, such as GPDNS is used, this assumption breaks down, as requests appear to come from the PoP that handled a user request.

To address this problem an extension to the DNS called *EDNS0 Client Subnet* (ECS) [2] was introduced. This extension allows recursive resolvers to include part of the IP address of the client that sent a query in requests to authoritative name servers. This can assist CDNs in more accurately determining where clients using public resolvers come from.

Interestingly, the use of ECS by name servers has unintended side effects. By sending ECS-enabled queries to CDNs that support the extension, it becomes possible to study the geographic distribution of their services, and how clients are mapped to certain services, as a number of existing studies have shown [3]–[7]. ECS, however, can also be used to examine how a public resolver works and is used. In this paper, we are the first to study a large scale public DNS resolver (GPDNS) over a 2.5-year period using passive observations of ECS data in DNS queries collected at a major authoritative name server.

The main **contributions** of our work are as follows:

- We show that while anycast routing is generally considered relatively stable [8], performance of Google's anycast network varies over time. Additionally, we show that traffic is frequently routed to out-of-country

Manuscript received March 19, 2019; revised June 12, 2019; accepted July 27, 2019. Date of publication August 19, 2019; date of current version March 11, 2020. Work at the University of Twente was supported by SURFnet Research on Networks. The associate editor coordinating the review of this article and approving it for publication was M. Mellia. *(Corresponding author: Wouter B. de Vries.)*

W. B. de Vries, P.-T. de Boer, and A. Pras are with the Design and Analysis of Communication Systems, University of Twente, 7522 NH Enschede, The Netherlands (e-mail: w.b.devries@utwente.nl).

R. van Rijswijk-Deij is with the Design and Analysis of Communication Systems, University of Twente, 7522 NH Enschede, The Netherlands, and also with NLnet Labs, 1098 XH Amsterdam, The Netherlands.

Digital Object Identifier 10.1109/TNSM.2019.2936031

1932-4537 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on November 02,2020 at 10:26:19 UTC from IEEE Xplore. Restrictions apply.

Points-of-Presence (PoPs), even if a local, in-country PoP is available. This potentially exposes DNS traffic to state-level surveillance.
- We show that, based on geolocation of IP addresses, there is often a PoP available that is closer, in terms of physical distance, to the end-user, than the one that is being used.
- We show that end-users switch away from their ISP's resolver if it is underperforming, and more importantly, that these users will not switch back.
- We show that surprisingly large numbers of SMTP servers are configured to perform lookups through GPDNS. This is a potential privacy leak, as it allows the public resolver and any of the authoritative name servers involved in DNS lookups to infer that there is likely communication between two parties. For verification, we validate that a number of common SMTP daemons perform DNS lookups in their default configurations.
- We make our full dataset covering 2.5 years and 3.7 billion queries available as open data to the research community at https://traces.simpleweb.org.

This work was originally published in the Network Traffic Measurement and Analysis Conference 2018 [9]. Here we have extended it with an analysis based on the geolocation of end-users and validated that SMTP daemons indeed perform DNS lookups. We have also included an extensive discussion on how other operators of large public DNS resolvers treat EDNS Client Subnet and we provide guidelines for operators on the deployment of ECS.

The rest of this paper is organized as follows: Section II provides background information. Then, Section III describes our methodology, including how data was collected. Next, Section IV contains data analysis and results. Section V describes related work. In Section VI we discuss some of the privacy implications of ECS. Finally, Section VII provides conclusions and an outlook on future work.

II. Background

A. EDNS and EDNS Client Subnet

The original DNS protocol [10], [11] puts limits on both the size of DNS responses (512 bytes in a UDP datagram) and what options and flags a DNS message can have. Many modern applications of the DNS have requirements that exceed the limits of the original protocol. For this reason, the Extension Mechanisms for DNS (EDNS0) [12] were introduced. EDNS0 uses a special DNS pseudo-record in the additional section of a DNS query or response. This so-called `OPT` record specifies EDNS parameters (e.g., the maximum message size), can be used to specify additional flags (e.g., DNSSEC flags) and can be used to specify new DNS options. The latter, the options, are encoded in the form of *<tag, value>* pairs and can, for example, be used to convey metadata about a DNS message.

Many CDNs and other applications make use of the IP address from which queries are made to their authoritative DNS servers. This IP address is used as a proxy for the location of an end customer. It is used to perform a so-called Geo IP lookup, to determine (roughly) where a customer is coming from, and is used to make decisions about, e.g., which

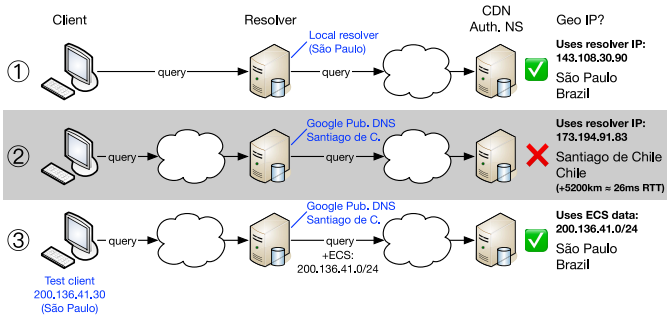


Fig. 1. Explanation of EDNS0 Client Subnet.

content to serve or whether or not to allow access to certain content. With the advent of public DNS resolvers, such as Google Public DNS, this model of identifying where clients are coming from runs into problems. The reason for this is that authoritative name servers for the CDN will now see queries as coming from GPDNS. Even if Geo IP databases are accurate enough to identify the country in which the GPDNS servers that handled the request are located, this may not provide the information to pinpoint the origin region of a request with sufficient accuracy.

To remedy this, the EDNS Client Subnet (ECS) option [2] was introduced. This option can be used by DNS resolvers to provide information about where a query originated. To do this, a DNS resolver includes two fields in the ECS option: the *IP prefix* from which the query originated and a *source prefix length* field that specifies the size of the provided prefix (e.g., `/24`). For privacy reasons, DNS resolvers typically limit how specific the scope is that they send in a request. The ECS standard [2] recommends using a maximum scope of `/24` for IPv4 and `/56` for IPv6. An authoritative name server can then use this information to decide which region-specific response to return to a query. To ensure that responses from the authoritative name servers are only cached for users in the correct prefix, the authoritative name server also includes its own *scope prefix length* field in the response. This field must be used by the DNS resolver when caching the response. For more information on what DNS servers should do in case of prefix-length mismatches, we refer to the RFC [2].

Figure 1 shows an example of 1) a local resolver, 2) a public resolver without ECS and 3) a public resolver with ECS. The figure shows the potential impact of not using ECS for a public resolver. The example is based on a client we control, located in São Paulo, Brazil. Without ECS, a CDN using Geo IP will assume this client is in Santiago de Chile, 2600km away as the crow flies, adding a potential 26ms to each network round-trip.

III. Methodology

As discussed in the introduction, EDNS Client Subnet provides a unique opportunity to observe the behaviour of large public DNS resolvers. In this section, we outline how we collected our data, and how we will use this to study one particular public DNS resolver operator: Google.

A. Data Collection

1) Collection Point: We used one of the authoritative name servers of SURFnet, the National Research and Education

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on November 02,2020 at 10:26:19 UTC from IEEE Xplore. Restrictions apply.

TABLE I
DISTRIBUTION OF GOOGLE POPS (STATE OF 8 NOVEMBER 2017) AND RECEIVED QUERIES AT OUR AUTHORITATIVE NAME SERVER

Continent	PoPs	Prefixes IPv4	IPv6	# of queries
Asia	4	13	4	391,523,557
Europe	6	19	6	1,800,743,147
North America	8	40	8	1,450,006,164
Oceania	1	3	1	2,633,248
South America	2	3	2	29,143,338
Total	*21*	*78*	*21*	*3,711,406,022*

Network (NREN) in The Netherlands, to passively collect DNS queries from Google Public DNS. Data was collected from the end of June 2015 to the end of December 2017. Only DNS queries that include an ECS option are collected, and for these queries, we record the *origin IP* of the query (i.e., the IP of the Google resolver that sent the query), and the *IP prefix* and *source prefix length* included in the ECS option. In addition to this, we use CAIDA's IP prefix to Autonomous System (AS) dataset [13] to map the ECS IP prefix to an AS and we use the free IP2Location dataset to map the ECS IP prefix to a country, as well as to coordinates.

The SURFnet name server we used is authoritative for approximately 10,000 DNS zones, including a number of popular public suffices[1] such as `.ac.uk`, `.gov.uk` and `.ac.be`. As a result of this, this name server sees a wide spread of queries from all over the Internet and world. Note though, as we discuss in more detail below in Section III-C, we do expect bias in which Google PoPs send traffic to this server, due to resolver-to-authoritative RTT optimisation. Table I shows an overview of the data we collected for this study, broken down per continent from which queries originated.

2) BIND Patch: Google Public DNS automatically detects support for ECS on authoritative name servers. In order to do this, Google regularly sends probing queries that include an ECS option. If an authoritative name server includes an ECS option in the response, this is interpreted as an indication of support. To ensure that Google would detect our name server as ECS-capable, we implemented a patch for the popular BIND DNS implementation. After this patch, BIND will accept the ECS option, and will include an ECS option in the response that mirrors the source address and prefix length in the scope prefix length field. Figure 2 shows the number of ECS-enabled queries per 5 minutes from Google PoPs increasing after we have enabled ECS on our patched name server as resolvers at these PoPs detect support for ECS.

3) Ethical Considerations: As the ECS standard [2] already specifies, there are inherent privacy concerns in the protocol, as a resolver that supports ECS includes a (sometimes significant) part of the client's IP address in queries. We are interested in how clients are routed to GPDNS and in general terms how GPDNS is used at the network operator level. Therefore, to protect user privacy, we take two measures: 1) we do not store query names, but we do make a hash of the query name

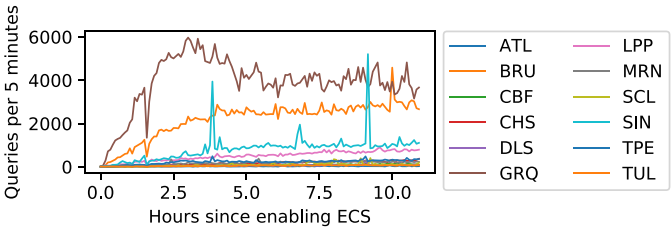


Fig. 2. Ramp-up of GPDNS detecting and enabling ECS.

available in the open data and 2) we aggregate ECS client IP prefixes at the AS level when analysing the data, with two exceptions; if we believe the prefix contains servers that use GPDNS (rather than individuals), we analyse if certain types of hosts (specifically, e-mail servers) exist in these prefixes (Section IV-E). We also use the specific ECS client IP prefix to perform the geo-location lookup (Section IV-C), however we only present aggregated results.

A secondary concern is the effect on query and cache efficiency. While we implement ECS on the authoritative name server where we collect data, we do not differentiate DNS responses based on ECS. Since DNS resolvers that implement ECS should cache responses based on the ECS information, this may impact caching efficiency. Consequently, GPDNS may have to cache responses from our patched name server for every client prefix they send in ECS-enabled queries. The standard [2], however, provides clear guidelines for resolver implementers to avoid cache pollution. In addition to this, the impact of us implementing ECS will only have a limited impact, as the other authoritative name servers for domains for which our patched name server is authoritative do not implement ECS. In many cases this means only one in four queries sent from Google will result in an ECS-enabled response (of course depending on how Google's resolvers distribute queries over the set of authoritative name servers for a domain).

B. Resolver IP to Point-of-Presence Mapping

While end-users query Google Public DNS via the front-end IP addresses `8.8.8.8` and `8.8.4.4` or their IPv6 counterparts, an individual Google DNS resolver then uses different IP addresses to actually resolve the query. The IP-prefixes that are used are published and are frequently updated [14]. Google identifies their PoPs using the three-letter IATA code of the nearest airport. During the first part of the measurement period we did not store the mapping of prefixes to Google PoPs, but we recovered it using The Wayback Machine (TWM).[2] Specifically, we collected 25 mappings between the start of our measurement period and the 3rd of August 2017 through TWM. From that point onward, we collected the mapping on a daily basis directly from Google through a DNS query, as described in Google's FAQ [14].

Figure 3 shows the number of prefixes associated with each PoP and how this varies over time. As the figure shows, several new PoPs were added over the period covered by our dataset, for example, approximately halfway through 2017 the Sydney

[1]For an explanation of public suffices, see https://publicsuffix.org.

[2]https://archive.org/

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on November 02,2020 at 10:26:19 UTC from IEEE Xplore. Restrictions apply.

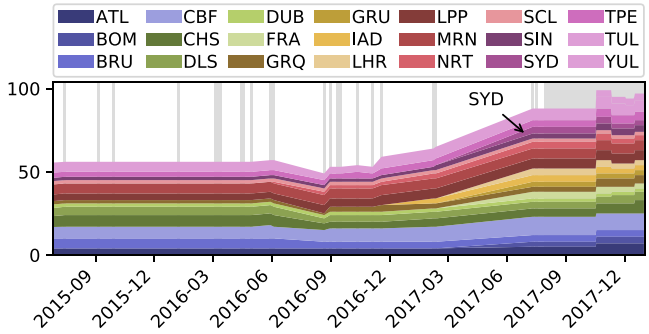


Fig. 3. Number of prefixes associated with each of Google Public DNS PoPs. The gray background indicates where we have data.

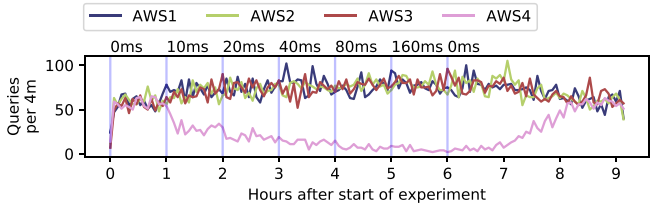


Fig. 4. Division of queries from Google over 4 name servers. Every vertical line indicates a change in latency of AWS4 (e.g., +10ms, +20ms).

PoP was added. We also observe, over the total duration, 4 instances where a prefix was reassigned to a different PoP. It is likely that due to the significant delay in mappings which we obtained via TWM we mismapped a portion of the traffic when such a reassignment occured. However, considering the large amount of total prefixes, the vast majority of which were not reassigned at any point, the impact of this is likely to be small. For our study, we use the prefix-to-PoP mapping we recorded to map queries to Google PoPs based on the prefix in which the source IP of a query is contained. In total we were unable to map 26,548,020 queries to their corresponding PoP (0.7% of all queries in our dataset).

C. Distribution of Queries Over Authoritative Name Servers

As we discussed in Section III-A, we collect data on a single authoritative name server. The median number of name servers configured per DNS zone on that name server, however, is 4.0, which means that not all queries from Google for domains hosted on that name server will be sent to that particular name server. Typical resolver implementations will distribute queries over all authoritative name servers for a domain, usually favoring servers with shorter RTTs [15], [16].

While it is infeasible to exhaustively determine RTTs from all Google PoPs to all authoritative name servers for domains for which our test server is also authoritative, we did want to get an idea how GPDNS resolvers factor in RTT when selecting an authoritative name server. Therefore, we conducted an experiment in which we set up four authoritative name servers for a single domain, each with a different public IPv4 address, but hosted on a single machine. This ensures uniform performance characteristics from an external viewpoint. We then measured the distribution of queries by GPDNS over several hours, where we artificially increase the RTT for one of the name servers every hour.

Figure 4 shows that a server which has an increased latency compared to the others, receives fewer queries. The ratio appears to be constant given a certain RTT distribution. In other words, as long as the latency remains constant, so does the distribution of queries over the authoritative name servers.

Based on this experiment, it is clear that by counting queries to our single vantage point, we cannot make claims about the total number of queries from GPDNS for domains for which our vantage point is authoritative. Since, however, the distribution of queries appears directly linked to RTT, and given that our vantage point is well-connected (a single hop away from major IXPs, including AMS-IX and LINX), we can measure trends in traffic coming from GPDNS over time. Improvements in RTT towards the other authoritatives could cause us to miss a shift in the trend, however, we deem it unlikely that this systematically occurs.

IV. Results

A. Query Distribution

The front-end IP address of GPDNS is `8.8.8.8`, however, as the service is anycasted, this means that an end-user can potentially reach any of the PoPs, as determined by BGP routing [17]. In this section, we look at the actual distribution of queries over the PoPs that are available. Figure 5 shows the relative distribution of the traffic over PoPs. The three letter acronyms in the legend indicate IATA airport codes. Since the authoritative DNS server where we captured the traffic is located in The Netherlands, and is authoritative for mostly Dutch domain names, we expect the PoPs near or in The Netherlands to handle most of the load.

Prior to October 2015 most traffic was handled in the BRU PoP (Brussels, Belgium). Then, when GRQ (Groningen, The Netherlands) was brought online, marked by (1) in the plot, there was a major shift. The traffic to the BRU PoP was significantly reduced at the same time, with all the other PoPs showing a reasonably constant amount of traffic. This is likely due to the fact that the majority of users in The Netherlands have a shorter path to GRQ than to BRU. In the period marked by (2), the situation temporarily reverted to its previous state as, for an unknown reason, the GRQ PoP was deactivated. We see that, as with the previous change, the amount of traffic handled by BRU PoP increases significantly.

After the GRQ PoP was re-enabled, in the period marked by (3), the distribution of traffic is largely stable with no significant changes in almost a year, other than a slow increase in the share of traffic from PoPs in Asia (TPE and SIN).

Figure 6 shows the distribution of the top-10 query types. The fraction of `PTR` records is surprisingly high, almost equivalent to the fraction of `A` queries. This can be explained by the fact that the authoritative name server on which we collected queries, is also responsible for over 2,100 reverse DNS zones (including many at the `/16` level in the IPv4 hierarchy). We suspect that most of these `PTR` queries are sent by mail servers, and examine this in more detail in Section IV-E.

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on November 02,2020 at 10:26:19 UTC from IEEE Xplore. Restrictions apply.

TABLE II
QUERIES ANSWERED OUT OF COUNTRY, WHILE AN IN-COUNTRY RESOLVER EXISTS. R = RESOLVER COUNTRY, C = CLIENT COUNTRY

	A – before (4) 1 month before 2016-04-29			**B** – during (4) 1 month after 2016-04-29			**C** – after (4), before (5) 1 month after 2016-09-07			**D** – during (5) 1 month before 2017-07-10			**E** – after (5) 1 month after 2017-07-10		
#	R	C	Count	R	C	Count	R	C	Count	R	C	Count	R	C	Count
1	US	NL	55,258,986	NL	BE	825,902	BE	NL	33,545,772	BE	NL	8,246,080	BE	GB	18,155,553
2	US	SG	1,633,927	BE	IE	733,032	US	NL	14,007,550	US	SG	2,179,035	BE	DE	4,011,011
3	US	BE	895,542	BE	NL	394,837	JP	US	966,666	BE	IE	1,035,833	US	BR	3,092,432
4	US	IE	709,332	TW	SG	184,409	BE	IE	922,468	BE	GB	613,909	SG	IN	2,573,892
5	US	TW	520,368	TW	US	173,502	TW	JP	822,037	TW	JP	476,647	BE	NL	2,509,740

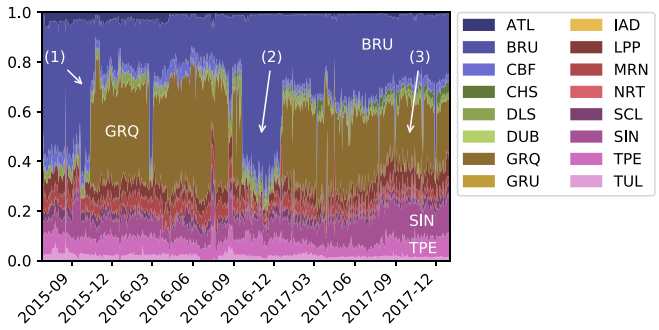


Fig. 5. Ratio of DNS queries per PoP resolved via Google Public DNS.

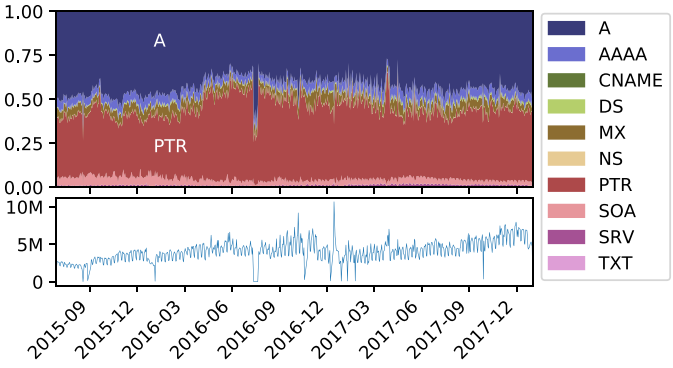


Fig. 6. Ratio of DNS query types and absolute number of queries per day.

Lastly, we see no significant increase in AAAA queries (IPv6) over the full measurement period, indicating a surprising lack of uptake of IPv6 among clients of GPDNS.

B. Out-of-Country Answers

Earlier work [18] showed that for public DNS services such as GPDNS, the distance between a resolver and a client varies greatly. This can lead to performance penalties if the anycast PoP serving the client is geographically remote from the client. Another question to ask in this context is: *are clients in a certain country always served by a PoP in that country?* This is especially relevant in an age of ubiquitous surveillance by intelligence services, because if traffic is routed through or to another country, this exposes that traffic to potential prying eyes. With an anycasted service such as GPDNS, one might expect that if there is a PoP available in country X, while making a request from that same country, that queries are answered from that PoP. However, since BGP has no notion of physical proximity or country borders, this is not necessarily the case.

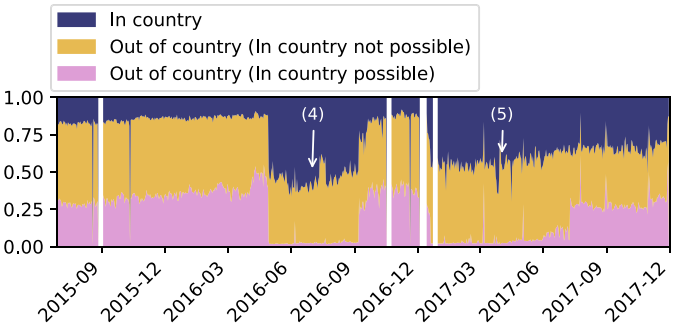


Fig. 7. Ratio of out of country answers.

Figure 7 shows the fraction of queries answered from outside the country of the end user, while a resolver inside was available. This is generally the case for 30% of queries. We observe two deviations from the overall trend, marked by (4) and (5), from late April 2016 to early September 2016 and from December 2016 to July 2017 respectively.

In Table II we show the top 5 queries that are answered out of country, while an in-country resolver was available. Queries are grouped by resolver country and client country. We compiled this top 5 over five time periods, A through E, each representing a month of data, either before or after the beginning or end of one of the deviation periods (4) and (5), as labeled at the top of the table.

In period A, before the start of event (4), the top 5 are all answered from the U.S., while at the same time the Brussels datacenter is clearly active (see Figure 5). Then, in period B, during event (1), the total number of OOC queries drops dramatically, and the ones that do still occur are significantly closer in terms of geographic distance. The relative amount of OOC queries returns to its previous level in period C, between events (4) and (5), although the distribution has changed significantly, arguably for the better (i.e., less geographical distance between resolver and client).

The changes that occur at the event marked as 5 are less dramatic, the number of clients who receive answers from a resolver in Belgium while located in Great Britain does increase significantly. The fact that these countries are relatively close to each other means that the performance impact is limited, although there is still a privacy impact. The situation for clients located in The Netherlands improved, as the number of queries served from Belgium decreased.

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on November 02,2020 at 10:26:19 UTC from IEEE Xplore. Restrictions apply.

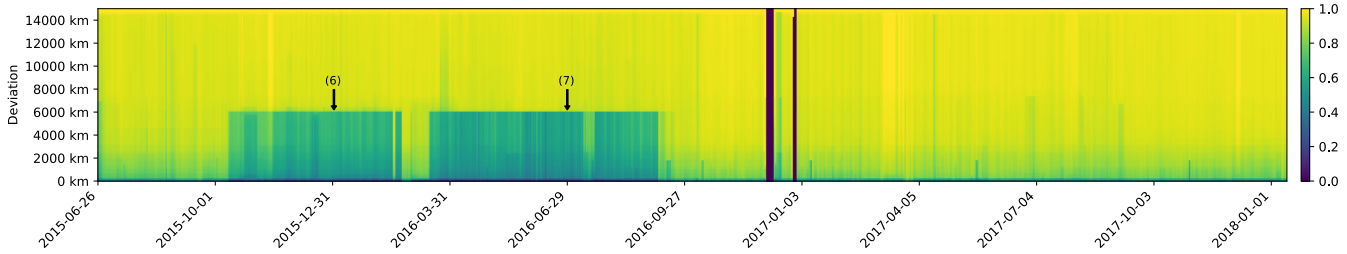


Fig. 8. CDF time-series colour plot of Deviation between Optimal and Actual PoP.

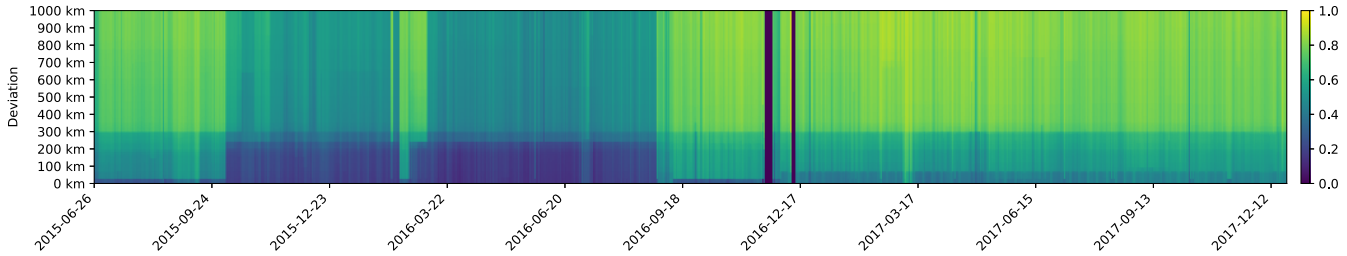
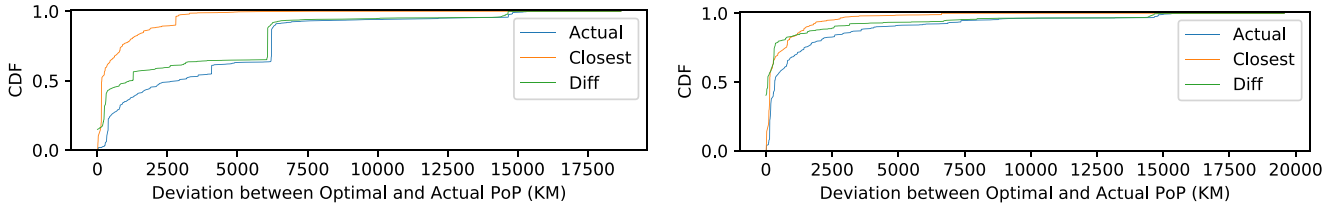


Fig. 9. Zoomed in on the first 1000 km of Figure 8.



(a) 2016-04-29

(b) 2017-07-10

Fig. 10. Deviation between optimal and actual PoP.

C. Physical Distance Between End-User and PoP

While traffic being routed in or out of the user's country is particularly relevant for privacy, in terms of performance the physical distance between the end-user and the PoP is more important. Consider that an end-user might reside close to a country's border, and as such a PoP in the neighboring country is potentially physically closer.

In this section we investigate the physical distance between the end-user, the PoP that end-user *was served by*, and the closest PoP that end-user *could have been served by*. We calculate the distance between the end user and each possible PoP, using the Haversine formula. This formula calculates the great circle distance between two coordinates, and it works under the assumption that the earth is a perfectly round sphere. We subtract the distance between the end-user and the used PoP and between the end-user and the PoP that had the shortest distance. We refer to this value as the deviation between optimal and actual PoP.

To perform this analysis we have used the public geolocation database IP2Location. There have been many studies [19]–[22] that have analyzed the accuracy of such databases. In general, the accuracy on the country-level is considered to be relatively high, however on the city and coordinate level there is a considerable error margin, generally in the order of hundreds, but sometimes up to thousands of kilometers. In our analysis we are interested in large scale changes that happen over time, where in our opinion the impact of this error margin is limited. Special care should however be taken when interpreting the graphs based on these results.

Considering Table II, we expect CDFs of the deviation between the actual and optimal PoP on 2016-04-29 and on 2017-07-10 to show a significant difference, considering that the distance between The Netherlands and the United States is much larger than between Belgium and The Netherlands. As shown in Figure 10a and Figure 10b, this is indeed the case.

To show the development of the deviation between optimal and actual PoP, we plotted this deviation as a colour plot. Figure 8 shows the result; in essence, each vertical line in the plot can be viewed as if looking down on a CDF as plotted in, e.g., Figure 10a or Figure 10b. Because the maximum deviation can be quite large, we also include Figure 9, which provides a view of the first 1000 km for clarity.

Interestingly, these do not (completely) align with the previous analyses (e.g., Figure 4 or Figure 7). The reason is that queries being answered in-country or out-of-country does not necessarily involve a big change in distance. Likewise, the GRQ (Groningen, The Netherlands) PoP going down saw most clients moving to BRU (Brussels, Belgium), which is a relatively short distance. The large blocks marked by (6) and (7) are caused by a small number of *heavy hitters*, doing reverse lookups, originating in The Netherlands, now reaching the IAD PoP in Washington, DC.

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on November 02,2020 at 10:26:19 UTC from IEEE Xplore. Restrictions apply.

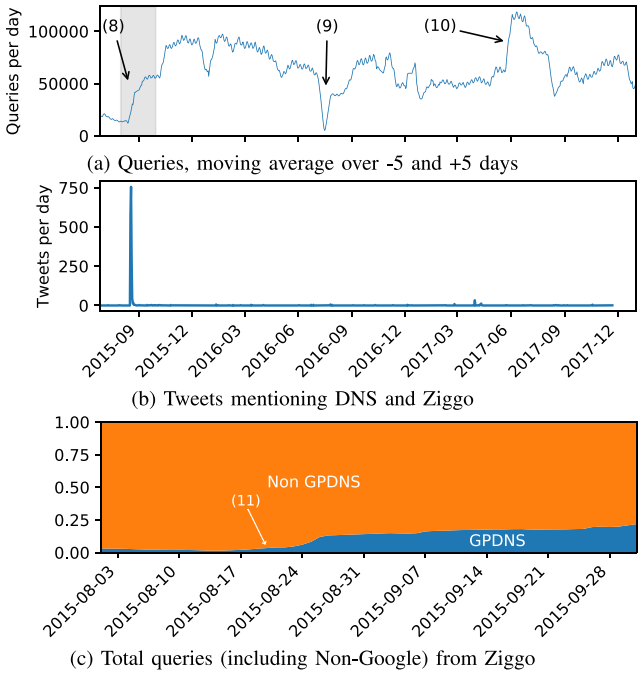
D. Events Leading to Google DNS Adoption

There are various reasons why an end-user might switch from their ISP's DNS resolvers to GPDNS, such as performance, security (in the form of DNSSEC) or resilience. In this section we take a closer look at an event that lead to a drastic increase in the use of GPDNS for a particular ISP.

The example we analyze in this section involves Ziggo, one of the largest ISPs in The Netherlands. Around August 2015 the DNS servers of this ISP suffered a Distributed Denial-of-Service (DDoS) attack, causing serious service disruption for its customers. Major national news services (e.g., [23]) reported that configuring a third-party DNS service could help users. We asked ourselves: *does an attack on a major provider and subsequent media coverage suggesting the use of, e.g., GPDNS lead to increased adoption of GPDNS?*

As it turns out, this is indeed the case. In Figure 11a we show the number of queries that originate from Ziggo's AS (9143) per day, using a moving average of -5 to $+5$ days. The uptake around the date of the attack can be clearly seen, marked by (8). The event indicated by (9) indicates a gap in our data collection, while (10) is caused by a single `/24` temporarily issuing tens of thousands of requests. Figure 11b shows the number of tweets that used the words DNS and Ziggo over the entire measurement period. The tweets were collected using Twitter's Web API. The clear spike coincides with the DDoS attack, and marks the beginning of the increase in GPDNS use.

Another takeaway from Figure 11a is that uptake remains high, even after the attack has passed and Ziggo's DNS servers return to normal operation. This shows that, while DNS is a fairly technical concept, in case of a major outage such as this, people will switch away from their ISP's DNS servers, and more importantly: never switch back. How dramatic this effect really is, is illustrated by Figure 11c. This graph (which zooms in on the two-month period around the attack) shows what fraction of queries to our vantage point arrives directly from Ziggo's AS, and what fraction arrives through Google. The time of the attack is marked by (11).



(a) Queries, moving average over -5 and +5 days

(b) Tweets mentioning DNS and Ziggo

(c) Total queries (including Non-Google) from Ziggo

Fig. 11. Ziggo (AS9143).

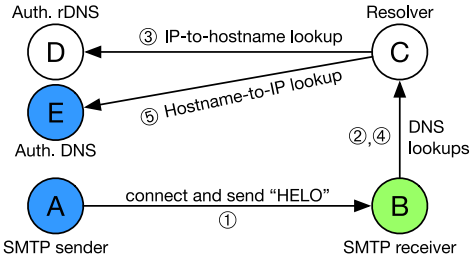


Fig. 12. SMTP DNS Lookup process.

E. SMTP, Google, and EDNS0 Client Subnet

As we hinted at in Section IV-A, the distribution of query types shows a large percentage of `PTR` queries. Pointer (`PTR`) records are used to define reverse DNS names for IP addresses. For example, given IP address `10.0.0.1` there might be a `PTR` entry for `1.0.0.10.in-addr.arpa` which points to a hostname, for example `my.host.com`. For a complete configuration there should then also be an `A` record for `my.host.com` that resolves to `10.0.0.1`. Using this methodology an IP address can be converted to its corresponding hostname and vice versa.

Reverse DNS (rDNS) is commonly used by mail servers to authenticate a sending host. Upon an incoming connection, an SMTP server typically performs a lookup of the reverse hostname of the connecting IP. If a hostname is returned, it will subsequently attempt to resolve this hostname back to the corresponding IP address. In typical configurations SMTP servers may not accept e-mail from other SMTP servers if they do not have a correctly configured reverse hostname [24].

The fact that we see large numbers of `PTR` queries coming via Google, suggests that there may be SMTP servers that have configured GPDNS as their resolver. This is a potential privacy issue, as an SMTP server then discloses to Google which servers are connecting to it to deliver e-mail. Even worse, though, GPDNS may in turn disclose this information to authoritative name servers through the ECS extension. While intuitively one might think this is not a serious issue, consider that these queries are not only sent to the authoritative name server for a domain or reverse DNS zone, but also to the name servers of their parent domain. Concretely, this exposes information about e-mail traffic to TLD and Root operators, as well as reverse DNS operators higher up the DNS hierarchy.

Figure 12 illustrates this scenario. In step (1) a sending SMTP server A connects to a receiving SMTP server B. In step (2), B then performs a reverse DNS lookup for the IP address of A via resolver C (which could be Google Public DNS). Resolver C resolves the actual IP address to a hostname by contacting the authoritative nameserver D in step (3). Once the reverse hostname is known usually the reverse hostname is again resolved to an IP address, following a similar pattern as

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on November 02,2020 at 10:26:19 UTC from IEEE Xplore. Restrictions apply.

TABLE III
INFORMATION LEAK SCHEMA

	Google DNS at Sender	
Operator of service below can see	About sender	About receiver
Google DNS*	IP (Query source)	IP (DNS record)
Authoritative DNS for Reverse**	n/a	n/a
Intermediate DNS**	n/a	n/a
Authoritative DNS for Forward**	/24 prefix (ECS)	IP (DNS record)
Intermediate DNS**	/24 prefix (ECS)	IP (DNS record)
	Google DNS at Receiver	
Operator of service below can see	About sender	About receiver
Google DNS*	IP (DNS record)	IP (Query source)
Authoritative DNS for Reverse**	IP (DNS record)	/24 prefix (ECS)
Intermediate DNS**	IP (DNS record)	/24 prefix (ECS)
Authoritative DNS for Forward**	IP (DNS record)	/24 prefix (ECS)
Intermediate DNS**	IP (DNS record)	/24 prefix (ECS)

* Unless already in local cache (subject to TTL),
** Unless already cached by Google (Subject to TTL)

TABLE IV
RESULTS OF CONNECTING TO EACH OF THE SMTP SERVERS

	IPs (PTR)	IPs (MX)
Connectable	15.374	42.693
Valid SMTP Status code	9.681	32.391
DNS query in timeframe	6.503	20.107
From Google AS	3.188	11.208
Total unique	**14.204**	
Total DNS queries received	11.076	27.723

TABLE V
SMTP DAEMONS

	Qmail v1.06	Sendmail v8.15.2	Postfix v3.1.9	Exim v4.89
1. Performs reverse lookup on connecting IP	Yes	Yes	No	Yes
2. Performs forward lookup on result of 1	No	Yes	No	Yes
3. Performs lookup on domain of From email address	No	Yes	No	No

before, in steps (4) and (5). If GPDNS is used, it may include ECS information in the queries to name servers D and E (if they support ECS), exposing information about which host is sending mail to this mail server to the authoritative name server operators. This becomes worse if GPDNS does not have sufficient information in its cache to contact D and E directly, and first needs to perform a full DNS recursion, hitting servers further up the DNS hierarchy (TLD, root, ...). If these also support ECS, information also leaks to these parties. Table III describes for different scenarios what information is leaked.

While we cannot be certain that it is in fact an SMTP daemon that performs the lookups as opposed to, for example, the firewall, this makes no difference to the privacy risks.

In order to verify this scenario in practice, we first extracted the subnets responsible for the bulk (80%) of the `PTR` queries, resulting in approximately 2,000 `/24` subnets. We then used a standard scanning tool (nmap) to find systems which had port 25 (most likely SMTP) open. The scan found that a little over 50% of the subnets contain at least one system listening on port 25, for a total of roughly 15k systems.

We connect to each of the systems that have port 25 open, with a timeout of 3 seconds. We immediately transmit our identity in the client initiation phase of the SMTP session. We then read data for 6 seconds, checking for SMTP status codes as specified in [25], or until we receive a `250` message, indicating that our `HELO` message has been accepted. To determine if the SMTP daemon uses Google Public DNS to lookup the reverse hostname of our connecting system, we monitor the incoming DNS queries on system E in Figure 12. If we see an incoming DNS query for our domain between the time of connecting and the time of disconnecting we assume that this DNS query is a result of our connection.

Table IV summarizes our results. Of the approximately 10k SMTP servers that we found that transmitted a valid status (a 3 digit number at the beginning of a line), we saw an incoming DNS query from roughly two thirds, and half of those came through GPDNS. We repeated the experiment without sending an initial `HELO` message from our side, with similar results.

For comparison, we also scanned the top 2,000 `/24`s responsible for `MX` queries. In contrast to `PTR` queries, these are likely to originate from "sending" SMTP servers. We find approximately 43k systems with this scan. Similar to our previous results, approximately two thirds of these perform a DNS query on connection, and half do this via GPDNS.

Summarizing, SMTP servers that (indirectly) use GPDNS as a resolver are common. Worryingly, we find 14,204 SMTP servers that, upon connection, leak our IP address or our prefix to GPDNS and any DNS servers that are hit during recursion.

F. Verification of Lookup Behaviour of SMTP Daemons

To verify that SMTP daemons do indeed perform DNS lookups as part of their default configuration we performed a lab experiment testing four common open source SMTP daemons. These are Qmail, Sendmail, Postfix and Exim, which appear to be in wide spread use across the Internet, although real statistics of their market share are lacking.

We setup a Docker environment with a container for each of the daemons, and an additional container running the BIND DNS daemon. BIND is configured with a reverse `PTR` record for the IP address that we will use to connect to each of the SMTP daemons, as well as a forward record for the hostname.

The containers in which the SMTP daemons run are configured to resolve their DNS queries via the BIND daemon. They also run `tcpdump` to capture any DNS queries going to or coming from this container. Once configured we attempt to send a single e-mail through the SMTP daemon, to a local user (i.e., `root@dns-smtp-${daemon}.localhost`). The resulting DNS lookups are listed in Table V.

Notably, all daemons but Postfix perform a DNS lookup in the standard configuration (as provided by the Debian maintainers). Sendmail and Exim both also perform a forward lookup, and the former also performs a lookup on the domain in the "From" address in the message. We note that Postfix can also trivially be configured to perform these DNS lookups.

V. RELATED WORK

The idea for the EDNS Client Subnet extension was first tabled in 2011, supported by a coalition of parties promoting

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on November 02,2020 at 10:26:19 UTC from IEEE Xplore. Restrictions apply.

a "Faster Internet."[3] Partners in this project include both CDN operators and operators of large public DNS resolvers.

Otto *et al.* were the first to study ECS. In their paper [3], they study the impact of the use of public DNS resolvers on Web CDN performance, and highlight the performance improvement ECS could offer in this context. Furthermore, they study the first preliminary uptake of ECS by CDN operators. In follow-up work [4], Sánchez *et al.* study the performance improvement of CDN Web delivery if ECS is used via Google Public DNS. Streibelt *et al.* use ECS to study the infrastructure of CDNs that support ECS. Their paper [5] shows how ECS can be used to provide insight into CDN server deployments, and CDN server-to-client mappings. They highlight that they can perform such mappings from a single vantage point, by inserting arbitrary prefixes in the scope field of an ECS query, provided that CDN operators do not limit from which sources they are willing to respond to ECS-enabled DNS queries. Additionally, Streibelt *et al.* also study aggregation by ECS-enabled CDNs, showing different strategies where some CDNs return ECS responses with larger scopes (i.e., returning an ECS response for an IPv4 `/16` prefix when a smaller `/24` prefix is specified in the DNS query), whereas others respond with narrower scopes than asked for, going as low as a `/32`. The authors speculate that CDNs that follow the latter practice essentially want to force DNS resolvers to cache the result only for a single client.

Calder *et al.* use ECS for a longitudinal study of Google's service delivery CDN. Their work [6] shows that using ECS they can create a complete mapping of this CDN and can uncover dramatic growth of this CDN over a ten-month period. Fan *et al.* also use ECS to study Google's CDN, but rather than focusing on the CDN infrastructure itself, they study changes in client prefix to CDN front-end mappings over time [7].

Chen *et al.* study the impact of ECS deployment from inside the Akamai CDN [18]. The introduction of ECS at Akamai resulted in a 30% improvement in startup time for connections to the CDN, at the cost of an eight-fold increase in the number of DNS queries to their name servers. Chen *et al.* are the first to show the RTT performance penalty incurred by users due to their DNS requests getting routed to geographically remote public resolvers. In this work we significantly extend on this by using longitudinal data covering 2.5 years, showing, e.g., changes in out-of-country query handling over time.

A common denominator of these related works to date has been that it exclusively focused on using ECS to study service delivery by CDNs or to study how ECS can improve this service delivery. In contrast, in this work, we leverage ECS to study the behavior of and use of a large public DNS provider.

Kintis *et al.* discuss some of the privacy implications of ECS [26]. Their focus is the privacy risks imposed by on-path attackers between the public DNS resolver and authoritative name servers on the Internet. They observe how an on-path attacker can perform selective surveillance on clients of public DNS resolvers. In addition to this, they also show how an attacker can selectively poison a public DNS resolver's cache for specific clients using ECS. In this work, we extend this by showing new privacy risks where the use of public DNS resolvers by SMTP servers to perform DNS resolution leaks information about the IP addresses and domains of hosts sending e-mail to these servers.

Liu *et al.* studied the performance of anycast based on two days of data of the root servers, specifically the letters C, F and K. In their paper [27] they study data obtained directly from the root operators. It is unclear how big the used data set is, however the analyzed time span is relatively short. In our paper we look at similar metrics, however we look at a large recursive resolver instead of the root servers, the former having a larger impact on actual end-user quality of service. Additionally our data set spans a longer time period, which allows us to see larger changes as the anycast network is evolving.

Similar to Liu *et al.*, Castro *et al.* study the root servers, using a partially overlapping data set. They study the usage patterns of the root server across three days, in three different years (2006, 2007 and 2008). The work [28] focuses on characterizing queries that are received at the roots, and to show what changed in those three years.

Finally, Li *et al.* study the performance of IP anycast at Internet scale [29]. In Section IV-C we use a similar methodology as Li *et al.* to determine the deviation in distance between the optimal GPDNS PoP and the one that traffic is actually routed to.

VI. Discussion

The EDNS Client Subnet RFC is already very critical about the potential privacy implications of the use of ECS. In the privacy note included *before* the actual protocol description, the authors write:

> *"If we were just beginning to design this mechanism, and not documenting existing protocol, it is unlikely that we would have done things exactly this way."*

They go on further to suggest that ECS, if supported in DNS resolver software, should be turned off by default. Clearly, as the research in this paper illustrates, Google has not followed this suggestion. On the contrary, Google actively attempts to detect support for ECS on authoritative name servers, and will include the option in queries when an authoritative name server responds with an ECS option to their probing queries. This makes it trivial to solicit ECS information from Google.

This raises the question what other operators of large public resolvers do. Table VI shows five commonly used public resolver operators. As the table shows, there are wildly varying policies. Two operators do not support ECS at all, in both cases explicitly citing privacy as a reason not to support ECS. The three others all support ECS, but all have a different policy on when they send ECS to authoritative name servers. Of the three, the OpenDNS policy of active whitelisting probably makes the best tradeoff between protecting the privacy of users versus maintaining an efficient service for users towards CDN providers that support ECS and ask to be whitelisted. Arguably, in terms of privacy, VeriSign's policy is the least favourable of the three, as *any* authoritative name server operator is likely to receive ECS information (and thus able to

[3]http://www.afasterinternet.com/participants.htm

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on November 02,2020 at 10:26:19 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
EDNS CLIENT SUBNET POLICIES OF PUBLIC RESOLVER OPERATORS

Operator	ECS policy	Scope IPv4	Scope IPv6	Respects Scope-Zero?
Google	Auto-detect	/24	/56	Yes
OpenDNS	Whitelist	/24	/48	No
Quad9	No ECS	n/a	n/a	n/a
Cloudflare	No ECS	n/a	n/a	n/a
VeriSign	Always	/24	/64	Yes

record this), regardless of whether their deployed name server software actually supports ECS.

Table VI also shows the scope prefix each operator includes in queries. All three operators that send ECS information include the smallest prefix recommended in the RFC for IPv4, but there are significant differences in the prefix size included for IPv6, from relatively large (OpenDNS), to a very narrow prefix that may identify individual end users (VeriSign).

The last column in Table VI shows whether those operators that send ECS information allow clients control over whether or not ECS information should be sent to authoritative name servers for their queries. The RFC allows clients to exert control over the ECS behaviour of resolvers by including an ECS option in the query they send to a resolver. If a client includes an ECS option with the scope prefix length set to zero, then resolvers should interpret this as an indication that the client does not want information about their IP to be included in queries to authoritative name servers. As Table VI shows, both Google and VeriSign respect such requests from clients, whereas OpenDNS does not. It is debatable whether this option really gives clients control, as the stub resolvers included in all main stream operating systems do not support this behaviour.

Finally, we reflect on what operators of DNS resolvers should do with respect to ECS. Many open source implementations of DNS resolver software now support ECS. Given the potential privacy impact demonstrated in this paper, and identified by others [26], we argue that the default should be not to enable ECS. Operators of large networks, or of public DNS resolvers, that do wish to enable ECS support to better serve their clients when querying CDNs should practice active whitelisting of authoritative name servers that will receive ECS information, and should also carefully consider what source prefix length to include in the ECS option. While a separate study would be required to recommend specific prefix lengths to support, e.g., correct identification of users at the country or regional level, we note that initial experiments with different prefix sizes for IPv4 against popular Geo IP databases shows that sending a /22 source prefix length allows accurate identification at the country level in up to 95% of cases.

VII. CONCLUSION

There has been much debate about the privacy risks of using public DNS resolvers. The obvious argument is that the operator of such a resolver gets access to extremely privacy-sensitive information in the form of DNS queries. One aspect of privacy in the context of public DNS resolvers remains underexposed. In order for CDNs to be able to make Geo IP-based decisions, many public resolvers use a DNS extension called EDNS0 Client Subnet (ECS), which allows them to reveal part of a client's IP to the CDN. In essence, the need for ECS is an unintended side-effect of the use of public DNS resolvers.

Earlier work leveraged ECS to study content delivery networks. In this paper we show that ECS can also be used to study the day-to-day operations of a public DNS resolver, in our case GPDNS. This allowed us to show that traffic to GPDNS is frequently routed to out-of-country GPDNS PoPs for weeks at a time, even though an in-country PoP is available. This potentially exposes DNS traffic to state-level surveillance. Additionally, we showed that it is not uncommon for a suboptimal PoP to be selected, leading to additional round-trip times for DNS queries. We also showed that certain events such as DDoS attacks on ISP DNS resolvers cause users to switch to GPDNS *en masse*, and, more importantly, once users have switched to Google they do not switch back.

A previously unrecognized privacy issue is that e-mail servers frequently use GPDNS for DNS resolution. Obviously, this reveals information to Google where these servers receive mail from and send mail to. Much more insidious though, is, that this information also leaks to operators of authoritative name servers through Google's use of ECS. Where previously mail servers were hidden behind DNS resolvers of network operators, they are now exposed up to the /24 IP prefix level for IPv4, and /56 for IPv6 (was /64 until the 13th of December 2017).

Taken together, we can conclude that not only should the use of public DNS resolvers in general be questioned; given the privacy implications, the use of the ECS DNS extension introduced specifically for public DNS resolvers should also be re-examined. Given our findings, we strongly advocate restricting use of ECS toward content delivery networks only, on an opt-in basis, to prevent leaking unnecessary information.

ACKNOWLEDGMENT

This paper used IP2Location LITE data available from https://lite.ip2location.com.

REFERENCES

[1] Google. (2014). *Google Public DNS and Location-Sensitive DNS Responses*. [Online]. Available: https://webmasters.googleblog.com/2014/12/google-public-dns-and-location.html
[2] C. Contavalli, W. van der Gaast, D. Lawrence, and W. Kumari, "Client subnet in DNS queries," Internet Eng. Task Force, Fremont, CA, USA, RFC 7871, 2016. [Online]. Available: https://tools.ietf.org/html/rfc7871
[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. ACM IMC*, Boston, MA, USA, 2012, pp. 523–536.
[4] M. A. Sánchez *et al.*, "Dasu: Pushing experiments to the Internet's edge," in *Proc. USENIX NSDI*, Lombard, IL, USA, 2013, pp. 487–499.
[5] F. Streibelt, J. Böttger, N. Chatzis, G. Smaragdakis, and A. Feldmann, "Exploring EDNS-client-subnet adopters in your free time," in *Proc. ACM IMC*, Barcelona, Spain, 2013, pp. 305–312.
[6] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan, "Mapping the expansion of Google's serving infrastructure," in *Proc. ACM IMC*, Barcelona, Spain, 2013, pp. 313–326.

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on November 02,2020 at 10:26:19 UTC from IEEE Xplore. Restrictions apply.

[7] X. Fan, E. Katz-Bassett, and J. Heidemann, "Assessing affinity between users and CDN sites," in *Traffic Measurements and Analysis* (LNCS 9053), M. Steiner, P. Barlet-Ros, and O. Bonaventure, Eds. Heidelberg, Germany: Springer, 2015, pp. 95–110.

[8] D. Giordano *et al.*, "A first characterization of anycast traffic from passive traces," in *Proc. IFIP Workshop Traffic Monitor. Anal. (TMA)*, 2016, pp. 30–38.

[9] 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. IEEE Netw. Traffic Meas. Anal. Conf. (TMA)*, 2018, pp. 1–8.

[10] P. Mockapetris, "Domain names—Concepts and facilities," Internet Eng. Task Force, Fremont, CA, USA, RFC 1034, 1987. [Online]. Available: https://tools.ietf.org/html/rfc1034

[11] P. Mockapetris, "Domain names—Implementation and specification," Internet Eng. Task Force, Fremont, CA, USA, RFC 1035, 1987. [Online]. Available: http://tools.ietf.org/html/rfc1035

[12] J. Damas, M. Graff, and P. Vixie, "Extension mechanisms for DNS (EDNS0)," Internet Eng. Task Force, Fremont, CA, USA, RFC 6891, 2013. [Online]. Available: http://tools.ietf.org/html/rfc6891

[13] CAIDA. (2018). *The CAIDA UCSD Routeviews Prefix to AS Mappings Dataset (pfx2as) for IPv4 and IPv6*. [Online]. Available: http://www.caida.org/data/routing/routeviews-prefix2as.xml

[14] Google. (2018). *Google Public DNS FAQ*. [Online]. Available: https://developers.google.com/speed/public-dns/faq#locations

[15] Y. Yu, D. Wessels, M. Larson, and L. Zhang, "Authority server selection in DNS caching resolvers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 2, p. 80, 2012.

[16] M. Müller, G. Moura, R. de O Schmidt, and J. Heidemann, "Recursives in the wild: Engineering authoritative DNS servers," in *Proc. ACM Internet Meas. Conf.*, 2017, pp. 489–495.

[17] W. B. de Vries, R. de O Schmidt, W. Hardaker, J. Heidemann, P.-T. de Boer, and A. Pras, "Broad and load-aware anycast mapping with verfploeter," in *Proc. ACM IMC*, 2017, pp. 477–488.

[18] F. Chen, R. K. Sitaraman, and M. Torres, "End-user mapping: Next generation request routing for content delivery," in *Proc. ACM SIGCOMM*, London, U.K., 2015, pp. 167–181.

[19] B. Gueye, S. Uhlig, and S. Fdida, "Investigating the imprecision of IP block-based geolocation," in *Proc. Int. Conf. Passive Active Netw. Meas.*, 2007, pp. 237–240.

[20] B. Huffaker, M. Fomenkov, and K. Claffy, "Geocompare: A comparison of public and commercial geolocation databases-technical report," Cooperat. Assoc. Internet Data Anal., La Jolla, CA, USA, Rep., 2011.

[21] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, "IP geolocation databases: Unreliable?" *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 53–56, 2011.

[22] Y. Shavitt and N. Zilberman, "A geolocation databases study," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 10, pp. 2044–2056, Dec. 2011.

[23] NOS. (Aug. 2015). *Is Your Provider Under Attack? This Is How You Restore Service. [in Dutch]*. [Online]. Available: https://nos.nl/op3/artikel/2052944-aanval-op-je-provider-zo-krijg-je-weer-verbinding.html

[24] M. Kucherawy, "Message header field for indicating message authentication status," Internet Eng. Task Force, Fremont, CA, USA, RFC 7601, 2015. [Online]. Available: https://tools.ietf.org/html/rfc7601

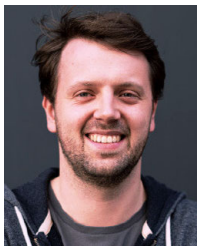
[25] J. Klensin, "Simple mail transfer protocol," Internet Eng. Task Force, Fremont, CA, USA, RFC 5321, 2008. [Online]. Available: https://tools.ietf.org/html/rfc5321

[26] P. Kintis, Y. Nadji, D. Dagon, M. Farrell, and M. Antonakakis, *Understanding the Privacy Implications of ECS* (LNCS 9721). Cham, Switzerland: Springer, 2016, pp. 343–353.

[27] Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, and K. Claffy, "Two days in the life of the DNS anycast root servers," in *Proc. Int. Conf. Passive Active Netw. Meas.*, 2007, pp. 125–134.

[28] S. Castro, D. Wessels, M. Fomenkov, and K. Claffy, "A day at the root of the Internet," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 5, pp. 41–46, 2008.

[29] Z. Li, D. Levin, N. Spring, and B. Bhattacharjee, "Internet anycast: Performance, problems and potential," in *Proc. ACM SIGCOMM*, Budapest, Hungary, 2018, pp. 59–73.



Wouter B. de Vries is currently pursuing the Ph.D. degree with the Design and Analysis of Communication Systems Group, University of Twente. His current research area is the optimization of anycast catchments, specifically to improve resilience against distributed denial-of-service attacks.

Roland van Rijswijk-Deij received the Ph.D. degree in computer science from the University of Twente in 2017, where he is an Assistant Professor of network security with the Design and Analysis of Communication Systems Group. He is also a Principal Scientist with NLnet Labs, a not-for-profit foundation that develops open source software for core Internet protocols, including DNS and RPKI. His research interests include network security and network measurements, with a particular interest in DNS and DNSSEC.

Pieter-Tjerk de Boer received the M.Sc. degree in applied physics and the Ph.D. degree in computer science from the University of Twente in 1996 and 2000, respectively, where he is an Assistant Professor with the Design and Analysis of Communication Systems Group. His research interests include communication networks, their mathematical performance modeling and efficient simulation techniques, and software-defined radio.

Aiko Pras received the Ph.D. degree from the University of Twente, The Netherlands, with a thesis entitled "Network Management Architectures" in 1995, where he has been a Full Professor of network operations and management with a focus on Internet security with the Faculty of Electrical Engineering, Mathematics and Computer Science in 2013. He was a recipient of the IFIP/IEEE "Salah Aidarous Memorial Award" for providing unremitting service and dedication to the IT and Telecommunications Network Operations and Management Community in 2016. He is a member of the Design and Analysis of Communication Systems Group.

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on November 02,2020 at 10:26:19 UTC from IEEE Xplore. Restrictions apply.