Towards Identifying Networks with Internet Clients Using Public Data

Weifan Jiang*
Columbia University

Tao Luo*
Columbia University

Thomas Koch
Columbia University

Yunfan Zhang
Columbia University

Ethan Katz-Bassett
Columbia University

Matt Calder
Microsoft / Columbia University

ABSTRACT

Does an outage impact any users? Can a geolocation database known to be good at locating users and bad at infrastructure be trusted for a particular prefix? Is a content-heavy network likely to peer with a particular network? For these questions and many more, knowing which prefixes contain Internet users aids in interpreting Internet analysis. However, existing datasets of Internet activity are out of date, unvalidated, based on privileged data, or too coarse. As a step towards identifying which IP prefixes contain users, we present multiple novel techniques to identify which IP prefixes host web clients without relying on privileged data. Our techniques identify client activity in ASes responsible for 98.8% of Microsoft CDN traffic and in prefixes responsible for 95.2% of Microsoft CDN traffic. Less than 1% of prefixes identified by our technique as active do not contact Microsoft at all. We present measurements of Internet usage worldwide and sketch future directions for extending the techniques to measure relative activity levels across prefixes.

CCS CONCEPTS

• **Networks** → **Network measurement**.

KEYWORDS

Network Mapping, Replicable, Internet Measurement.

ACM Reference Format:
Weifan Jiang, Tao Luo, Thomas Koch, Yunfan Zhang, Ethan Katz-Bassett, and Matt Calder. 2021. Towards Identifying Networks with Internet Clients Using Public Data. In *ACM Internet Measurement Conference (IMC '21), November 2–4, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3487552.3487844

1 INTRODUCTION

Internet researchers would benefit from knowing which networks host users. This information is key to interpreting research results and operational aspects, and to weighting analysis. The impact of an outage, a slow route, or a network being added to a blocklist [4, 30] varies depending on whether the network has users. For a concrete illustration of how knowledge of which networks host users can impact analysis, consider 2015 analysis on Internet path lengths from Google [11]. Google peered directly with 41% of networks overall, but with 61% of networks hosting end users. Although most research historically considers all networks and routes equivalently (*e.g.,* consider how many CDFs across networks one has seen in IMC papers), this example shows that knowledge of where Internet users are can give a very different impression of even seemingly simple questions like "how long are paths from the cloud?' ' These differences can have important implications. When conducting research on how to optimize cloud performance for users, one might arrive at very different solutions if under the impression that most routes pass through intermediate networks (when considering all networks, only 41% do not) versus most routes being direct (when considering user networks, 61% are). As another example, geolocation databases like MaxMind are more accurate for end-user networks [16], and so knowing which networks host end-users provides insight into which geolocation results are trustworthy.

Recognizing how this knowledge can impact conclusions, some research uses various data sources of end-user activity, but the existing datasets are private, unvalidated, out of date, or opaque. The 2015 study mentioned above used a private CDN dataset [11] that cannot be shared and is now out of date. One study used IP prefixes seen in BitTorrent swarms [12] to indicate those prefixes hosted users and hence could be trusted for geolocation [7]. However, BitTorrent is no longer as popular. The ISI Internet Census data measures Internet activity in terms of responsive addresses [18], and so is not suited for inferring which networks have users.

APNIC's network population data [19] based on ad impressions has been used in studies [5, 6, 17, 22], but APNIC's methodology has a number of limitations. First, the approach has not been validated (to the best of our knowledge). Second, APNIC aggregates data at an AS granularity, which is too coarse-grained for use cases that require prefix-level information, like our geolocation example above, and threat-intelligence solutions. Third, the approach, which relies on placing Google Ads and collecting the IP addresses of users to which they are shown, is expensive. One study spent $5000 and only observed 8,589 distinct user IP addresses [27], a small fraction of the user prefixes (much less addresses) active in the world. This cost may make it prohibitive for other groups to set up similar measurements that address the limitations of APNIC and then keep the results updated over time. Fourth, network coverage is difficult to control due to the non-deterministic nature of the ad-bidding process, even when making use of targeted advertisements. Finally,

[*] These authors were primary contributors as Columbia students. They since started Ph.D. programs at Harvard University (Weifan) and University of Pennsylvania (Tao).

IMC '21, November 2–4, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-9129-0/21/11…$15.00
https://doi.org/10.1145/3487552.3487844

ad-based measurements raise ethical concerns due to the inability to gather informed user consent before launching the campaigns.

New techniques. We propose two new, replicable techniques that estimate which networks host active Internet users by identifying which contain web clients. Some clients may be bots and crawlers, but most networks with clients contain (human) users, and so our work is an important step towards the goal of identifying networks hosting users. Our two techniques fundamentally differ and have different tradeoffs.

Our first technique probes Google Public DNS caches to infer which prefixes have been querying for particular domains (§3.1). If a prefix queries for user-facing services, we infer that it likely hosts users. Our technique relies on the fact that Google Public DNS accepts queries that specify an EDNS0 Client Subnet (ECS) prefix, which means that Google only returns a cached DNS record if an IP address from that prefix previously queried for the same domain. This mechanism allows us to scan for activity over the entire IPv4 address space. Because Google Public DNS is anycast, it will only cache for an ECS prefix at the Google site where anycast routes the prefix's queries, and so we geo-distribute our probing to reach different sites. We are the first to probe Google Public DNS with ECS queries to discover Internet activity worldwide.

Our second technique crawls traces from the root DNS servers to find queries from Chromium-based browsers, since networks sourcing significant Chromium queries likely host users (§3.2). As we explain in more detail below, Chromium-based browsers use DNS probes to detect DNS interception [35]. Chromium sends these probes whenever it is launched or whenever the device IP or DNS configuration changes, and the domains probed are designed to not be cached, and so a count of these queries is a good approximation of Chromium usage in a network. Since Chromium-based browsers represent a large majority of browser usage, Chromium usage is a good approximation of web browser usage.

Validation of existing and new techniques. We compare the results from our techniques to each other, to APNIC network population estimates and to server-side logs of client IP addresses from Microsoft (§4). CDN client data offers the broadest view of Internet activity at the AS level, capturing 97% of all ASes seen using any method, but is not widely available to researchers. Our methods identify 29,973 ASes containing clients not seen by APNIC. Moreover, our results suggest that most prefixes in at least 15% of ASes do not contain clients. Hence, AS-level activity data such as APNIC is too coarse to understand activity at the IP-level. Finally, we compare to Microsoft CDN logs and find that the prefixes identified by our techniques as hosting web clients are responsible for 95.2% of queries to Microsoft, with 99.1% of them sending at least some queries to Microsoft. Hence, our methods can identify client prefix activity with precision and broad, global coverage rivaling a major cloud/CDN provider.

We conclude with a brief roadmap sketching future work for going from our lists of active client prefixes to relative activity levels across prefixes (§6).

2 GOALS

Building a map of Internet activity could come in different forms suited for answering different questions. For this preliminary investigation, we prioritize the following goals and discuss tradeoffs:

Focus on client activity. Measuring *client-driven* activity provides a basis to assess how Internet events and properties affect clients. By identifying clients of popular user-facing services, we seek to approximate the (human) user activity (so-called *eyeballs*) that is our ultimate goal. Users are clients, but we do not yet know how to filter out all non-human clients such as bots and crawlers.

Use replicable approaches. Prior work which used Internet activity measures to answer research questions often used privileged data sets that cannot be shared with the community [11, 21]. We want to base our map on datasets/techniques accessible to other researchers, and we are happy to share our data (except proprietary data we use for validation). Replicable approaches and public data help more researchers tackle difficult problems and allow future work to directly compare findings, but they may not be able to achieve the coverage of private datasets.

Provide fine-grained global coverage. We aim to cover as many client networks as possible, in terms of countries, ASes, and prefixes. A fine-grained map in time and network allows researchers to answer questions about time of day effects, and the effects of Internet events on specific geographic areas. This focus means we do not consider methods that are effective only in certain networks or regions, and we exclude methods that operate on the AS granularity, since it is too coarse (§4). We do not yet consider IPv6.

3 MEASUREMENT METHODS

3.1 Probing DNS caches for client activity

When users access websites they often issue DNS queries, populating caches in users' recursive resolvers. Our first approach, referred to as CACHE PROBING, is based on DNS cache snooping for activity from clients around the world. In DNS cache snooping, one sends a non-recursive DNS request to a recursive resolver. If the resolver returns a record, it must have had the record in cache (because the query was non-recursive), meaning a client of the recursive must have queried for the record (within the record's starting TTL).

One possible approach would be to try to cache snoop recursive resolvers in ISPs around the world [2, 7, 33]. However, the number of recursive resolvers that respond to queries from outside their ISPs has significantly reduced over time [25, 28], denying our goal of global coverage. One study overcame this limitation by probing misconfigured customer-premises equipment that would forward to recursive resolvers and return the result to the prober [26]. The study found such open forwarders in 4,905 ASes which, while substantial, is far below our goal of global coverage.

Instead, we leverage Google Public DNS, which has multiple advantageous properties. First, it is extremely popular as a recursive resolver, contributing 30-35% of all DNS queries to Microsoft Azure authoritative DNS servers in January 2019 [9]. Second, it supports EDNS0 Client Subnet (ECS), a DNS extension in which the recursive resolver includes a prefix of the querying client's IP address as part of the query [13], enabling client-specific responses [10]. An implication of Google supporting ECS is that it has to maintain separate cache entries per client prefix for domains that support ECS. Third, although normally the recursive resolver sets the ECS prefix based on the IP address of the querying client, if a client query instead includes an ECS prefix (not necessarily related to the client's IP address), Google Public DNS will use the supplied prefix in its queries instead.[1] We verified this behavior by sending queries for a domain for which we operate the authoritative resolver.

3.1.1 Methodology. We issue queries to Google Public DNS, varying the ECS prefix to scan the entire IPv4 space. A cache hit for ⟨`prefix, domain`⟩ suggests that a client in `prefix` issued a query for `domain`. Our non-recursive queries do not pollute the cache. In addition to our validation above, a recent study also verified that Google Public DNS does not query authoritative DNS servers on cache misses if the recursion desired flag is set to off [31].

Realizing our approach requires addressing a number of challenges. First, Google Public DNS has PoPs around the world, each with a set of independent caches, and so it is necessary to query the PoP that any clients in a prefix would use to determine whether they have accessed a domain. Google Public DNS relies on anycast to direct clients to a PoP, so we must issue queries from around the world to probe the behavior of clients around the world. Anycast does not always route clients to the nearest PoP [8, 21, 24], and so it can be challenging to know which PoP to query for a particular ECS prefix. Second, since records are only cached for the duration of their TTLs, which vary by record, which domains are queried for when can impact which ECS prefixes return cache hits.

Identifying candidate prefixes for ECS *queries.* Since ECS rarely uses prefix scopes more specific than /24 [34], we start with the set of 15,527,909 public /24 prefixes ($\approx 12M$ of which are currently routed). We use a technique to reduce probing overhead. Authoritative resolvers often return a less specific prefix scope for an ECS response than the request [34], meaning that the recursive resolver can cache and return the response for all addresses with the returned scope. If a returned scope is less specific than a /24, we need not query for other /24s within the less specific. While it is straightforward for us to use this observation to reduce probing if a Google Public DNS cache hit for a /24 query returns a less specific scope, a cache miss does not inform us whether or not other nearby /24 prefixes are worth querying. So, we first issue queries directly to the authoritative resolver to learn the scope it returns for the full address space, then use these returned scopes as our query scopes to Google Public DNS. For example, if a query to the authoritative for `a.b.c.0/24` returns a scope of `a.b.0.0/16`, we only issue queries to Google for `a.b.0.0/16`, saving the overhead of issuing probes for each /24 in the /16. Appendix A.2 validates this approach.

Geo-distributing measurements to probe caches worldwide. To probe caches worldwide, we need to issue queries from locations worldwide that anycast routes to different PoPs. The query `dig @8.8.8.8 o-o.myaddr.l.google.com -t TXT` returns which PoP is reached. We run our measurements from AWS and Vultr cloud VMs around the world to cover 22 Google Public DNS PoPs (red dots in Figure 1), out of 45 listed by Google. We tested all AWS regions and reached 16 PoPs, plus 6 more from Vultr. Our measurements include PoPs in the United States (seven states) and Canada (two provinces), Asia (five countries/regions), Europe (five countries), South America (two countries), and Australia. The 22 PoPs we cover account for 95% of Google Public DNS queries to Microsoft services. We verified with Google that some PoPs we did not reach are not active, and 18 of the 23 PoPs we do not cover do not issue any queries to Microsoft, suggesting they may be inactive. Figure 5 in Appendix A.1 shows the PoPs we do not measure.

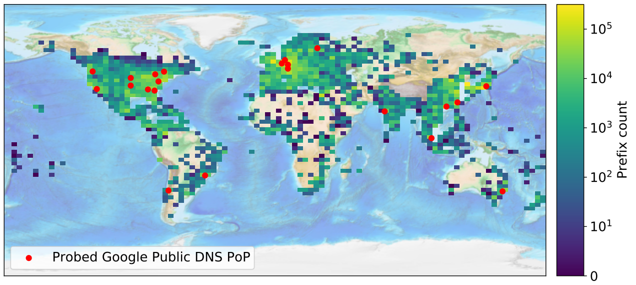

Figure 1: Density of active prefixes identified by our CACHE PROBING.

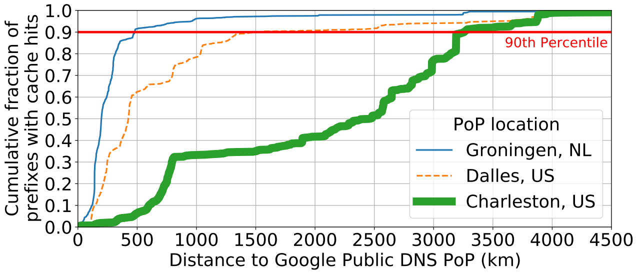

Figure 2: CDF of cache hits where the prefix is $\leq X$ km from the PoP.

To reduce measurements, we do not query for each prefix at each Google Public DNS PoP, instead querying for a prefix only at its most likely PoPs. Since anycast routes most clients to nearby PoPs for Google Public DNS [23], we use MaxMind to map each /24 prefix to a geolocation (which should be accurate enough for the user prefixes of interest to us). We first query each PoP with 78,637 prefixes selected randomly out of all the public IPv4 address space for which MaxMind indicates an error radius smaller than 200 km. For each PoP, we then determine the radius that contains 90% of those prefixes that return a cache hit for at least one of the four most popular domains in the Alexa top global sites list that both support ECS and have TTLs greater than one minute. We consider this radius to be a likely *service radius* for that PoP. In the rest of our measurements, we query a PoP only for prefixes that MaxMind places as possibly within the PoP's service radius (combining the MaxMind location and error radius for the prefix). Figure 2 justifies this approach—for three PoPs with diverse geographies, the service radius ranges from 478 km to 3273 km. Using per-PoP service radii results in an average of 2.4 million /24 prefixes to probe at each PoP, compared to an average 4.4 million if we used the maximum service radius of 5,524 km (used for Zurich) for all PoPs.

Probing details. We select the four most popular domains from the Alexa top sites global list that both support ECS and have TTLs greater than one minute (as of 9/22/2021): `www.google.com` (rank 1), `www.youtube.com` (rank 2), `facebook.com` (rank 7), and

[1] Other major public resolvers such as Cloudflare's do not display this behavior.

www.wikipedia.org (rank 13). We also include one popular domain hosted by Microsoft Azure Traffic Manager that supports ECS and has a TTL of 5 minutes, which we will use to validate our methodology. We issued cache probes for 120 hours at a rate of 50 prefixes per second per domain at each PoP, looping over the list of assigned prefixes continuously. Since Google Public DNS employs multiple independent cache pools at each PoP [31], we issue 5 redundant queries for each ⟨PoP, prefix, domain⟩ combination to increase the likelihood that our queries can cover multiple caches. We queried Google Public DNS through DNS over TCP instead of UDP, as probing the same domains repeatedly using UDP triggers a rate limit much lower than the normal 1,500 QPS limit. We consider a prefix active if Google Public DNS returns a cache hit for any domain indicating the prefix with return scope > 0 for an ECS query (a return scope of 0 indicates the cache entry was for the whole address space rather than for a particular prefix). In the future (§6), we will assess how queries for different domains with varying popularity, TTLs, and user bases can be combined to obtain a rich picture of the types and relative activity levels for a network.

3.1.2 Strengths and Limitations. Google Public DNS CACHE PROBING can be replicated by anyone, without requiring any privileged access or data. It directly measures (likely) active client prefixes, rather than measuring activity from recursive resolvers or another proxy of client activity. It also lends itself to developing rich signals by combining observations across time and domains. However, the approach has limitations. It measures active use of Google Public DNS (and not of other recursive resolvers), which is popular (§3.1) but may have skewed adoption along various dimensions. DNS's use of recursive resolvers and caching also introduces complexities in comparing activity levels across ECS prefixes. It is likely impossible to quantify how much a DNS record was used from cache within its TTL, and cross-prefix comparisons are tricky because of differences in addressing (including NAT) within an ECS prefix.

3.2 Crawling DNS for Chromium Queries

We call our second approach DNS LOGS. We look for DNS queries matching a signature of the Chromium web browser codebase, which is part of browsers including Chrome, Microsoft Edge, Brave, and Opera. The number of Chromium DNS queries from a prefix is intuitively an indicator of the level of client activity.

3.2.1 Methodology. Chromium detects DNS interception by querying for random strings of 7-15 lowercase letters [35]. Chromium sends queries when the browser starts, and when the device's IP address or DNS configuration changes. Because these queries often have no valid TLD appended (e.g., COM), they should not result in cache hits at recursive resolvers, so the queries go to a DNS root server [35]. To separate Chromium queries from others (*e.g.*, "sdhfjssf" vs. "columbia"), we use the heuristic that randomly generated strings likely have few collisions. Using empirical simulations, we found Chromium queries would collide fewer than 7 times per day across all roots with 99% probability.

We look for queries matching this pattern queried less often than our daily threshold in the DITL traces, 2 days of traces of queries to most root DNS servers [15]. The queries in the traces contain the IP address of the querier, which is generally the recursive resolver used by the Chromium client. We consider a matching query to be a strong indicator that a recursive resolver with that IP address is used by a user of a Chromium browser. We process J, H, M, A, K and D root, the roots that offer un-anonymized, complete traces in the most recent available (2020) DITL.

3.2.2 Strengths and Limitations.

A direct, precise signal with global coverage. Using Chromium queries as a proxy for client activity provides per-resolver counts proportional to the number of clients, assuming counts over large populations are proportional to the number of clients. Counting Chromium queries offers truly global coverage—if a recursive resolver forwards Chromium queries to the roots, they are in DITL. Most major browsers use Chromium, and the market share is growing. Counting Chromium queries can be done by many researchers (through access to DNS-OARC, or via collaboration with a root deployment, some of which are hosted by academic institutions).

But . . . it's not perfect. First, IP addresses seen in root DNS packet traces are of recursive resolvers, and so Chromium queries provide a signal of client activity at the recursive resolver level rather than at the prefix or AS level.

Second, user activity and the presence of Chromium queries are not perfectly correlated. The analysis excludes users of Safari and Firefox. Chromium queries are (only) executed each time the browser starts up, and each time the system's IP address or DNS configuration changes [35]. Also, DITL traces are only available yearly and do not contain all root letters, so time-based analysis is not possible from DITL alone. Moreover, the implementation of this feature is subject to change. Since Chromium queries cause a considerable load on the root DNS, the Chromium team has shown interest in reducing the number of DNS queries going to the root DNS [36]. We verified in September 2021 with B root that a few percent of all B root queries are Chromium queries, although that number is only 30% of what it was in 2020.

4 VALIDATION & CROSS-COMPARISON

Datasets. We compare client activity indicators obtained using CACHE PROBING and DNS LOGS to measures of activity used in prior work. First, we compare our results to APNIC user estimates (APNIC), which use a heuristic based on Google Ad volumes to generate user population estimates by AS. APNIC is publicly available, so it is useful to see how these new methodologies augment existing, widely accessible methods of estimating activity. Second, we compare three private datasets which contain measures of client activity for two popular Microsoft Azure services: CDN and DNS Traffic Manager. These services are used by billions of users in tens of thousands of ASes and hundreds of countries/regions daily. The first measure (MICROSOFT CLIENTS) is proportional to the number of times clients access the CDN, aggregated by client IP address. The second (MICROSOFT RESOLVERS) is a count of client IP addresses that the CDN observes using each recursive resolver, aggregated by recursive resolver IP address. The third (CLOUD ECS PREFIXES) is the set of ECS prefixes observed in DNS queries for authoritative records of Traffic Manager, Azure's DNS-based load balancing for cloud tenants. We aggregate the data by prefix and by AS.

DNS activity is a good proxy for web client activity. To demonstrate that DNS-based techniques like ours can be a good proxy for identifying Internet clients, we compare a full day of (1) /24 prefixes from Microsoft clients (no ECS) with (2) ECS prefixes from cloud ECS prefixes. The CDN sees HTTP requests from prefixes including ones responsible for 97.2% of the DNS queries. Prefixes seen in the ECS queries are responsible for 92% of the HTTP requests to the CDN. This large overlap shows that client prefixes with DNS activity usually have HTTP(S) activity, and our techniques will be able to identify most *prefixes with web activity* (the goal) if they are able to identify most *prefixes with DNS activity* (what they measure). Not all prefixes that query CDNs are human users: both services see some bot, crawler, and machine-to-machine traffic.

Cache probing recovers most DNS activity. We compare the results of our cache probing for a popular Microsoft Azure domain with ground truth ECS data observed at its authoritative resolver. Our cache probing includes 91% of the ground truth ECS/24 prefixes, showing that our approach can uncover the vast majority of a service's client population that is using Google Public DNS.

AS-level results. We consider the overlap in ASes detected as hosting clients by our two techniques and the ones we compare to. Although our methods identify activity at finer granularities, we start with AS-level comparisons to compare with APNIC. Table 3 in Appendix B.1 presents pairwise comparisons. In total, 66,804 ASes were in at least one dataset, with 64,766 of those (97%) being in the Microsoft clients. The APNIC dataset, despite being widely used, misses 64% of the ASes observed as hosting Microsoft clients. Our techniques perform better, missing only 40.1% (DNS logs) and 44.5% (cache probing), and recovering 74.2% and 81.9% of the ASes observed by APNIC. DNS logs detects about the same number of ASes as cache probing (39,652 vs 36,989), and the overlap between them is fairly low—combined, they detect 51,859 ASes. The low overlap could result from DNS logs measuring some ASes that host clients' recursive resolvers but not clients. The low overlap means that combining our datasets yields more overlap with others. For example, cache probing ∪ DNS logs observes 21,866 ASes in APNIC (93.8%) and 50,006 ASes in Microsoft clients (77.2%). To help understand the ASes our techniques detect as hosting web clients but that APNIC does not consider as hosting customers, we consider what types of ASes they are, according to ASdb [38]. Of all 29,973 ASes detected by our methods but absent in APNIC, ASdb categorizes 27,773 (92.7%). Of these, 10,998 (39.5%) are Internet Service Providers (ISPs). Outside the ISPs, 4,823 (17.4%) are hosting/cloud providers (which may reflect non-human web clients), and 1,723 (6.2%) are schools, which likely host human users.

The ASes we miss are generally small. ASes that at least one of our techniques identifies as hosting clients account for 98.8% of the Microsoft clients queries (compared to 92% for APNIC). Table 4 in Appendix B.2 presents pairwise comparisons by volume. Although our DNS logs technique only identifies 74.2% of the APNIC ASes, those ASes account for 97.6% of the world's Internet population, per APNIC estimates. ASes that include prefixes that cache probing detected as active also account for 97.6% of APNIC's Internet population. Figure 3 breaks this analysis down per country. In most countries, cache probing uncovers client activity in ASes that APNIC identifies as hosting all or almost all of the Internet users. Many of the larger countries (in terms of # of users) where cache probing coverage is worse are in South America, even though our probes covered all Google PoPs in South America and most in the southern United Stages (Appendix A.1).

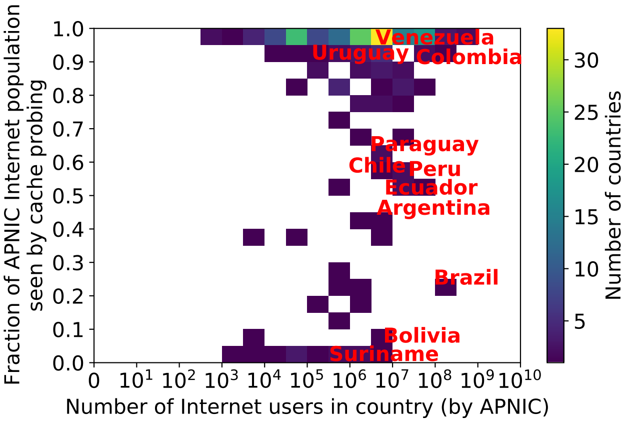

Figure 3: Fraction of a country's Internet users (according to APNIC) in ASes where our cache probing technique identified client activity. We identified most eyeballs in most countries, including ≈ 100% in the U.S., 99% in India, and 98% in China.

Despite these gaps, our cache probing results in global coverage: Figure 1 plots the MaxMind geolocations of prefixes where cache probing detects activity. For each prefix with return scope larger than /24, we make the simplifying assumption that all its /24 subprefixes are active. For (rare) return scopes smaller than /24, we assume the entire /24 prefix it belongs to is active. Our measurements infer more activity in some regions than others, e.g. Europe is more active than China, although we cannot easily differentiate how much this is a result of differences in prefix allocation policies, Google Public DNS use, popularity of the domains we probe, or coverage of our vantage points. Within a region, the distribution of active prefixes often roughly follows the distribution of population. For example, activity in the US and Brazil is densest near coasts.

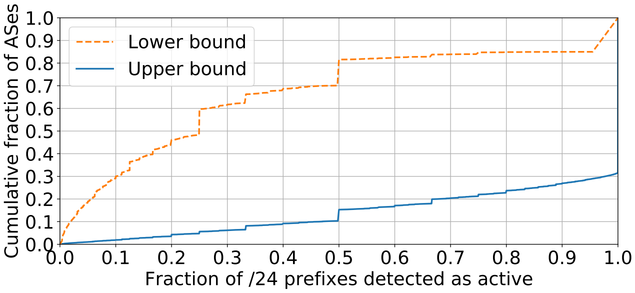

Figure 4: Fraction of AS's prefixes detected as active by cache probing.

Prefix-level measurements reveal variations within and across ASes. Figure 4 depicts the fraction of /24 prefixes announced by each AS that our cache probing technique detects as active (out of all the AS announces [1]). When Google Public DNS returns a cache hit for a prefix with scope bigger than /24, we know at least one /24 in the prefix has client activity, but we cannot infer exactly which or how many. So, we estimate upper and lower bounds. The `lower bound`

	cache probing	DNS logs	cache probing ∪ DNS logs	Microsoft clients	Microsoft resolvers
cache probing	9712.2K (100.0%)	650.5K (6.7%)	9712.2K (100.0%)	6614.4K (68.1%)	932.6K (9.6%)
DNS logs	650.5K (94.0%)	692.2K (100.0%)	692.2K (100.0%)	661.2K (95.5%)	419.8K (60.6%)
cache probing ∪ DNS logs	9712.2K (99.6%)	692.2K (7.1%)	9753.9K (100.0%)	6647.8K (68.2%)	954.1K (9.8%)
Microsoft clients	6614.4K (74.7%)	661.2K (7.5%)	6647.8K (75.1%)	8849.9K (100.0%)	940.5K (10.6%)
Microsoft resolvers	932.6K (96.4%)	419.8K (43.4%)	954.1K (98.6%)	940.5K (97.2%)	967.7K (100.0%)

Table 1: Each entry shows the size of intersection of the set of /24 prefixes observed in the two datasets. In parentheses is the percent of the row dataset also observed in the column dataset. The diagonal gives the size of each dataset.

line is the minimum activity consistent with our measurements, a single active /24 per non-overlapping prefix with a cache hit. The `upper bound` is the maximum amount of activity, where we assume activity exists within all /24 prefixes in a prefix with a cache hit.

The results vary widely across ASes—some have only a small fraction of prefixes active, while some have most/all prefixes active. This result shows that APNIC's per-AS granularity is too coarse for use cases that need to understand activity at the IP-level. Our techniques help fill that gap. The result also shows that our current technique allows a wide range of interpretations—the median percentage of active prefixes per AS could be anywhere between 25% and 100%—suggesting room to refine our techniques.

Prefix-level analysis. Table 1 shows the overlap in /24 prefixes found to be hosting clients using our methods and in privileged Microsoft traces. We upper bound our cache probing coverage by assuming that, if it found a prefix to contain clients, all /24 prefixes within that (possibly larger) prefix contain clients. Our methods capture prefixes that include 9.75M /24 prefixes in total, including 75.1% of /24 prefixes seen by Microsoft clients, and those /24 prefixes represent 95.2% of Microsoft clients volume. Although our DNS logs method only finds 692.2K prefixes, 95.5% of these prefixes were also in Microsoft clients, suggesting that these prefixes do host clients (high precision). However, only 6.6M (74.7%) of the /24 prefixes included in cache probing were also seen by Microsoft clients, suggesting that our upper bound on cache probing is too generous. Our future work will try to improve the precision. Still, 99.1% of prefixes returned as the scope for our cache probing queries contain at least one /24 in Microsoft clients, so our cache probing method has few false positives.

5 RELATED WORK

Our cache snooping approach is inspired by earlier approaches that either cache snoop Google Public DNS or use ECS to simulate access to vantage points worldwide, although we are not aware of any earlier work that uses these approaches for our goal of understanding global Internet usage. Two studies used ECS scans of the entire IPv4 space to uncover the client-to-server mapping for CDNs [7, 34]. That work did not use Google Public DNS and was interested in where a CDN would direct an ECS prefix, not in whether actual clients from that prefix had queried for popular domains. A recent study demonstrated how cache snooping on Google Public DNS (and other public DNS services) can be used to estimate the usage of rare domains [31]. The particular approach in that study did not achieve the global coverage that is our goal, as it did not use ECS and limited itself to measurements from 43 United States vantage points. So, its cache snooping was limited to results for ≤ 43 client prefixes to 7 Google PoPs. Other work investigated the ECS behavior of recursive resolvers [3].

Previous work estimated the popularity of websites by using open resolvers [29, 37]. Our work instead uses popular domains to identify Internet clients. A recent study analyzes ECS queries from Google Public DNS as seen in traces at an authoritative resolver to understand usage of Google Public DNS [14]. The study reveals interesting aspects of Google Public DNS adoption, although the queries do not provide an unbiased view of global usage because the resolver is mainly authoritative for Dutch domain names. A study measured connection logs from a CDN to estimate Internet activity at the IP address granularity [32]. They use activity metrics as a proxy to assess IPv4 address scarcity and characterize Internet growth. We used similar logs for our Microsoft clients dataset.

6 CONCLUSIONS AND FUTURE WORK

Measuring user Internet activity would provide a rich source of data to help answer research questions. We present preliminary work on new techniques for measuring client activity—cache probing Google Public DNS and crawling root DNS traces. Our techniques have global coverage, can be replicated by other researchers without privileged data, and provide fine-grained *client* activity data (at the resolver or prefix level). Going forward, we will explore how to infer which prefixes with *client* activity likely include (human) *user* activity, using signals such as activity across a range of user-facing services, patterns over time (*e.g.,* diurnal patterns), and consistency across methods (*e.g.,* using Chromium *and* querying popular services).

Future work will focus on obtaining a relative activity ranking across prefixes, similar to how APNIC lists ASes by Internet user population [19]. We envision two major directions. First, we want to combine the information from our two techniques, which is difficult since cache probing measures client prefix activity whereas DNS logs measures recursive resolver prefix activity. One possibility is to join on geolocation—since users are often physically close to and in the same AS as their recursive resolver [10], we can estimate activity at the ⟨`region, AS`⟩ granularity and associate that activity with active prefixes in that ⟨`region, AS`⟩. Second, the cache probing technique yields different results depending on where and when it is run, and on which domains. We are developing techniques to estimate a prefix's cache hit rates over time and across domains, as a step towards a relative ranking of prefix activity levels. Our contemporaneous workshop paper presents initial validation of this approach to measuring relative prefix activity levels as part of a vision for an Internet traffic map [20].

Future work can investigate which methodologies for measuring activity—including both ours and others—are best for particular questions, to bring us closer to understanding how users are affected by, interact with, and are a part of different areas of the Internet.

Acknowledgements. This paper has been partially funded by NSF grant CNS-1836872. We would like to thank our shepherd Philipp Richter and the anonymous reviewers for their insightful comments, Google Public DNS operators for information about their system, and John Heidemann for validating some of our root DNS analysis. We acknowledge the Trufflehunter authors [31], especially Audrey Randall and Aaron Schulman, for inspiring aspects of our approach and for answering our questions about their work.

REFERENCES

[1] Routeviews prefix to AS mappings dataset for IPv4 and IPv6, 2021. URL https://www.caida.org/catalog/datasets/routeviews-prefix2as/.

[2] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis. A multifaceted approach to understanding the botnet phenomenon. In *ACM IMC*, 2006.

[3] Rami Al-Dalky, Michael Rabinovich, and Kyle Schomp. A look at the ECS behavior of DNS resolvers. In *ACM IMC*, 2019.

[4] Pelayo Vallina Alvaro Feal, Julien Gamba, Sergio Pastrana, Antonio Nappa, Oliver Hohlfeld, Narseo Vallina-Rodriguez, and Juan Tapiador. Blocklist babel: On the transparency and dynamics of open source blocklisting. *IEEE Transactions on Network and Service Management*, April 2021.

[5] Todd Arnold, Ege Gürmeriçliler, Georgia Essig, Arpit Gupta, Matt Calder, Vasileios Giotsas, and Ethan Katz-Bassett. (How much) does a private WAN improve cloud performance? In *IEEE INFOCOM*, 2020.

[6] Todd Arnold, Jia He, Weifan Jiang, Matt Calder, Italo Cunha, Vasileios Giotsas, and Ethan Katz-Bassett. Cloud provider connectivity in the flat Internet. In *ACM IMC*, 2020.

[7] Matt Calder, Xun Fan, Zi Hu, Ethan Katz-Bassett, John Heidemann, and Ramesh Govindan. Mapping the expansion of Google's serving infrastructure. In *ACM IMC*, 2013.

[8] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, and Jitendra Padhye. Analyzing the performance of an anycast CDN. In *ACM IMC*, 2015.

[9] Matt Calder, Xun Fan, and Liang Zhu. A cloud provider's view of EDNS client-subnet adoption. In *TMA*, 2019.

[10] Fangfei Chen, Ramesh K Sitaraman, and Marcelo Torres. End-user mapping: Next generation request routing for content delivery. In *ACM SIGCOMM*, 2015.

[11] Yi-Ching Chiu, Brandon Schlinker, Abhishek Balaji Radhakrishnan, Ethan Katz-Bassett, and Ramesh Govindan. Are we one hop away from a better Internet? In *ACM IMC*, 2015.

[12] David R Choffnes and Fabián E Bustamante. Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems. In *ACM SIGCOMM*, 2008.

[13] Carlo Contavalli, Wilmer van der Gaast, David C. Lawrence, and Warren Kumari. Client subnet in DNS queries. RFC 7871, RFC Editor, May 2016. URL https://datatracker.ietf.org/doc/html/rfc7871.

[14] Wouter de Vries, Roland van Rijswijk-Deij, Pieter-Tjerk de Boer, and Aiko Pras. Passive observations of a large DNS service: 2.5 years in the life of Google. In *TMA*, 2018.

[15] DITL. DITL traces and analysis | DNS-OARC, 2020. URL https://www.dns-oarc.net/oarc/data/ditl.

[16] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. A look at router geolocation in public and commercial databases. In *ACM IMC*, 2017.

[17] Petros Gigis, Vasileios Kotronis, Emile Aben, Stephen D Strowes, and Xenofontas Dimitropoulos. Characterizing user-to-user connectivity with RIPE Atlas. In *ACM/IRTF ANRW*, 2017.

[18] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. Census and survey of the visible Internet. In *ACM IMC*, 2008.

[19] Geoff Huston. How big is that network?, 2014. URL https://labs.apnic.net/?p=526.

[20] Thomas Koch, Weifan Jiang, Tao Luo, Petros Gigis, Ethan Katz-Bassett, Matt Calder, Georgios Smaragdakis, Lefteris Manassakis, Emile Aben, and Narseo Vallina-Rodriguez. Towards a traffic map of the Internet. In *ACM HotNets*, 2021.

[21] Thomas Koch, Ke Li, Calvin Ardi, Ethan Katz-Bassett, Matt Calder, and John Heidemann. Anycast in context: A tale of two systems. In *ACM SIGCOMM*, 2021.

[22] Vasileios Kotronis, George Nomikos, Lefteris Manassakis, Dimitris Mavrommatis, and Xenofontas Dimitropoulos. Shortcuts through colocation facilities. In *ACM IMC*, 2017.

[23] Zhihao Li. *Diagnosing and Improving the Performance of Internet Anycast*. PhD thesis, University of Maryland, 2019.

[24] Zhihao Li, Dave Levin, Neil Spring, and Bobby Bhattacharjee. Internet anycast: performance, problems, & potential. In *ACM SIGCOMM*, 2018.

[25] Jared Mauch. Open resolver project revisited. In *OARC 31*, 2019.

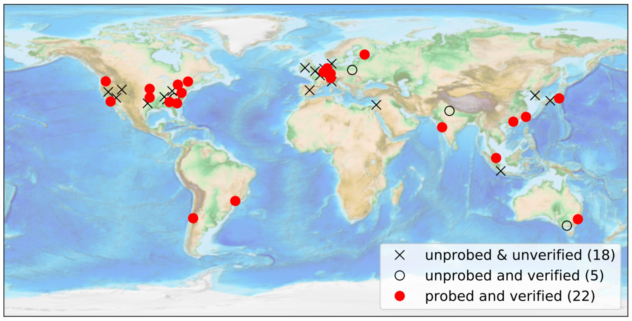
× unprobed & unverified (18)
○ unprobed and verified (5)
● probed and verified (22)

Figure 5: Google Public DNS PoPs that our measurements do and do not probe.

[26] Arian Akhavan Niaki, William Marczak, Sahand Farhoodi, Andrew McGregor, Phillipa Gill, and Nicholas Weaver. Cache Me Outside: A New Look at DNS Cache Probing. In *PAM*, 2021.

[27] Mark O'Neill, Scott Ruoti, Kent Seamons, and Daniel Zappala. TLS proxies: Friend or foe? In *ACM IMC*, 2016.

[28] Jeman Park, Aminollah Khormali, Manar Mohaisen, and Aziz Mohaisen. Where are you taking me? Behavioral analysis of open DNS resolvers. In *IEEE/IFIP DSN*, 2019.

[29] Moheeb Abu Rajab, Fabian Monrose, Andreas Terzis, and Niels Provos. Peeking through the cloud: DNS-based estimation and its applications. In *ACNS*, 2008.

[30] Sivaramakrishnan Ramanathan, Anushah Hossain, Jelena Mirkovic, Minlan Yu, and Sadia Afroz. Quantifying the impact of blocklisting in the age of address reuse. In *ACM IMC*, 2020.

[31] Audrey Randall, Enze Liu, Gautam Akiwate, Ramakrishna Padmanabhan, Geoffrey M Voelker, Stefan Savage, and Aaron Schulman. Trufflehunter: Cache Snooping Rare Domains at Large Public DNS Resolvers. In *ACM IMC*, 2020.

[32] Philipp Richter, Georgios Smaragdakis, David Plonka, and Arthur Berger. Beyond counting: new perspectives on the active IPv4 address space. In *ACM IMC*, 2016.

[33] Will Scott, Thomas Anderson, Tadayoshi Kohno, and Arvind Krishnamurthy. Satellite: Joint analysis of CDNs and network-level interference. In *USENIX ATC*, 2016.

[34] Florian Streibelt, Jan Böttger, Nikolaos Chatzis, Georgios Smaragdakis, and Anja Feldmann. Exploring EDNS-client-subnet adopters in your free time. In *ACM IMC*, 2013.

[35] Matthew Thomas. Chromium's impact on root DNS traffic, 2020. URL https://blog.apnic.net/2020/08/21/chromiums-impact-on-root-dns-traffic/.

[36] Duane Wessels. How Chromium reduced root DNS traffic, 2021. URL https://blog.apnic.net/2021/02/04/how-chromium-reduces-root-dns-traffic/.

[37] Craig E Wills, Mikhail Mikhailov, and Hao Shang. Inferring relative popularity of Internet applications by actively querying DNS caches. In *ACM IMC*, 2003.

[38] Maya Ziv, Liz Izhikevich, Kimberly Ruth, Katherine Izhikevich, and Zakir Durumeric. ASdb: A system for classifying owners of autonomous systems. In *ACM IMC*, 2021.

A ADDITIONAL VALIDATION OF METHODOLOGIES

A.1 Coverage of Google Public DNS PoPs

As discussed in Section 3.1.1 and shown in Figure 1, our current vantage points suffice to probe 22 Google Public DNS PoPs. Figure 5 shows the locations of those PoPs and the ones we currently do not probe. For the 23 we do not probe, 5 show up as recursive resolvers in our Microsoft resolvers dataset (`unprobed and verified` in Figure 5), indicating that they actively serve clients. The other 18 do not query Microsoft Azure's DNS Traffic Manager, (the source of Microsoft resolvers), during the week of 2021/09/20, suggesting that they are likely not actively serving users or announcing the anycast route, and explaining why we were unable to reach them from any cloud data center we tried. Additionally, the 5 `unprobed and verified` sites account for only

5% of the Google Public DNS queries seen by Microsoft, with the 22 sites in `probed and verified` that provide the results in our paper account for 95%. This result suggests that the `unprobed and verified` receive little traffic, so likely would result in relatively fewer ECS cache hits for CACHE PROBING and may have fewer anycast routes reaching them, explaining why the cloud providers we have tried do not reach them.

A.2 Validating ECS query scopes

Section 3.1.1 describes how we set the prefix scopes we use for ECS queries to reduce probing overhead, and this section demonstrates that the response scopes assigned by the authoritative name servers are stable, and therefore the reduction in probes does not significantly alter our cache probing results or the granularity of cache hit prefixes. If the response scopes assigned by the authoritatives are stable over time, our less specific query scopes will match the response scopes cached by Google Public DNS. In this case, our probes that use the less specific query scope would generate the same result as if we had probed for the /24 prefix.

Across all cache hits, 90% of the responses have the same scope as the query, suggesting that our approach to reducing probing overhead has little impact on our overall results. The results are shown in Table 2, with 97% of the hits having response scopes and queries scopes that differ by at most 2. Only 1% of the hits have response scopes and queries scopes that differ by more than 4. This result demonstrates that the vast majority of response scopes are stable over the period in which we probed Google Public DNS, and our query scope reduction technique significantly reduces probing overhead without having a large impact on results.

B ADDITIONAL MEASUREMENT RESULTS

B.1 Dataset overlap by AS

Expanding the results from Section 4, Table 3 shows the overlap in ASes detected as hosting users by five techniques: our CACHE PROBING for per-ECS cache hits, our crawls of root DNS LOGS for Chromium queries, plus APNIC, MICROSOFT CLIENTS, and MICROSOFT RESOLVERS. Each entry shows the size of intersection of the set of ASes observed in the two datasets. The parentheses in a column indicates the coverage of that technique relative to the others.

B.2 Dataset overlap by activity level

Expanding the results from Section 4, Table 4 captures how much of each dataset's activity volumes are from the ASes that overlap other datasets (*i.e.,* the ASes from Table 3). Columns CACHE PROBING and CACHE PROBING ∪ DNS LOGS do not have a row since they do not have a measure of volume. The DNS LOGS column captures how much activity from each other dataset is from ASes inferred as active by our DNS LOGS technique. Although our DNS LOGS technique only uncovers 74.2% of the APNIC ASes, those ASes account for 97.6% of the world's Internet population, according to APNIC estimates. The ASes we identify as hosting clients account for 98.8% of the MICROSOFT CLIENTS queries and 100.0% of the MICROSOFT RESOLVERS client IP addresses (compared to 92.0% and 95.7% for APNIC).

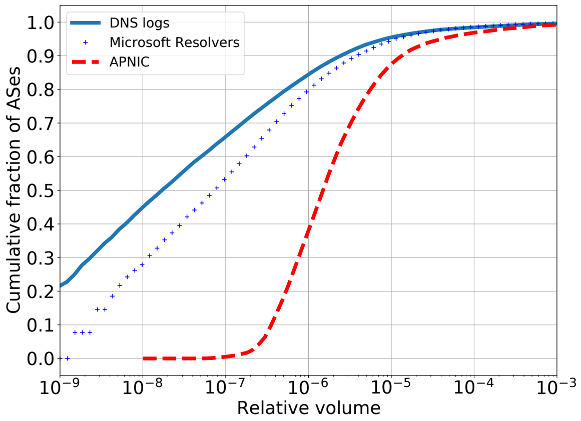

Figure 6: Distribution of relative volume among ASes for three methods of measuring AS client activity. DNS LOGS results in similar client activity estimates to MICROSOFT RESOLVERS, which makes sense since they both rely on signals from recursive resolvers.

B.3 Comparing relative activity by AS

Although we leave relative activity estimates for client prefixes largely as future work (§6), we present preliminary analysis of one activity estimate—DNS LOGS, since DNS LOGS provides the *number* of Chromium queries as a direct per-resolver relative activity measure. We compare this relative measure to two other relative measures—the number of IP addresses using a recursive resolver (MICROSOFT RESOLVERS) and the estimated Internet population of an AS (APNIC). In our contemporaneous work [20], we present preliminary methods to adding activity estimates to our CACHE PROBING methodology.

As an aggregated view of all activity estimates, Figure 6 shows the distribution across ASes of estimates of client activity using three methods (DNS LOGS, APNIC, MICROSOFT RESOLVERS). DNS LOGS and MICROSOFT RESOLVERS have similar relative distributions, whereas APNIC tends to have far fewer ASes with smaller numbers of Internet users.

To supplement the aggregate view in Figure 6, Figure 7 shows the difference in an AS's relative activity levels as estimated by different approaches. The datasets disagree by at most 1e-5 for 90% of ASes, suggesting all three datasets would roughly group all ASes into similar levels of client activity. Again, we see DNS LOGS is the most similar to MICROSOFT RESOLVERS, which makes sense since DNS LOGS measures activity by recursive. In particular, we expect APNIC estimates to differ from DNS LOGS and MICROSOFT RESOLVERS when clients use a resolver outside their AS. For example, Google public DNS clients are likely not in the same AS as that service. In this case, both DNS LOGS (.5%) and MICROSOFT RESOLVERS (20%) would assign a higher weight to Google's AS, whereas APNIC (9e-6%) would likely distribute this weight over the ASes whose users use Google public DNS.

Scope difference	Google	YouTube	Facebook	Wikipedia	Microsoft CDN	Overall
Exact match	297,891 (89%)	187,204 (88%)	153,373 (89%)	62,928 (96%)	128,519 (94%)	829,915 (90%)
Within 2	327,159 (97%)	206,006 (96%)	168,539 (98%)	63,435 (97%)	129,423 (94%)	894,562 (97%)
Within 4	334,088 (99%)	211,575 (99%)	170,581 (99%)	63,735 (97%)	130,990 (96%)	910,969 (99%)

Table 2: The number of Google Public DNS cache hit prefixes that have response scopes and query scopes that are equal, or that differ by at most 2 or 4. Overall, for 90% of the cache hits, the query scopes match the return scopes exactly. Only 1% of the cache hits have query scopes and return scopes that differ by more than 4.

	cache probing	DNS logs	cache probing ∪ DNS logs	APNIC	Microsoft clients	Microsoft resolvers
cache probing	36,989 (100.0%)	24,782 (67.0%)	36,989 (100.0%)	19,118 (51.7%)	35,915 (97.1%)	25,602 (69.2%)
DNS logs	24,782 (62.5%)	39,652 (100.0%)	39,652 (100.0%)	17,323 (43.7%)	38,787 (97.8%)	34,573 (87.2%)
cache probing ∪ DNS logs	36,989 (71.3%)	39,652 (76.5%)	51,859 (100.0%)	21,886 (42.2%)	50,006 (96.4%)	37,500 (72.3%)
APNIC	19,118 (81.9%)	17,323 (74.2%)	21,886 (93.8%)	23,344 (100.0%)	23,264 (99.7%)	18,121 (77.6%)
Microsoft clients	35,915 (55.5%)	38,787 (59.9%)	50,006 (77.2%)	23,264 (35.9%)	64,766 (100.0%)	39,825 (61.5%)
Microsoft resolvers	25,602 (63.4%)	34,573 (85.6%)	37,500 (92.8%)	18,121 (44.9%)	39,825 (98.6%)	40,394 (100.0%)

Table 3: Each entry shows the size of intersection of the set of ASes observed in the two datasets. In parentheses is the percent of the row dataset also observed in the column dataset. The diagonal gives the size of each dataset.

	cache probing	DNS logs	cache probing ∪ DNS logs	APNIC	Microsoft clients	Microsoft resolvers
DNS logs	98.4%	100.0%	100.0%	96.3%	99.8%	100.0%
APNIC	97.6%	97.6%	99.4%	100.0%	100.0%	98.3%
Microsoft clients	94.9%	97.4%	98.8%	92%	100.0%	96.7%
Microsoft resolvers	97.7%	99.9%	100.0%	95.7%	99.3%	100%

Table 4: Each entry gives the percent of total activity volume in the row dataset represented by ASes that also appear in the column dataset (Table 3 gives the number of ASes in these intersections).

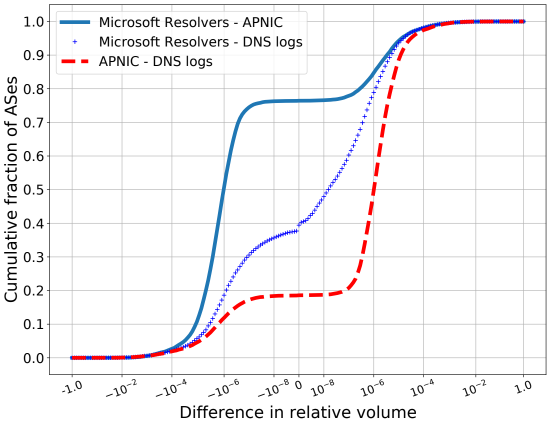

Figure 7: Difference in activity metrics for each AS between each of the three methods of measuring AS client activity. DNS logs results in similar client activity estimates to Microsoft resolvers, which makes sense since they both rely on an intermediate signal from a recursive resolver.

B.4 cache probing Results by domain

When probing Google Public DNS, we probe for several domains that users frequently issue DNS requests for. For our cache probing measurements, we selected the four highest-ranked domains in the Alexa top sites global list that both support ECS (as of 9/26/2021) and have TTLs longer than 60 seconds: `www.google.com` (rank 1), `www.youtube.com` (rank 2), `facebook.com` (rank 7), and `www.wikipedia.org` (rank 13). Facebook only supports ECS when `www` is not included. In addition to domains selected from the Alexa top sites global list, we also probed a popular Microsoft CDN domain that supports ECS and used its ECS traces. We use Microsoft logs of clients accessing this domain to validate our cache probing technique. In the main body we present results for cache probing aggregated across all these domains.

Table 5 provides results on how well each individual domain performs in discovering activity at prefix and AS granularities (top), and cross-comparing each pair of domains (bottom). Since different domains may reply with different response scopes, we treat prefixes returned by different domains as matching as long as one prefix contains the other. Although `Wikipedia` has many fewer active prefixes than other domains, we identify a large number of unique ASes by probing `Wikipedia`. `Wikipedia` returns fewer prefixes because it usually replies with response scopes between 16 and 18 while the rest of the domains usually reply with scopes between 20 and 24. We identify the largest number of unique prefixes and ASes by probing for `Google`, possibly due to the site's popularity and wide-spread usage. Conversely, we identify relatively few new prefixes and ASes by probing for `YouTube`, despite its popularity as #2 on Alexa top sites global list and the large number of active prefixes found by probing it. This lack of new information is largely explained by the overlap of active prefixes between `Google` and `YouTube`, as 89% of active prefixes discovered with `YouTube` are also

	Google	YouTube	Facebook	Wikipedia	Microsoft CDN
Total prefixes	336,261	214,069	164,697	65,462	137,011
Unique prefixes	15,630 (4.6%)	2,644 (1.2%)	590 (0.3%)	2,756 (4.2%)	2,720 (2.0%)
Total ASes	19,859	14,296	11,991	13,367	14,818
Unique ASes	3,607 (18%)	584 (4%)	963 (8%)	2,536 (19%)	2,384 (16%)
Google	336K (100%)	198K (59%)	192K (57%)	274K (82%)	231K (69%)
YouTube	191K (89%)	214K (100%)	136K (64%)	174K (82%)	139K (65%)
Facebook	136K (82%)	130K (79%)	165K (100%)	152K (92%)	131K (79%)
Wikipedia	56K (85%)	51K (78%)	45K (69%)	65K(100%)	42K (64%)
Microsoft CDN	110K (80%)	108K (79%)	99K (72%)	117K (85%)	137K (100%)

Table 5: For the top half of the table: each column shows the number of prefixes that had cache hit(s) to this domain (*`Total prefixes`*), the number of prefixes that had cache hit(s) to only this domain (*`Unique prefixes`*), the number of ASes had cache hit(s) to this domain (*`Total ASes`*), and the number of ASes had cache hit(s) to only this domain (*`Unique ASes`*). For the bottom half of the table: each entry shows the number and the percentage of prefixes found through cache hit(s) in row's domain that also had cache hit(s) to the column domain.

found in Google. Finally, we discovered that probing Facebook does not add many new prefixes or ASes. We attribute this to the fact that Facebook only supports ECS without www. Since Facebook uses its domain with www by default, the version without www may be queried less often by real users.