Exploring EDNS-Client-Subnet Adopters in your Free Time*

Florian Streibelt
TU Berlin
florian@inet.tu-berlin.de

Jan Böttger
TU Berlin
jan@inet.tu-berlin.de

Nikolaos Chatzis
TU Berlin
nikolaos@inet.tu-berlin.de

Georgios Smaragdakis
T-Labs/TU Berlin
georgios@net.t-labs.tu-berlin.de

Anja Feldmann
TU Berlin
anja@inet.tu-berlin.de

ABSTRACT

The recently proposed DNS extension, EDNS-Client-Subnet (ECS), has been quickly adopted by major Internet companies such as Google to better assign user requests to their servers and improve end-user experience. In this paper, we show that the adoption of ECS also offers unique, but likely unintended, opportunities to uncover details about these companies' operational practices at almost no cost. A key observation is that ECS allows to resolve domain names of ECS adopters on behalf of any arbitrary IP/prefix in the Internet. In fact, by utilizing only a single residential vantage point and relying solely on publicly available information, we are able to (i) uncover the global footprint of ECS adopters with very little effort, (ii) infer the DNS response cacheability and end-user clustering of ECS adopters for an arbitrary network in the Internet, and (iii) capture snapshots of user to server mappings as practiced by major ECS adopters. While pointing out such new measurement opportunities, our work is also intended to make current and future ECS adopters aware of which operational information gets exposed when utilizing this recent DNS extension.

General Terms

Measurement.

Keywords

Content Delivery; DNS; CDN.

1. INTRODUCTION

In the current Internet, hostnames are typically resolved using the local resolvers provided by the respective Internet Service Provider (ISP). Unless the answer is cached, the ISP's domain name server performs a recursive lookup to receive an authoritative answer which it can then cache. Large Content Delivery Networks (CDNs) and Content Providers (CPs) use the domain name system (DNS) to map users to possible locations and consequently to appropriate servers [27, 38].

Unfortunately for the CDNs and CPs, the DNS query is not directly issued by the end-user but only by the local resolver (see [37] for details). Thus, the assumption underlying this solution is that the end-user is close to the local resolver or that a resolver serves clients that are close to each other. However, several studies [11, 26] have shown that these assumptions do not always hold which, in turn, can lead to degraded end-user experience. In particular, the introduction of third party resolvers like Google Public DNS [4] or OpenDNS [7] has exaggerated this trend as end-users, taking advantage of these as resolvers, may experience poor performance[11, 28, 32]. This empirical evidence is mainly due to the fact that these popular third-party resolvers are typically not located within the end-users' ISP and are therefore often not close to the end-user.

The solution, as proposed by Google and others to the IETF [17], is the EDNS-Client-Subnet DNS extension (ECS). While traditional DNS queries do not include the IP address of the query issuer (except via the socket information), ECS includes IP address information of the original query issuer in the query to the authoritative name server. The IP address information can then be used by the CDN or CP to improve the mapping of end-users to servers. Thus, it is not surprising that major Internet companies (e.g., Google, Edgecast, and OpenDNS) have already adopted ECS and have established the consortium "a faster Internet" [1]. The fact that ECS can indeed help improve end-user performance is highlighted by extensive active measurement studies [28, 32].

For the Internet measurement community, the adoption of ECS by some of the major Internet players turns out to offer unique but clearly unintended opportunities. To illustrate, we show in this paper how ECS can be used to uncover details of the operational practices of ECS adopters with almost no effort. Our key observation is that ECS allows anyone to issue queries on behalf of any "end-user" IP address for domains with ECS support. Thus, ECS queries can help uncovering the sophisticated techniques that CDNs and CPs use for mapping users to servers (e.g, see [24, 23, 26]). Indeed, this currently hard-to-extract information can now be collected using only a single vantage point and relying on publicly available information. In the past, to obtain similar information, network researchers had to find and use open or mis-configured resolvers [10, 22, 35], have access to a multitude of vantage points in edge networks [28, 32], rely on volunteers [11, 12], analyze proprietary data [25, 30], or resort to searching the Web [34].

In summary, the three contributions of this paper to the area of Internet measurement are:

- We show that a single vantage point combined with publicly available information is sufficient to uncover the global footprint of ECS adopters and track their expansions.
- We demonstrate how to infer the DNS cacheability and end-user clustering strategies of ECS adopters.

* **This work would not have been possible without the help, engagement, and commitment of Walter Willinger.**

IMC'13, October 23–25, 2013, Barcelona, Spain.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-1953-9/13/10 ...$15.00.
http://dx.doi.org/10.1145/2504730.2504767.

- We illustrate how to capture snapshots of the assignment of users to server locations performed by major ECS adopters.

At the same time, this work is also intended to increase the awareness of current and future ECS adopters about which operational information gets exposed when enabling this recent DNS extension. Despite the fact that experts in the operational community may be aware of some of the shortcomings and consequences of the ECS adoption, a systematic study is still missing. Our software and measurements are publicly available[1].

2. ECS BACKGROUND

The EDNS-Client-Subnet DNS extension [17] was introduced to tackle the problem of mis-locating the end-system which originates the DNS request. The problem is that the end-system's IP information is typically hidden from the authoritative name server. With ECS, client IP information is forwarded by all ECS-enabled resolvers to the authoritative name server in the form of network prefixes.

2.1 Protocol Specification

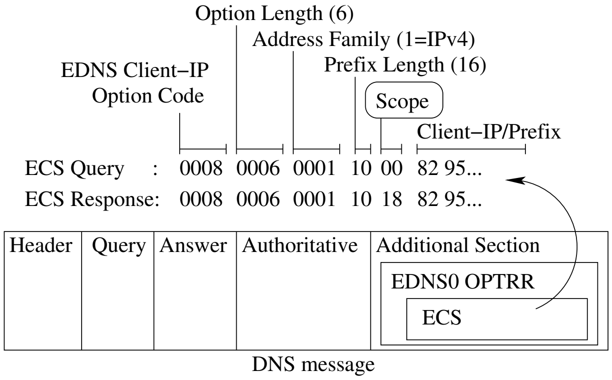
ECS is an EDNS0 DNS extension [36] proposed by the IETF DNS Extensions Working Group. EDNS0, which is also needed for DNSSEC, uses an ADDITIONAL section in DNS messages to transfer optional data between name servers. Since all sections from a DNS query are present in the DNS response, bidirectional data transfer is enabled as the responder can modify this section. Name servers that do not support EDNS0 either strip the EDNS0 OPTRR in the ADDITIONAL section or forward it unmodified.

An example ECS-enabled query and response is shown in Figure 1. The ADDITIONAL section includes an OPTRR resource-record containing the ECS header and data. The ECS payload consists of the address family used, i.e., IPv4 or IPv6, prefix length, scope and client prefix. To protect a client's privacy, [17] recommends to use prefixes less specific than 32. In each query the scope field must be zero and is a placeholder for the returned scope.

The response from an ECS-enabled DNS server differs in one byte, namely the scope, which is needed for DNS caching. The answer can be cached and used for any query with a client prefix that is a more specific or equal to the prefix as specified by the scope. We note that the response may contain a different scope than the query network mask, and we have indeed observed larger as well as shorter scopes than prefix length in our measurements. In the example, the query prefix length is 16 while the returned scope is 24. The scope is the essential element that allows us to infer operational practices of ECS adopters.

2.2 Challenges in Enabling ECS

While ECS is transparent to the end-user, it requires significant efforts by the DNS server operators, mainly because all involved DNS servers have to at least forward the ECS information. Among the major obstacles are: (i) ECS-support in server software is not widely available[2], (ii) all involved DNS servers need to be upgraded[3], and (iii) third-party resolvers are not necessarily sending ECS queries by default. To change the latter, an engineer of, say, Google Public DNS or OpenDNS, has to manually check the authoritative name servers and white-list them as ECS compliant.



```
# dig www.google.com +client=130.149.0.0/16 @ns1.google.com
```

Figure 1: Example of ECS query and response.

Moreover, appropriate cache support has to be added to the DNS resolvers. Here, ECS with its notion of scope introduces another problem. For each query, the resolver has to check if the IP address of the client lies within the scope of any cached result. If not, the query has to be relayed with the appropriate prefix information. Just imagine the extreme scenario—scope of 32. Then, the resolver should keep a separate entry per client making caching largely ineffective.

Overall, handling ECS is quite complicated as the draft requires DNS forwarders to forward the ECS information sent by the client. It may modify the prefix mask to a less specific. If no ECS information is present in the DNS request, the forwarder may add an OPTRR record based on information from the socket. This is the rule followed e.g., by Google's public DNS servers. Note, that until Google enabled EDNS0 for DNSSEC support, it stripped the ECS records. Now, this information seems to be forwarded unmodified.

3. DATASETS

The following two different kinds of datasets are used in this paper: (i) prefixes to be used as pretended "client location" for the ECS queries, and (ii) popular ECS adopters.

3.1 Network Prefixes

In principle, one list of prefixes may be considered sufficient. However, because we want to uncover the operational practices of certain CDNs and CPs with regards to client localization and clustering, we explore different prefix sets of varying scopes and magnitudes. To that end, we use both private and public sets of network prefixes.

Academic Network (UNI). This network includes a very diverse set of clients, ranging from office working spaces to student dorms and research facilities which complicates usage profiling. This network uses two /16 blocks, does not have an AS, and is localized to a single city in Europe.

Large ISP (ISP). The dataset includes more than 400 prefixes, ranging from /10 to /24, announced by a European tier-1 ISP. This ISP offers services to residential users as well as enterprise networks and also hosts CDN servers.

Large ISP, de-aggregated prefixes (ISP24). We also use the de-aggregated announced prefixes of our large ISP at the granularity of /24 blocks to investigate if the finer granularity lets us uncover additional operational details.

Popular Resolvers (PRES). This proprietary dataset consists of the 280K most popular resolver IPs that contacted a large commer-

[1] `http://projects.inet.tu-berlin.de/projects/ecs-adopters`

[2] We find that only PowerDNS supports ECS as authoritative nameserver but not as resolver. Moreover, only for some clients e.g., dig and dnspython patches are available.

[3] CDNs may internally use multiple resolution levels, e.g., Akamai [27].

cial CDN. These resolvers are distributed across 21K ASes, 74K prefixes, and 230 countries.

RIPE. RIPE RIS [8] makes full BGP routing tables from a multitude of BGP peering sessions publicly available. This data includes 500K prefixes from 43K ASes.

Routeviews (RV). This is another public BGP routing table source offered by the University of Oregon [9].

3.2 Content Provider Datasets

For our experiments, we need to identify ECS adopters and the corresponding hostnames. For this purpose, we utilize Alexa [2] (April 20, 2013), a publicly available database of the top 1 Million second-level domains, as was done by Otto et al.[28]. Since the ECS extension does not allow us to directly find out if a name server is ECS enabled or not, we use the following heuristic. We re-send the same ECS query with three different prefix lengths. If the scope is non-zero for one of the replies, we annotate the server and hostname as ECS-enabled.

We obtain two disjoint groups of (domain names, name-servers) pairs. The first group fully supports ECS and accounts for 3% of the second-level domain names. The second group, about 10%, is ECS-enabled according to the IETF draft [17] but does not appear to use the additional section information for the tested domains (it may well just return a copy of the additional section). Thus, roughly 13% of the top 1 million domains may be ECS-enabled. This number is only slightly larger than what was reported in a 2012 study [28]. However, some of the big players are among the ECS adopters, including Google (and YouTube), Edgecast, CacheFly, HiCloud, and applications hosted in the cloud such as MySqueezebox.

To estimate the potential traffic affected by ECS, we use a 24 hour anonymized packet-level trace from a large European ISP. The trace is from a residential network with more than 10K active end-users and was gathered using Endace cards and analyzed with Bro [29]. It contains 20.3 million DNS requests for more than 450K unique hostnames and 83 million connections. While Alexa only includes the second-level domain names, this dataset allows us to identify full hostnames, which we use in a similar manner as above. In total, we find that roughly 30% of the traffic involves ECS adopters. This highlights that while the number of ECS adopters is relatively small (less than 3% of the authoritative name servers), it includes some of the relevant players responsible for a significant fraction of traffic.

From the set of identified ECS adopters, we select a large CDN, two smaller CDNs, and an application deployed in the cloud to explore their operational practices. In the rest of the paper we mainly focus on:

Google is a founding member of the consortium "a faster Internet" and one of the main supporters of ECS. It has adopted ECS in all their resolvers and name servers. Moreover, Google uses a sophisticated backend with many data centers, edge-servers, and Google Global Cache (GGC) servers located inside ISPs [3, 13, 19]. It is known that there can be anywhere between tens to thousands of Google servers behind a single Google IP [33].

Edgecast is a large CDN that also offers streaming solutions. It is also one of the participants in the "a faster Internet" consortium.

CacheFly is another CDN that has adopted ECS.

MySqueezebox is a Logitech product that runs on top of Amazon's Web cloud Service EC2.

4. METHOD

For our experiments, we take advantage of the ECS extension of the python DNS libraries provided by OpenDNS [7]. Based on this library, we have developed a framework which utilizes the above API to send ECS DNS queries with arbitrary ECS client subnet information to authoritative name servers. By embedding this library into our test framework, we can handle failures and retries efficiently which would have been more complicated with a standalone utility like the patched dig tool [17].

We emphasize that a single vantage point is sufficient for performing our experiments, as with ECS the answers exclusively depend on the client prefix sent. This is confirmed by synchronized measurements from two research networks (US, Germany) and a German hosting provider.

It is sufficient to use a commodity PC, issuing queries at the rate of 40 to 50 queries per second. This rate that can be achieved at a residential vantage point with no complications [11].

Scaling up the query rate is easy by using multiple vantage points in parallel, e.g., by utilizing PlanetLab nodes, but our experiments show that a simple setup serves the purpose of this study.

With regards to the queries, we use hostnames from the Alexa list and the ISP traces and the source prefixes from our prefix datasets. Note, that because a large fraction of IPv6 connectivity is still handled by 6to4 tunnels [16] and related techniques, we do not include IPv6 in this preliminary study.

For each query we issue we add an entry to our SQL database which includes all parameters including the timestamp, the returned records (answers) including TTL and returned scope. Before and after each experiment, we collect the most recent prefixes for each dataset. To speed-up the experiments, we compile a set of unique prefixes before starting an experiment.

5. EVALUATION

To evaluate the capabilities of ECS as a measurement tool, we explore how difficult it is to (i) uncover the footprint of ECS adopters, (ii) assess the effect of ECS on cacheability of DNS records, and (iii) capture snapshots of how ECS adopters assign users to server locations. All queries are sent for a single hostname (e.g., `www.google.com`) to one of the authoritative name servers of the service provider (e.g., `ns1.google.com`).

While for the RIPE, RV, ISP and ISP24 dataset we use the prefixes as announced, for the UNI dataset an ECS prefix length of 32 is chosen, as this dataset contains individual IP addresses.

5.1 Uncovering Infrastructure Footprints

We first report on our experiences with using ECS to uncover the footprint of the four selected ECS adopters. Apparently the operational community also did some investigation [5] in enumerating CDN servers using ECS.

Table 1 summarizes the number of unique server IPs, subnets, ASes, and locations. The footprint of Google is by far the most interesting one, with more than 6,300 server IPs across 166 ASes in 47 countries[4]. We also notice that GGCs are typically located in ASes that are "categorized" as enterprise customers and small transit providers [18] in both developed and developing countries. In March 2013, Google servers are found in 81 enterprise customers, 62 small transit providers, 14 content/access/hosting providers, and only 4 large transit providers (a small number of ASes that host Google server IPs do not belong to any of these categories). While

[4] For geolocation we use MaxMind [6]. We are aware that geolocation of CDN servers can be inaccurate, e.g., MaxMind maps the IPs of the main Google AS (AS15169) to California, but it is accurate on the country level for IPs that belong to ISPs [31] and thus, good enough for the purpose of this study. A more sophisticated approach of geolocating Google server IPs in the Google AS is presented in [14].

	Prefix set	Server IPs	Sub nets	ASes	Countries
Google (03/26/13)	RIPE	6,340	329	166	47
	RV	6,308	328	166	47
	PRES	6,088	313	159	46
	ISP	207	28	1	1
	ISP24	535	44	2	2
	UNI	123	13	1	1
MySqueezebox (03/26/13)	ALL \ UNI	10	7	2	2
	UNI	6	4	1	1
Edgecast (04/21/13)	RIPE/RV/PRES	4	4	1	2
	ISP/ISP24/UNI	1	1	1	1
CacheFly (04/21/13)	RIPE/RV	18	18	10	10
	PRES	21	21	11	11
	ISP	6	6	5	5
	ISP24	5	5	4	4
	UNI	1	1	1	1

Table 1: ECS adopters: Uncovered footprint.

Date (RIPE)	IPs	Sub nets	ASes	Countries
2013-03-26	6340	329	166	47
2013-03-30	6495	332	167	47
2013-04-13	6821	331	167	46
2013-04-21	7162	346	169	46
2013-05-16	9762	485	287	55
2013-05-26	9465	471	281	52
2013-06-18	14418	703	454	91
2013-07-13	21321	1040	714	91
2013-08-08	21862	1083	761	123

Table 2: Google growth within five months.

illustrative and also informative (e.g., we uncover more locations than previously reported [12, 33]), the main and more surprising finding is the simplicity with which we can uncover this infrastructure using ECS from a single vantage point in less than 4 hours.

For validation purposes, we check each server IP—all of them serve us the Google search main page. In addition, the reverse look-up reveals that while all servers inside the official Google AS use the suffix 1e100.net [21], those deployed in third-party ASes use different hostnames (e.g., cache.google.com, or names containing the strings ggc or googlevideo.com). In some cases we observe legacy names that indicate the prior use of the IP range by the ISP. This means we cannot infer the presence of a GGC purely by looking at the reverse DNS zones.

5.1.1 Choosing the Right Prefix Set

Both the RIPE as well as the RV prefix sets are sufficiently complete to yield the same results. We attribute this to the fact that the advertised address space of both datasets overlaps significantly. We see around 500K announced prefixes in the data sets at various aggregation levels. Using only the most specifics without overlap this reduces to about 130K prefixes. For our experiments we decided to use the prefixes as announced. We think this corresponds to the distribution to be seen at an ECS enabled nameserver[5] and reflects the public IP-address space being used.

Next we compare our results (RIPE only) to a study of Calder et al.[14], where queries were made using /24 prefixes. We see a 94% overlap in the discovered Google server IP-addresses while issuing significantly less DNS queries in our approach.

PRES however is not sufficient to uncover the full set of Google Web servers, but yields a major fraction of them in only 55 minutes per experiment. Alternatively, one can use a subset of the RIPE/RV prefix sets. Using a random prefix from each AS reduces the number of RIPE/RV prefixes to 43,400 (8.8% of RIPE prefixes) and results in 4,120 server IPs in 130 ASes and 40 countries in 18 minutes (with 40 requests/second). By doubling the number of selected prefixes to two per AS, we uncover 4,580 server IPs in 143 ASes, and 44 countries.

When relying on the ISP, ISP24, and UNI data sets, we see the effect of mapping end-users to server IPs using ECS. In the case of Google we uncover a much smaller number of servers. However, by using the de-aggregated prefix set of the ISP (i.e., ISP24), we are able to expand the coverage from 200 to more than 500 server IPs. More than 95% of them are in the Google AS while the rest is located in a neighbor AS to that ISP. A more careful investigation reveals that the client prefixes served from the neighbor AS are from a customer of this ISP whose prefix is not announced separately but only in aggregated form (i.e., together with other prefixes of the ISP). Our conjecture is that this is due to the BGP feed sent to the GGC by the ISP [13].

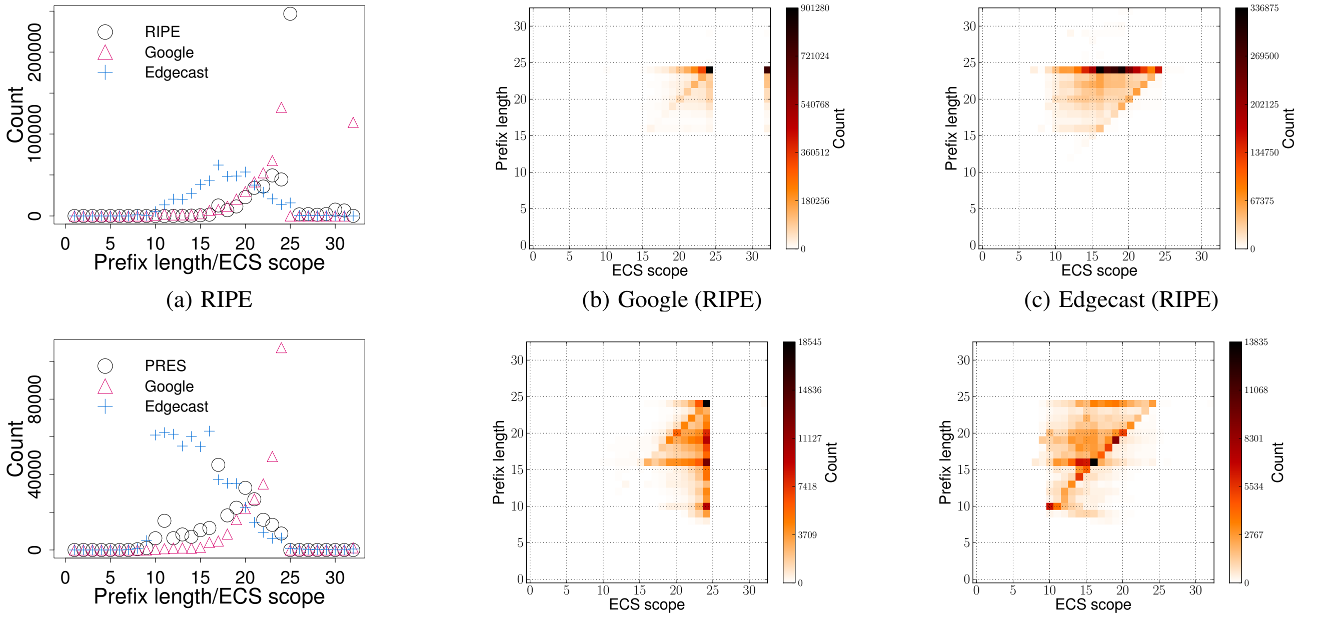
Of the ASes uncovered by using the RIPE prefix set, only 845 and 96 server IPs are in the ASes of Google and YouTube, respectively. All the others IPs are in ASes not associated with Google. This shows the profound effect of GGCs which have been deployed to many ASes. We repeat the experiments by using the Google Public DNS server and observe that the returned answers are almost always identical (99%). This is not necessarily the case when using Google's Public DNS server for other lookups. However, we find Google's Public DNS server forwarding our ECS queries unmodified to white-listed authoritative DNS servers of other ECS adopters. Therefore, we can even (ab)use Google's Public DNS server as intermediary for measurement queries and thus (i) hide from discovery or (ii) explore if these ECS adopters use a different clustering for Google customers.

Table 1 also shows that the footprints of the other ECS adopters are "less" interesting, mainly because their footprint is not as widely distributed compared to Google. Nevertheless, we see in principle similar results. Most of the infrastructure can be uncovered with the RIPE/RV/PRES prefix sets. The ECS adopters again use clustering such that the ISP, ISP24, and UNI prefixes are all mapped to a single server IP. Note that Edgecast may use HTTP-based redirection which cannot be uncovered using only DNS. While Edgecast uses a single AS, CacheFly, and MySqueezebox are utilizing infrastructures across multiple ASes. We also observe that both players map the UNI and ISP/ISP24 prefixes to infrastructures in Europe (e.g., MySqueezebox maps them to the European facility of Amazon EC2).

5.1.2 Tracking the Expansion of CDNs Footprints

Our method allows us to track the expansion of ECS adopters' footprints over time. This becomes increasingly important as many CDNs continuously deploy servers at the network edges or within ISPs. Thus, one can not infer the operator of a cache by simply looking at the IP address or AS number [15]. As we show above, RIPE and RV public prefix sets uncover by far more IPs than the other prefix sets. We use the RIPE prefix set to track the expansion of ECS adopters as it is updated more frequently than RV. In Table 2 we report the rapid increase of discovered Google server IPs over a four month period (March-August 2013). We observe that the number of Google server IPs at least triples (345%), the number of ASes hosting Google infrastructure increases by 595 (458%) and the global presence at least doubles (261%). In August 2013, Google servers are found in 372 enterprise networks, 224 small transit providers, 102 content/access/hosting providers, and 11 large transit providers. Starting mid May we include the YouTube website in our measurements and notice that while the

[5] A study on this is currently being performed.



(a) RIPE (b) Google (RIPE) (c) Edgecast (RIPE)

(d) PRES (e) Google (PRES) (f) Edgecast (PRES)

Figure 2: Prefix length vs. ECS scope for RIPE and PRES (Google: March 2013, Edgecast: May 2013).

number of ASes with YouTube servers almost triples (271%) from 220 to 598, these ASes overlap with already uncovered Google infrastructure. We figure this being a result of incorporating the YouTube infrastructure into Google's global platform. By merging the sets of IP addresses for Google and YouTube, the count only increases to 24048. For an in-depth study of the expansion of Google infrastructure since November 2012 we refer to [14]. Our study did not uncover significant growth of serving infrastructure for the other ECS adopters.

5.2 Uncovering DNS Cacheability

Next, we examine the ECS specific information included in the DNS response: the scope. In principle, if the ECS information corresponds to a publicly announced prefix, one may expect that the returned scope is equal to the prefix length. However, this is not necessarily the case. Content providers often return either coarser- or finer-grained scopes e.g., they respond either with aggregated or de-aggregated prefixes. This indicates that they perform the end-user clustering for client-to-server assignment on a different granularity than the routing announcements.

We note that the scope may have a major effect on the re-usability of the DNS response i.e., the cacheability of DNS responses. While most of the responses have a non-zero TTL, a surprisingly large number have a /32 scope. An ECS scope of /32 implies that the answer is valid only for the specific client IP which issued the DNS request. In this section, we explore DNS cacheability for two ECS adopters in detail: Google and Edgecast. The others are less interesting as CacheFly always uses a /24 scope and MySqueezebox is similar to Edgecast.

Figure 2(a) shows the RIPE prefix length distribution (circles). In addition, it includes the returned scopes from using the RIPE prefixes to query our ECS adopters. We note, that the distributions vary significantly. There is massive de-aggregation by Google but massive aggregation by Edgecast which operates a smaller infrastructure.

When executing back-to-back measurements for Google (e.g., 4 queries within a second), we find that typically both the answer and scopes are consistent within the duration of the TTL (300 seconds for Google). The answer as well as the scope can change in some cases within seconds[6]. A detailed study of the temporal changes of the returned scope is part of our future work. Overall, as seen in Figure 2(a), for almost a quarter of the queries, the returned scope is 32. This indicates that currently Google severely restricts the cacheability of ECS responses or may want to restrict reuse of the answers to single client IPs. For approximately 27% of the queries prefix length and scope are identical. For 41% of the queries we see de-aggregation while there is aggregation for 31%. The difference between the announced prefix and the returned scope may also be due to the the BGP feed sent to the GGC by the ISP [13] that is not necessarily the same that is publicly announced and collected by RIPE or Routeviews. We again note, that the returned scopes for the RIPE and RV prefixes are almost identical.

Exploring the scopes returned by Edgecast may at first glance appear useless, because they only returned a single IP with a TTL of 180 seconds. However, Edgecast is using significant aggregation for all prefix lengths across all prefix sets. For example, when using the RIPE prefixes, the return scope is identical for 10.5% but less specific for 87%.

When using the ISP prefix set, the overall picture is similar even though the specific numbers vary. An initial study of the prefixes with scope 32 indicates that Google performs profiling, e.g., Google returns scope /32 for all CDN servers of a large CDN provider inside the ISP. In future work, we plan to explore if there exists a natural clustering for those responses with scope /32.

Given that the size of the UNI prefix set is limited, we issue queries from all IP addresses with prefix length 32. The returned scopes vary heavily from /32 to /15, even for neighboring IP addresses.

For the PRES prefixes, Figure 2(d) shows extreme de-aggregation. For more than 74% of the prefixes, the scope is more restric-

[6] This can be attributed to the fact that authoritative nameservers may use anycast or load balancing [11] or because the rapid increase of DNS queries (e.g., from our measurements) triggers a change on IP/prefix to server mapping.

tive than the prefix length, and in 17% they are identical. Only few returned scopes are /32s. This may indicate that Google treats popular resolvers differently than random IP addresses. Google may already be aware of the problem regarding caching DNS answers as discussed in Section 2.2. For Edgecast we see significant aggregation.

To highlight the relationship of prefix length in the query to scope in the reply, Figures 2(b) and 2(e) show heatmaps of the corresponding two-dimensional histograms. For the RIPE dataset we notice the two extreme points at scopes /24 and /32. For the PRES dataset, the heatmap highlights the de-aggregation. Figures 2(c) and 2(f) show the heatmaps for Edgecast. While for the RIPE dataset we see the effect of the extreme prefix de-aggregation for Google very clearly, the picture for Edgecast is more complicated as there is mainly aggregation. For the PRES dataset, the heatmap shows even more diversity as there is de-aggregation as well as aggregation. This results in a blob in the middle of the heatmap.

5.3 User-Server Mapping Snapshots

So far we have not yet taken advantage of the Web server IP addresses in the DNS replies. These allow us to capture snapshots of the user-to-server mapping employed by an ECS-enabled CDN or CP that can be used to shed light on CDN mapping strategies. In the following, we illustrate the measurement capabilities offered by ECS. We explore snapshots of Googles' user-to-server mappings (based on the RIPE data set) and examine how stable this mapping is.

Google returns 5 to 16 different IP addresses in each reply. Almost all responses (>90%) include either 5 or 6 different IP addresses. We do not find any correlation between the ECS prefix length or the returned scope and the number of returned IP addresses. All IP addresses from a single response always belong to the same /24 subnet (the returned IPs are not necessarily in close geographic distance to each other [20]). We also notice that typically the announced prefix length of subnets that host Google servers is /24. Thus, based on a single ECS lookup per prefix, we always find a unique mapping between query prefix and the server subnet from the DNS reply.

Next we assess the mapping consistency at AS-level. First we map all all prefixes used in the ECS queries to their corresponding AS. Then, by looking at the returned A records, we find the corresponding server ASes for each client AS.

On March 26 2013, the majority of client ASes, around 41K, was served exclusively by Google servers from a single AS. About 2K ASes were served by servers in 2 ASes, and less than 100 ASes were served by servers from more than 5 ASes. On August 8, 2013 the number of ASes that served by a single AS dropped to around 38.5K and around 5K served by 2 ASes. ASes served by a large number of server ASes typically have a global footprint. We find that client prefixes of ASes that host GGC are also served by servers in other ASes. This is to be expected as GGC capacity may not always be sufficient to handle demand and also because different prefixes within an AS, e.g., those that host the GCC servers, may be handled differently. As illustrated in Figure 5.3 a small number of ASes hosts servers that serve a large number of ASes. By far the most popular AS is the official Google AS (AS15169) that served more than 41.5K ASes in March and around 40.5K in August 2013. In the top-10 we find the YouTube AS, as well as small and large transit providers that serve their customers. There is also a small number of ASes that exclusively serve their client subnets from GGC servers they host.

From our analysis we derive some important observations. First, the Google content is not any more exclusively served by servers in Google ASes, as was reported in [12]. Second, GCCs have been enabled in a significant number of ASes over a five months period. Third, within these five months the number of ASes that are served by GGC servers in other ASes has been significantly increased.

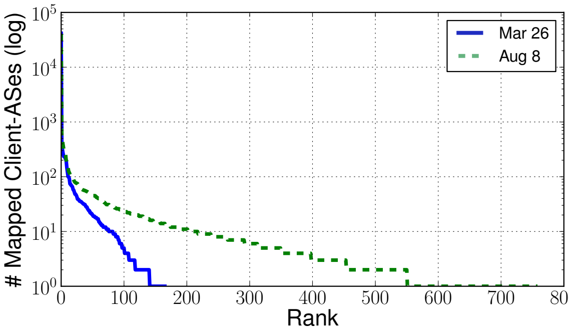


Figure 3: # ASes served by ASes with Google servers (RIPE).

This trend has significant implications to caching as Google content can be available in the same or a neighboring ASes. It also has implication to peering as the presence of GGCs reduces the inter-domain Google traffic and it is now possible for smaller networks to reduce their transit cost by either install GGCs or peer with ASes that host GGCs.

To assess the stability of user-server mapping over time, we analyze the returned IP addresses when asking back-to-back queries over two days (May 3-4, 2013). We found that around 35% of the prefixes are always served by a single /24 block over the 48 hours period. Given the highly distributed infrastructure of Google one may have expected larger churn. 44% of the query prefixes are mapped to two /24 and a very small percentage to more than five /24s. A possible explanation for this stable mapping is that Google uses local load-balancers [24, 33]. We leave the study of temporal dynamics in user-to-server mapping over longer periods and during flash crowds or other events as future work. Our future research agenda also includes the study of the temporal changes in user-to-server mapping not just for Google but also for other ECS adopters.

6. CONCLUSION

In this paper we show that the adoption of the EDNS-Client-Subnet DNS extension (ECS) by major Internet companies offers unique but most likely unintended measurement opportunities to uncover some of their operational practices. Using early ECS adopters like Google, Edgecast, and CacheFly as examples, our experimental study shows how simple it is (using a single vantage point as simple as a commodity PC) to (i) uncover the footprint of these CDN/CP companies, (ii) observe them clustering clients, and (iii) take snapshots of the user-to-server mappings. In addition, we point out potential implications that ECS can have on the cacheability of DNS responses by major DNS resolvers. We believe that the tools developed and the traces collected in this work, made available to the research community, shed light on the deployment and operation of CDNs given the central role they play in today's Internet. This work also highlights the need to increase the awareness among current and future ECS adopters about the consequences of enabling ECS.

Acknowledgments

We would like to thank our shepherd, Ethan Katz-Bassett, and the anonymous reviewers for their valuable feedback. This work was supported in part by the EU projects BigFoot (FP7-ICT-317858) and CHANGE (FP7-ICT-257422), EIT Knowledge and Innovation Communities program, and an IKY-DAAD award (54718944).

7. REFERENCES

[1] A Faster Internet Consortium. http://www.afasterinternet.com.

[2] Alexa top sites. http:///www.alexa.com/topsites.

[3] Google Global Cache. http://ggcadmin.google.com/ggc.

[4] Google Public DNS. https://developers.google.com/speed/public-dns.

[5] Mapping CDN domains. http://b4ldr.wordpress.com/2012/02/13/mapping-cdn-domains/.

[6] MaxMind, GeoIP databases. http://www.maxmind.com.

[7] OpenDNS. http://www.opendns.com.

[8] RIPE Routing Information Service. http://www.ripe.net/ris/.

[9] Routeviews Project, University of Oregon. http://www.routeviews.org/.

[10] V. K. Adhikari, S. Jain, Y. Chen, and Z. L. Zhang. Vivisecting YouTube: An Active Measurement Study. In *IEEE INFOCOM*, 2012.

[11] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig. Comparing DNS Resolvers in the Wild. In *ACM IMC*, 2010.

[12] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig. Web Content Cartography. In *ACM IMC*, 2011.

[13] M. Axelrod. The Value of Content Distribution Networks. AfNOG 9, 2008.

[14] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan. Mapping the Expansion of Google's Serving Infrastructure. In *ACM IMC*, 2013.

[15] N. Chatzis, G. Smaragdakis, J. Boettger, T. Krenc, and A. Feldmann. On the benefits of using a large IXP as an Internet vantage point. In *ACM IMC*, 2013.

[16] L. Colitti, S. H. Gunderson, E. Kline, and T. Refice. Evaluating IPv6 adoption in the Internet. In *PAM*, 2010.

[17] C. Contavalli, W. van der Gaast, S. Leach, and E. Lewis. Client subnet in DNS requests (IETF draft). http://tools.ietf.org/html/draft-vandergaast-edns-client-subnet-01.

[18] A. Dhamdhere and C. Dovrolis. Twelve Years in the Evolution of the Internet Ecosystem. *IEEE/ACM Trans. Networking*, 19(5), 2011.

[19] T. Flach, N. Dukkipati, A. Terzis, B. Raghavan, N. Cardwell, Y. Cheng, A. Jain, S. Hao, E. Katz-Bassett, and R. Govindan. Reducing Web Latency: the Virtue of Gentle Aggression. In *ACM SIGCOMM*, 2013.

[20] M. J. Freedman, M. Vutukuru, N. Feamster, and H. Balakrishnan. Geographic Locality of IP Prefixes. In *ACM IMC*, 2005.

[21] Google. What is 1e100.net? http://support.google.com/bin/answer.py?hl=en&answer=174717.

[22] C. Huang, A. Wang, J. Li, and K. Ross. Measuring and Evaluating Large-scale CDNs. In *ACM IMC*, 2008.

[23] B. Krishnamurthy and J. Wang. On Network-aware Clustering of Web Clients. In *ACM SIGCOMM*, 2001.

[24] R. Krishnan, H. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao. Moving Beyond End-to-end Path Information to Optimize CDN Performance. In *ACM IMC*, 2009.

[25] C. Labovitz, S. Lekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet Inter-Domain Traffic. In *ACM SIGCOMM*, 2010.

[26] Z. Mao, C. Cranor, F. Douglis, M. Rabinovich, O. Spatscheck, and J. Wang. A Precise and Efficient Evaluation of the Proximity Between Web Clients and Their Local DNS Servers. In *USENIX ATC*, 2002.

[27] E. Nygren, R. K. Sitaraman, and J. Sun. The Akamai Network: A Platform for High-performance Internet Applications. *SIGOPS Oper. Syst. Rev.*, 2010.

[28] 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 *ACM IMC*, 2012.

[29] Vern Paxson. Bro: A System for Detecting Network Intruders in Real-Time. *Com. Networks*, 31(23–24), 1999.

[30] I. Poese, B. Frank, B. Ager, G. Smaragdakis, and A. Feldmann. Improving Content Delivery using Provider-aided Distance Information. In *ACM IMC*, 2010.

[31] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye. IP Geolocation Databases: Unreliable? *ACM CCR*, 41(2), 2011.

[32] M A. Sanchez, 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 *USENIX/ACM NSDI*, 2013.

[33] M. Tariq, A. Zeitoun, V. Valancius, N. Feamster, and M. Ammar. Answering What-if Deployment and Configuration Questions with Wise. In *ACM SIGCOMM*, 2009.

[34] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci. Unconstrained Endpoint Profiling (Googling the Internet). In *ACM SIGCOMM*, 2008.

[35] S. Triukose, Z. Wen, and M. Rabinovich. Measuring a Commercial Content Delivery Network. In *WWW*, 2011.

[36] P. Vixie. Extension Mechanisms for DNS (EDNS0). RFC 2671 (Proposed Standard), 1999.

[37] P. Vixie. DNS Complexity. *ACM Queue*, 5(3):24–29, 2007.

[38] P. Vixie. What DNS is Not. *Comm. ACM*, 52(12):43–47, 2009.