Characterizing IPv4 Anycast Adoption and Deployment

Danilo Cicalese
Telecom ParisTech
danilo.cicalese@enst.fr

Jordan Augé
Telecom ParisTech
jordan.auge@enst.fr

Diana Joumblatt
Telecom ParisTech
joumblat@enst.fr

Timur Friedman
UPMC Sorbonne Universités
timur.friedman@lip6.fr

Dario Rossi
Telecom ParisTech
dario.rossi@enst.fr

ABSTRACT

This paper provides a comprehensive picture of IP-layer anycast adoption in the current Internet. We carry on multiple IPv4 anycast censuses, relying on latency measurement from PlanetLab. Next, we leverage our novel technique for anycast detection, enumeration, and geolocation [17] to quantify anycast adoption in the Internet. Our technique is scalable and, unlike previous efforts that are bound to exploiting DNS, is protocol-agnostic. Our results show that major Internet companies (including tier-1 ISPs, over-the-top operators, Cloud providers and equipment vendors) use anycast: we find that a broad range of TCP services are offered over anycast, the most popular of which include HTTP and HTTPS by anycast CDNs that serve websites from the top-100k Alexa list. Additionally, we complement our characterization of IPv4 anycast with a description of the challenges we faced to collect and analyze large-scale delay measurements, and the lessons learned.

CCS Concepts

•Networks → Network measurement; Network structure; •General and reference → *Measurement;*

Keywords

Anycast; Census; Network monitoring; Network measurement; IPv4; BGP

ACM CoNEXT'15 Dec 1-4, 2015, Heidelberg, Germany
© 2015 ACM. ISBN 123-4567-24-567/08/06…$15.00
DOI: 10.475/123_X

1. INTRODUCTION

Modern content-delivery networks (CDNs) employ L7-anycast, exploiting DNS and HTTP redirection techniques to direct traffic from a client to any server in a group of geographically dispersed but otherwise equivalent servers. Such redirection techniques perform load balancing among nearby replicas and map users to the closest replica, reducing user-perceived latency.

Network-level (IP) anycast [4] is another instantiation of the same principle, where a set of replicas spread across a number of locations around the world share a standard unicast IP address. BGP policies route packets sent to this address to the nearest replica according to BGP metrics, notably (though not only) the number of autonomous system (AS) hops.

L7 anycast and IP anycast are complementary. On one hand, L7 anycast allows for very dense server deployments with customized user-server mapping algorithms and complex operations to shuffle content among servers. Although this allows a fine grain control of the server selection, it also increases the management complexity [36]. On the other hand, IP anycast offers a loose control over user-server mapping, which limits the deployment density but considerably simplifies management by delegating replica selection to IP routing.

In recent years, the scientific community has made significant contributions to understand L7 anycast, e.g., to uncover deployments and geolocate points of presence (PoPs) with active measurements [15, 45–47], and characterize the performance of L7 anycast via passive measurements [5, 12]. Yet, with few exceptions [45] as these application-level deployments are diverse, and as PoPs are pervasive, such efforts generally focus on a single application/player such as Akamai [46], YouTube

[5,47], Amazon [12] and Google [15]. A recent trend in the area of Internet infrastructure mapping is to exploit the edns-client-subnet DNS extension (ECS) [20] to uncover the geographical footprints of major CDNs [15, 45]. Still, given the wide design space and flexibility in L7 anycast implementations, it is hard to generalize results of L7 anycast usage, performance, and geographical deployments across CDNs.

Conversely, most IP anycast studies [9,34,43] are limited to DNS, which has historically been the killer application of IP anycast [3,7,29,37]. This paper shows that the usage of IP anycast has significantly changed in recent years. In particular, major players of the Internet ecosystem including Internet service providers (ISPs), OTTs, and manufacturers provide a diversity of services with IP anycast (e.g., content distribution, cloud services, web hosting, web acceleration, DDoS protection). Yet, missing an Internet-scale study of IP anycast deployment, the scientific community is not up-to-date with such changes, and knowledge related to non-DNS anycast (e.g., anycast IP address ranges, the number of replicas behind each address, services provided) is anecdotal at best, which motivates our current work.

Indeed, while valuable research efforts (Sec. 2), started with seminal work such as [30] and culminated with [1, 22] more recently, focus on *unicast censuses*, this paper presents the first census of the use of IPv4 *anycast* in the Internet. First, we describe the challenges faced in designing a system able to collect and analyze Internet-scale delay measurements [17] in a short time frame (Sec. 3). Next, we discuss the results of a thorough experimental campaign, analyzing anycast adoption over multiple anycast IPv4 censuses (Sec. 4). To summarize our main contributions:

- We conduct and combine delay measurements from four full censuses, based on which we find about $O(10^3)$ IP/24 subnets to be anycasted.
- We characterize the geographical footprint of IP anycast deployments, that we (conservatively) find on average to have $O(10)$ replicas.
- We provide empirical evidence that IP anycast is used by ASes in the CAIDA top-10 rank and by ASes serving content over HTTP and HTTPS for websites in the Alexa top-100 rank.
- We show that anycast is used to serve a large diversity of stateful services (a complementary portscan finds 10,000 open port, about 500 of which are well known) running on top of TCP.
- We describe our distributed system design, able to perform and analyze one census in under 5 hours.
- We make our census results browsable at [21].

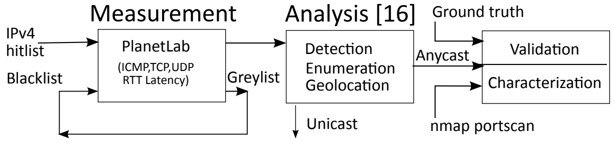


Figure 1: Overall workflow of the anycast census

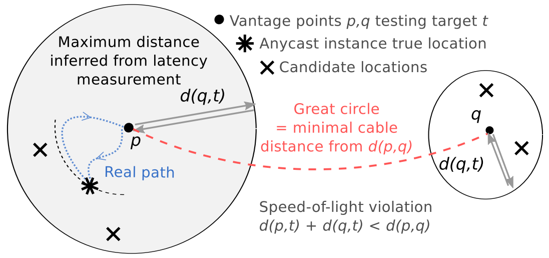

● Vantage points p,q testing target t
✱ Anycast instance true location
✕ Candidate locations
Maximum distance inferred from latency measurement
d(q,t)
p
Real path
Great circle = minimal cable distance from d(p,q)
q
d(q,t)
Speed-of-light violation
d(p,t) + d(q,t) < d(p,q)

Figure 2: Analysis technique: anycast detection (figure adapted from [17])

2. METHODOLOGY OVERVIEW

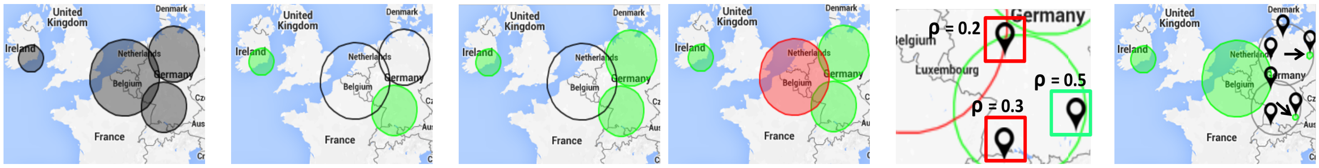
In this section, we present an overview of our work (Sec. 2.1) and put it in perspective with prior research efforts (Sec. 2.2). For the sake of readability, we describe our complete workflow with the help of Fig. 1.

2.1 Workflow

Measurements. We use a distributed software running over PlanetLab (PL) to conduct IPv4 anycast censuses with ICMP latency measurements. Each of the $O(10^2)$ PL vantage points (VP) receives a set of $O(10^7)$ IP/32 targets (namely, the IPv4 hitlist provided by [31]). We consider that each IP/32 in this hitlist is representative of the corresponding IP/24 subnet, and thus cover the entire IPv4 address space (we validate this assumption in Sec. 3.1). Later on, we thoroughly justify our choices of the measurement platform (e.g. PL over RIPE, MLab, Archipelago, etc. in Sec. 3.2), software (e.g., fastping/TDMI over Zmap in Sec. 3.3), and protocols (e.g., ICMP over TCP or UDP in Sec. 3.4).

Analysis. The dataset collected from the census is uploaded to a central repository. We run an iterative algorithm that we recently proposed [17] to detect, enumerate, and geolocate anycast replicas over the dataset. For the sake of completeness we provide an overview of the technique, which is based on detection of speed-of-light violations [35]: as depicted in Fig. 2, the main idea is that in case latency measurements from two vantage points toward the same target exhibit geo-inconsistency, then it is safe to assume the target to be anycast.

We illustrate an execution of the technique in Fig. 3. Briefly, given a specific target IP, (a) we first map each RTT latency measurement to a disk centered around the



(a) **Measurement**: Map RTT samples to disks centered around VPs

(b) **Detect**: Non-overlapping disks imply speed-of-light violation

(c) **Enumerate**: Solving a Maximum Independent Set (MIS) problem yields non-overlapping disk, each containing a different replica (two steps shown)

(d) **Geolocate**: Maximum likelihood classification problem (city-level)

(e) **Iterate**: Collapse disks around geolocated replicas until convergence

Figure 3: Illustration of the Analysis technique: enumeration and geolocation steps)

VP, that by definition contains the target contacted;(b) if two such disks do not intersect, as just discussed, we can infer that VPs are contacting two different replicas, as is the case for the green discs in Fig. 3(b). While observation of inconsistency among any pair lead to anycast *detection*, leveraging multiple observations it is possible to further *enumerate such replicas.* Enumeration is described in step (c): to provide a conservative estimation of the minimum number of anycast replicas, we solve a Maximum Independent Set (MIS) problem. MIS outputs a set of non-overlapping disks which contain a different replica of the same target: while MIS problem is NP-Hard, we solve it using a 5-approximation algorithm that greedily operates on disks of increasing radius size as in Fig. 3(c), and that in practice yield results that are very close to the optimum provided by a prohibitively more costly brute force solution [17]. Geolocation happens in step (d): in the smallest disk, we geolocate the replica at city-level granularity with a maximum likelihood estimator biased toward city population; actually, we find that the city population has sufficient discriminative power alone (about 75% accuracy [18]), so that our geolocation criterion boils down into picking the largest city in that disk. Finally, (e) we coalesce the disk to the classified city, which reduces disk overlap and allows iteration of the algorithm until convergence, thus increasing the recall (i.e., number of replicas discovered) along each iteration.

The analysis technique [17] is of course not an original contribution of this work, whose main aims are instead to scale up its application to an Internet-wide census on the one hand (Sec. 3), and to analyse and publish the gathered dataset on the other hand (Sec. 4).

Characterization and Validation. In addition to anycast detection, the previous steps allow to geolocate the replicas behind each anycast IP/24. As outlined in Fig 1, we validate the output of the geolocation step whenever a ground truth is available (as in Sec. 3.4 for CDNs such as CloudFlare and Edgecast, complementary to the validation limited to DNS in [17]).

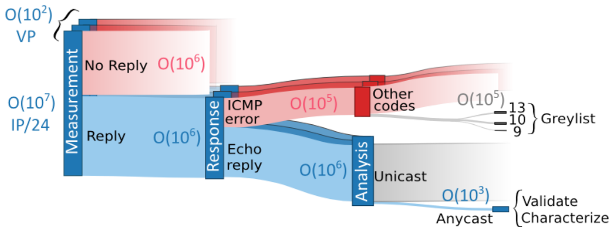


Figure 4: Anycast census at a glance: typical census magnitude

Finally, we provide a fine-grained characterization of IP/24 anycast services. While our detection methodology is service-agnostic, we use nmap [38] on a list of anycast IPs obtained from the census, to reveal open ports and the software their run. Given that an exhaustive portscan (i.e., of the 2^{16} TCP and UDP portspaces, over all replicas of all anycast deployments) still incurs a prohibitive measurement cost, we restrict the measurements to TCP services (i.e., the most unexpected ones) and interesting deployments (i.e., deployments with large geographical footprints). We discuss anycast services in Sec. 4.3, finding over 10,000 open ports, that map to about 500 well-known services, and fingerprinting some 30 software applications.

Scale, Completeness, and Accuracy. Although we described the anycast geolocation technique in [17], we had to overcome several challenges to run it at Internet-scale and within a short timespan, for which we went through multiple re-engineering phases. We believe that a number of lesson learned (e.g., as the counter-intuitive need to slow-down the sending rate to complete a census) are worth sharing, and discuss them in Sec. 3.

Notice that a large number of vantage points is required to provide an accurate picture of anycast deployment, especially in terms of the number of replicas discovered around the world. Related work that focuses on $O(1)$ targets (i.e., DNS root-servers) indeed run measurement campaigns involving from $O(10^4)$ [10] to

$O(10^5)$ [25] vantage points to achieve ≈90% recall [25]. In our case, given the sheer size $O(10^7)$ of our target set, we tradeoff completeness for scale, and possibly underestimate the number of IP-anycast replicas, as we use a mere $O(10^2)$ vantage points.

Still, our results provide a broad, conservative, yet accurate picture of Internet anycast usage: for targets for which we have the ground truth, our city-level geolocation is accurate in about 75% of the cases with a median error of 350 Km otherwise (Sec. 3.4).

Typical census. In this work, we perform four IPv4 censuses and analyse the results obtained from their combination. For each of the $O(10^2)$ VPs the magnitude of a typical census is illustrated in Fig. 4: starting from a hitlist of $O(10^7)$ targets, less than half send a reply (Sec. 3.1). ICMP replies include $O(10^5)$ errors, some of which relates to administratively prohibited communication: senders of these ICMP error messages are added to a *greylist*, to avoid probing them again in future censuses (Sec. 3.3). Finally, running the anycast geolocation technique over the $O(10^6)$ targets that generate valid ICMP echo reply messages, we discover roughly $O(10^3)$ IP/24 anycast deployments, corresponding to approximately 0.1‰ of the whole IPv4 address space – *the proverbial needle in the IPv4 haystack*.

2.2 State of the art

Anycast vs Unicast. In standard IP unicast censuses, the set of targets can be split among VPs for scalability reasons. In contrast, in the anycast case, all targets should be probed by all VPs to provide an accurate map of geographical footprints. Given that the number of active VPs in PL is around 300, and that only one IP/32 target for each IP/24 subnet needs to be probed in a given anycast census, it follows that the raw amount of probe traffic is only slightly larger than that of an unicast censuses.

In the unicast case, the relative location of VPs with respect to targets is irrelevant. Therefore, census strategies cover extremes such as a single centralized high-end server capable of $O(10^6)$ probes-per-second on a well provisioned 1 Gbps Ethernet connection as in Zmap [22], or a highly decentralized system that exploits $O(10^5)$ low-resource gateways as in the illegal Carna Botnet census [1]. Our system design sits halfway between these two extremes: in particular, it uses an efficient application-level multi-threaded scanner capable of $O(10^4)$ probes/sec, distributed over $O(10^2)$ PlanetLab nodes.

It is also worth pointing out that an independent analysis [33] of the Carna Botnet dataset found multiple campaigns, covering a cumulative number of probes exceeding a full IPv4 census, over a duration of 8 months. Additionally, [33] suggests that due to an overlap in the target set, not all hosts were probed, neither during the two fast scan campaign identified, nor during the whole measurement period. As such, our measurement campaign comprising *multiple anycast censuses* each lasting only *few hours*, constitutes an achievement per se.

Geolocation and infrastructure mapping. Unicast geolocation is a well investigated research topic. Numerous techniques based on latency measurements [23, 24, 28] and databases [41, 44] have been proposed. Yet, database techniques are not only unreliable with unicast [41], but also with anycast, since they advertise a single geolocation per IP. Similarly, latency-based techniques [23, 24, 28] use triangulation, and geolocate unicast addresses at the intersection of multiple latency measurements from geographically dispersed vantage points. However, this assumption no longer necessarily holds for anycast as depicted in Fig. 3.

L7-anycast infrastructure mapping studies [15,45] leverage ECS requests to geolocate servers: (millions of) requests are sent with different client IPs from one VP to unveil (thousands of) unicast IP addresses corresponding to PoPs of major OTTs. However, ECS support is becoming widespread to enhance the user online experience, but is not yet pervasive. Finally, the technique fails with alternative L7-anycast design relying on HTTP redirection.

To the best of our knowledge, no IP-anycast geolocation technique exists other than our own: as such no other study, apart from this work, deals with anycast infrastucture mapping. With respect to ECS-based technique for L7 anycast, our technique allows to reduce the cardinality of the problem without sacrificing geolocation accuracy (a qualitative comparison is provided in [17]). Additionally, since BGP provides a unified redirection technique, IP-anycast offers an unprecedented opportunity to broadly assess all deployments at once.

Anycast discovery and characterization. Prior work investigates different aspects related to anycast, with a focus on discovery and enumeration [25, 35] or on characterization [9–11, 19, 32, 34, 43], but to the best of our knowledge, not on anycast census. Closest to ours [17] is the work of [25, 35]. Specifically, [17] is a service-agnostic technique for detection, enumeration and geolocation. Similarly to [17], speed of light violation is used in [35] (however limited to detection and not capable of enumeration/geolocation). Conversely, [25] exploits DNS-specificities (i.e., CHAOS requests) to enumerate DNS replicas (but unlike [17] is neither capable of geolocation, nor applicable beyond DNS).

Other studies assess the performance of current IP anycast deployments, with a focus on metrics such as proximity [9,10,19,34,43], affinity [9–11,13,34,43], availability [10,32,43], and load-balancing [10,11]. Yet, these studies focus on DNS, which is just a piece of the current anycast puzzle.

Finally, some work study IP-anycast CDN, such as [16, 27]. However these focused studies add yet other useful pieces to the puzzle, that remained so far incomplete, lacking the broad perspective given by a Internet-wide coverage over all prefixes and services.

3. SYSTEM DESIGN

Anycast detection relies on measuring round trip delays between a set of vantage points and a target IP address to uncover geo-inconsistencies. Running an Internet census thus requires measurements towards millions of destinations, ideally in a short timeframe: we now describe and justify system design choices that allow us to perform multiple censuses, that we analyze later in Sec. 4. Items discussed in this section concern the selection of targets (Sec. 3.1), the measurement platform (Sec. 3.2) and software (Sec. 3.3), as well as the network protocol used (Sec. 3.4). Finally, we report considerations about the scalability of our workflow (Sec. 3.5).

3.1 Census targets

Census granularity. Unlike multicast, anycast addresses need no reservation into the IP space: as any IP address can be a candidate, this makes deployment easy, but the detection of anycast addresses hard. Luckily, to avoid a significant increase in the size of routing tables, BGP standard practice [4] is to ignore or block prefixes shorter that /24. Thus, /24 is the minimum granularity for anycasted services, which is a good granularity for our census. We validate this assumption with (spot) verifications for all IP addresses on some IP/24 (belonging to EdgeCast), confirming any IP in the /24 to be equivalent for anycast detection purposes. Additionally, a /24 granularity implies that announced BGP prefixes that are smaller than /24 are tested multiple times, one per each /24 they contain: the mapping between /24 and announced prefixes is still possible a posteriori, as we do in this work. This choice is reinforced by [35], which found 88% of announced prefixes to be /24, stated that “anycast prefixes are dominated by /24” and suggested that larger prefixes may be anycast only in part due to BGP prefix aggregation. We therefore fix the census granularity to a single target IP per /24.

Target liveness. As previously argued, any alive IP belonging to a /24 is equivalent in telling whether the whole /24 is anycast (or unicast). To identify a *responsive* IP address in every /24-prefix, we rely on the hitlist periodically published by [31]. The hitlist consists in generally one representative IP address for $O(10^7)$ prefixes, along with a score indicative of the host liveliness, computed over several measurement campaigns. When no alive IP has been observed in a /24, the hitlist contains an arbitrary address from that /24 (score ≤ -2). After covering the full hitlist with the first census, we confirm these hosts not being reachable and remove them to reduce the target size to $6.6 \cdot 10^6$ per VP.

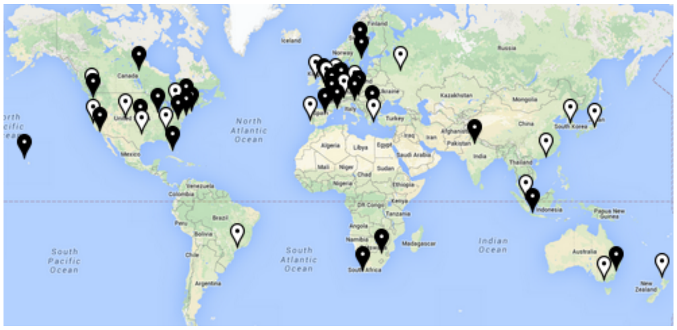


Figure 5: Microsoft deployment as seen from PlanetLab (21 replicas) vs RIPE (54 replicas). Notice that PlanetLab results (white markers) are a subset of RIPE (white *and* black markers)

Coverage. Given our census aim, we verify how well this hitlist covers all routed /24 prefixes. We therefore obtain from CAIDA a dump of routing tables originating from both RIPE RIS and RouteViews collectors. To compare the hitlist vs the advertised prefixes, we split the latter in /24, obtaining 10,616,435 /24 prefixes, of which 10,615,563 have a representative in the hitlist (over 99.99% coverage). We additionally cross-check our observed target responsiveness with the expected recall: specifically, recent ICMP scans [48] observe $4.9 \cdot 10^6$ used /24 subnets and our campaigns similarly capture $4.4 \cdot 10^6$ responsive subnets (90% coverage with respect to [48]).

3.2 Measurement dataset vs platform

Dataset. One option to avoid running a large scale measurement campaign is to exploit readily available datasets from public measurement infrastructures – yet we could not find any fitting our purpose. For instance, despite probing all /24 every 2-3 days, Archipelago [6] clusters its vantage points into three independent groups, each using random IPs selected in each /24 prefix: it follows that at most 3 monitors target each /24, with generally different IP addresses, and a hit rate of about 6%. Given the low hit-rate and low-parallelism, such dataset is not appropriate for our purpose, as it would not lead to a complete census, nor to an accurate geolocation footprint even in case of hits.

Platforms. There are a number of available measurement platforms in the community, each with its own advantages and limitations. Except for illustration purposes, in this paper we relied on PlanetLab (PL). While

RIPE Atlas (RIPE for short) is more interesting for geographical diversity due to its scale, it has a limited control on the rate and type (cf. Sec. 3.4) of measurements, as well as their instantiation for such a large scale campaign (i.e., upload of the hitlist, probing budget). Additionally, the larger number of vantage points would mechanically increase about 20-fold the amount of probes per census with respect to PL (in case all VPs are used). Conversely, measurement in PL are limited by node availability (generally around 300 vantage points), but offer full flexibility for deploying custom software and run it at high speed (cf. Sec. 3.3).

While in this work we limitedly use PL, we depict for illustration purposes an application of our technique from measurements collected from PlanetLab vs RIPE in Fig. 5: as PL results are a subset of RIPE results, white markers indicate replicas found from both platforms, while black markers pinpoint replicas that are only found with RIPE measurements. While this example has anecdotal relevance, it suggests that an intriguing direction is to *combine* both platforms, e.g., by refining via RIPE the geolocation of anycast /24 detected via PL.

3.3 Measurement software

Fastping. An efficient measurement tool is needed to maximize the probing capacity of our VPs. While at first sight Zmap [22] could seem the perfect tool for such large-scale campaign, it however exhibits a major blocking point in our setup: namely, Zmap generates raw Ethernet frames, which are very efficient in a local setup, but are not supported by the PlanetLab virtualization layer. We therefore resort to Fastping [26], a tool specialized, as the name implies, in ICMP scanning which is deployed on each PL node. Fastping is able to send about $O(10^4)$ probes per second – about two orders of magnitude slower than Zmap, but faster than the fastest nmap scripting engine scanner. As we will point out later concerning scalability (Sec. 3.5), in order to gather *complete censuses* in few hours, we had to undergo several rounds of re-engineering – including *purposely slowing down Fastping sending-rate.*

Greylist. Additionally, Fastping adopts the usual techniques to be a good Internet measurement citizen – i.e., a signature in the payload points to its homepage, Fastping probes the target list in a randomized order to reduce intrusiveness, and implements a greylist mechanism to honor requests to stop probing administratively prohibited hosts/networks inferred from ICMP return codes. Before running a census from $O(10^2)$ VP we initially run a census from a single VP in order to build an initial blacklist. During any census, we then collect addresses generating ICMP return codes (other than echo reply) in a temporary greylist, that we later incrementally merge with the the blacklist. This list has approximately $O(10^5)$ hosts, with 98.5% added due to administrative filtering [8] (type 3 code 13) and the remaining in reason of communications administratively prohibited at network or host levels (respectively 1.3%, code 10 [14] and 0.2%, code 9 [14]).

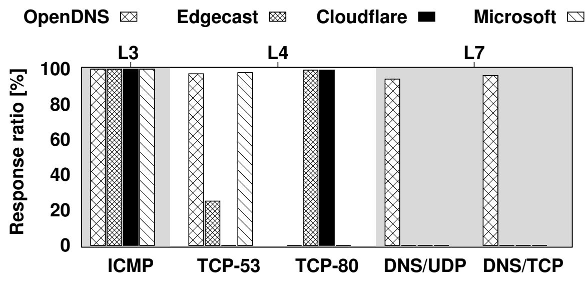


Figure 6: Response rates seen by heterogeneous protocols across different targets.

3.4 Network protocol

Recall. ICMP has often been used (and misused) in measurement studies: especially given recent work showing that ICMP latency measurements are often *not* reliable [40], we thus need to confirm the validity of our protocol selection. A major motivation for ICMP measurement is given by the high recall it offers [48]. Consider indeed that TCP and UDP measurements would need an a priori knowledge (or guess) of services running on the target under test. We therefore perform a test on a reduced set of targets, performing 100 measurements with different protocols: specifically, we consider network L3 (ICMP) and transport L4 (TCP SYN-SYN/ACK pair in the three-way handshake to port 53 or 80) measurements, as well as L7 (DNS/UDP vs DNS/TCP using `dig`) measurements. Fig. 6 shows that protocols other than ICMP have a binary recall: in other words, they work well only if the service is known *a priori*. Conversely, ICMP is the only reliable alternative, yielding high recall across all deployments, and is thus well suited for censuses.

Accuracy. While our technique relies on latency measurements, it leverages the discriminative power of side channel information (i.e., cities population within disks), to cope with latency measurement noise. While we validate the accuracy of the methodology for DNS in [17], a validation for stateful TCP connections is still missing. To do so, we build a ground truth (GT) for CloudFlare and EdgeCast by performing HTTP measurements with `curl` from PL: note that HTTP measurements are not available from RIPE, highlighting once more the com-

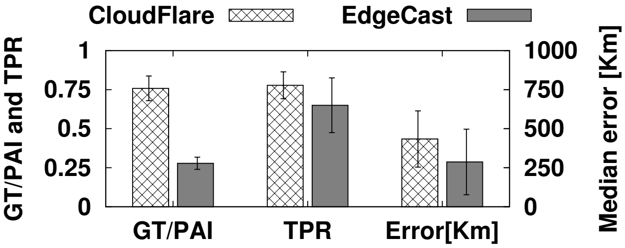


Figure 7: Validation with CloudFlare and EdgeCast ASes. Bars represent standard deviation among IP/24 of the AS.

plementarity of these platforms.

By inspection of the HTTP headers, we find that CloudFlare (EdgeCast) encode geolocation of the replica in the custom `CF-RAY:` (standard `Server:`) header field. Notice that the measured GT constitutes the upper-bound of what can be possibly achieved from PL measurements, while the publicly available information (PAI) displayed on the CloudFlare and EdgeCast websites contains a super-set of locations with respect to those measured from PL. We contrast true positive (TPR) classification of our census vs HTTP GT in Fig. 7: in 77% of the IP/24 for CloudFlare (65% for EdgeCast) there is agreement at city level, with a median error of 434 Km (287 Km for EdgeCast) in the (relatively few) misclassification cases. As expected, the low number of PL nodes possibly limits the portion of discoverable replicas (GT/PAI is fairly high for CloufFlare, but fairly low for EdgeCast), making our footprint estimates conservative and confirming the interest for alternative platforms such as RIPE.

Consistency. Additionally, in the case of openDNS, we verify consistency across multiple RTT latency measurement techniques used early in Fig. 6. In this case we rely on public information that maps 24 locations [39]. For all protocols, applying [17] on the dataset yields between 15 and 17 instances. Notice that all cities returned by the analysis are correct except Philadelphia (while the server is located in Ashburn at 260km or 2.6ms worth of propagation delay away): this misclassification is due to the bias enforced in [17] toward city population (Philadelphia is 33 times more populated than Ashburn), but as observed in [15] this is not problematic as the "physical" Ashburn location is actually serving the "logical" Philadelphia population.

3.5 Scalability

Probing rate. When designing census experiments, we take care of avoiding obvious pitfalls. For instance, while we target a single host per /24, nevertheless we perform measurements from all PL nodes. It follows that each node must desynchronize to avoid hitting ICMP rate limiting (or raising alert) at the destination. We do so by randomized permutation for target nodes, achieved via a Linear Feedback Shift Register (LFSR) with Galois configuration. Still, while the LFSR solves rate limiting *at the target*, it does not solve problems *at the source* (or in the network): indeed, while *requests* are well spread, *replies* do aggregate close to the VP, that receives an aggregate rate equal to the probing rate of Fastping (in excess of 10,000 hosts per second). In our preliminary (and incomplete) censuses, we noted heterogeneous (and possibly very high) drop rates for some VPs (likely tied to rate limiting spatially close to the VP). Given that the networks where PL machines are hosted are independently administered, we opted for a simple solution and slowed down Fastping by one order of magnitude[1], that we verified empirically not triggering the above problems. Consequently, probing $6.6 \cdot 10^6$ targets at 10^3 targets per second takes less than two hours: as shown in Fig.8 about 40% of PL nodes complete within this timeframe, and 95% in under 5 hours (longer duration likely due to load on the PL host).

Table 1: Textual (0) vs binary (1-4) censuses

Census ID	Format	Size (host,total)	Analysis
0	csv	(270M, 79G)	>3 days
1-4	binary	(21M, 6G)	3 hr

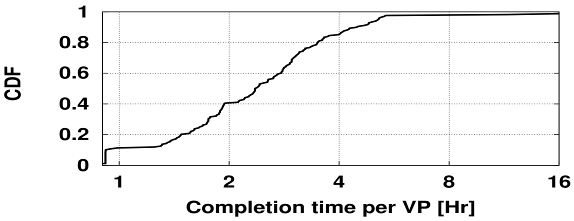


Figure 8: CDF of per-vantage point completion time, over all censuses

Output size and analysis duration. A second scalability issue concerns the output format. We initially overlooked this issue and logged, in textual format, a wealth of information amounting to 270M per node and 80GB overall per census (cf. Tab.1). We therefore opted for a radical reduction of the output size, dumping a stripped-down binary format containing a timestamp, delay and ICMP flag (encoding greylist return codes 9, 10, or 13 as a negative sign) for a total of about 20MB per node and 6GB overall per census.

A third challenge lies in the analysis of the data. For a single target, the running time of [17] is $O(10^{-1})$ sec,

[1] While it is possible to more finely tune the probing rate per VP, however coverage may benefit from samples coming from the slowest VP, especially if it resides in a geographical area which is not well covered by PL.

which compares very favorably to the $O(10^3)$ sec of the brute force optimal solution: at the same time, processing a *census* would still take *days* (we indeed have stopped processing the complete Census-0 after 3 days of CPU time, where textual format additionally led to slow processing due to disk fragmentation). Moreover, due to LFSR, the order of the target IPs in all files is not the same, meaning that an on-the-fly sorting of about 300 lists (one per VP) containing millions targets is needed. We therefore optimized our implementation, which currently runs in under three hours, i.e., about the same timescale of the census duration, so that in principle we could perform a continuous analysis. While this is not interesting for the anycast characterization use-case, it may become relevant for other applications of this technique (e.g. BGP hijacking inference mentioned in Sec. 5).

4. ANYCAST /0 CENSUSES

This section presents results of the first Internet-wide anycast study. We start by aggregated statistics (Sec. 4.1) and then incrementally refine the picture by providing a bird's-eye view of the most interesting deployments (Sec. 4.2) over which we perform an additional portscan campaign to reveal their running services (Sec. 4.3).

4.1 At a glance

Details about the of our censuses are reported in Fig. 10. Overall, 1696 IP/24 belonging to 346 ASes appear to have more than one anycast replica, while we were able to find only 897 IP/24 belonging to 100 ASes having at least 5 replicas with our technique. The plot also shows a geographical density map of anycast replicas: results of our censuses are available for browsing at [21], offering per-deployment (as in Fig. 5) or aggregated (as in Fig. 10) visualizations. Notice that results reported in this paper correspond to censuses performed during March 2015: with later censuses, we observed small but interesting changes in the anycast landscape. While we plan to run a continuous service, please be advised that (at time of writing) results at [21] refer to the censuses described in this paper.

Several remarks from Fig. 10 are in order. First, notice that our results are conservative since (i) in regions with low presence of PlanetLab VPs, we may miss some anycast replicas, e.g., when the BGP prefix is only locally advertised; (ii) the analysis technique provides a lower bound on the number of replicas, since overlapping disks may correspond to different anycast replicas but they will not be considered in the solution of the MIS problem (recall Fig. 3). Second, we investigate the CAIDA AS rank list, to cross check how many ASes using IP-anycast figure in the top-100: results tabulated in Fig. 10, show that 19 IP/24 of 8 ASes that play a central role in the Internet belong to the list. Similarly, we investigate the Alexa rank, to cross check how many webpages in the top-100k rank are hosted[2] by ASes using IP-anycast: here again, we find 242 IP/24 of 15 ASes that are among the major players of the Web.

	IP/24	ASes	Cities	CC	Replicas
All	1,696	346	77	38	13,802
≥ 5 **Replicas**	897	100	71	36	11,598
∩ **CAIDA-100**	19	8	30	18	138
∩ **Alexa-100k**	242	15	45	29	4,038

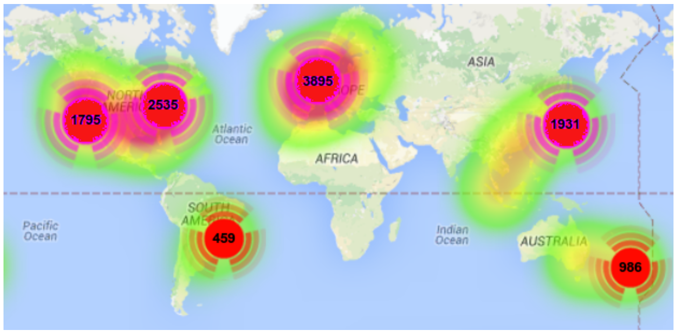


Figure 10: Anycast censuses results, at a glance.

4.2 Top-100 Anycast ASes

Albeit the amount of anycast IP/24 may seem deceiving at first in reason of its exiguous footprint, it is nevertheless very rich – *revealing silver needles in the haystack*. From the very coarse cross-check of CAIDA and Alexa ranks, we already expect that anycast usage is not only restricted to DNS, but rather covers important ISPs and OTTs. Fig. 9 presents a bird's-eye view of anycast adoption, depicting several information for the 100 ASes for which we detected at least 5 replicas, identified by their WHOIS name reported in the x-axis (capped to 12 characters). *Geographical and IP/24 footprint* are reported in the bottom: ASes are arranged left to right, in decreasing number of replicas (bottom bar-plot, with standard deviation across IP/24 belonging to the same AS), additionally reporting the number of anycast IP/24 for that AS (middle bar-plot). *Service footprint* is correlated to the open TCP ports in the AS (middle scatter-plot). Next, the *relative importance* of the AS in the Internet and for the Web are expressed in terms of the CAIDA and Alexa ranks respectively (top scatter-plots). Finally, a label reported on the top x-axis categorize the main activity of the ASes from a business perspective (category is informal and in case of ASes with multiple services, only the most prominent is selected).

[2] For the sake of simplicity, we resolve the domain name of the frontpage found in Alexa to an IP, and disregard content that is referenced in the frontpage.

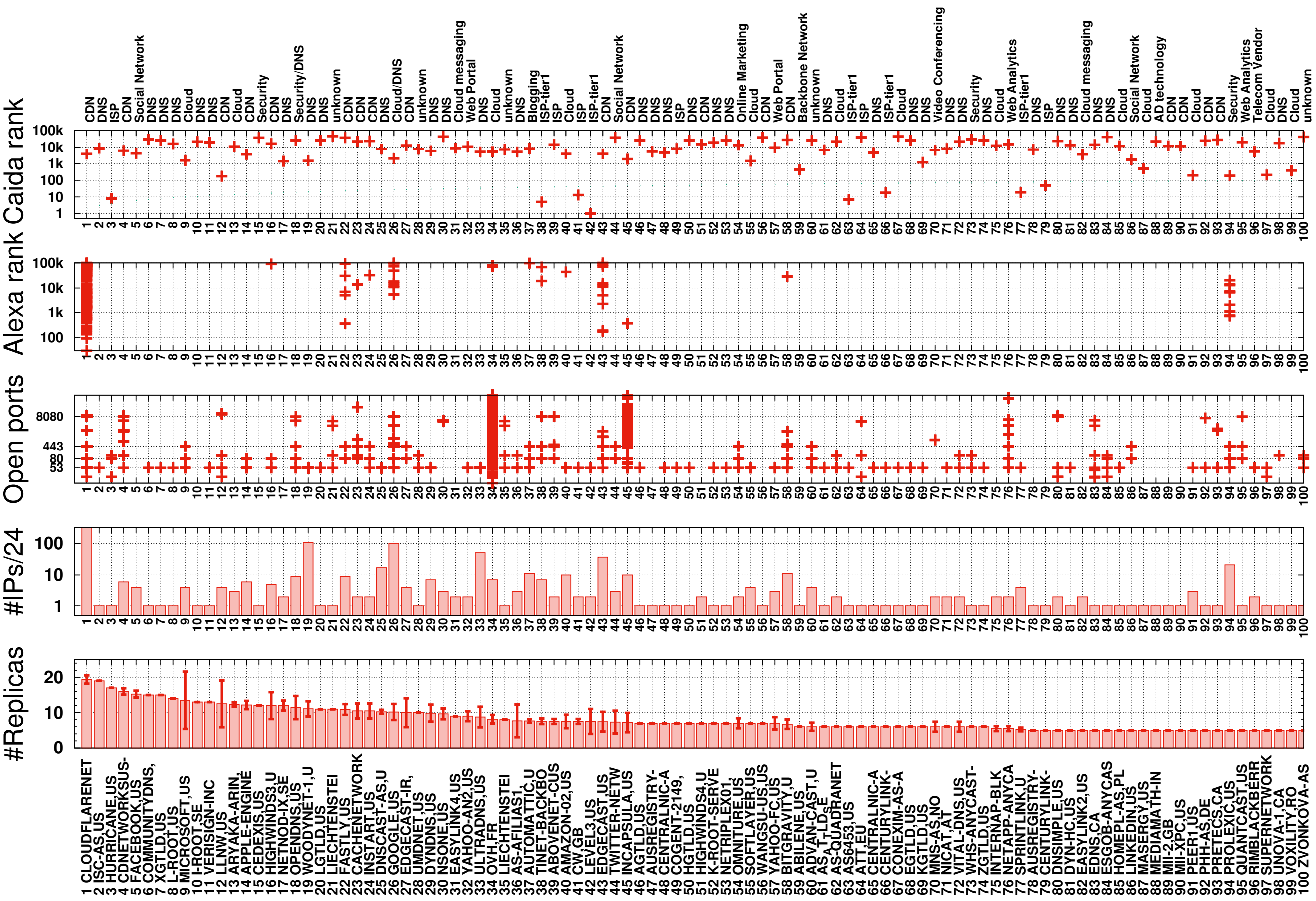


Figure 9: Bird's eye view of Top-100 anycast ASes (ranked according to geographical footprint)

Big fishes. Major players of the Internet ecosystem are easy to spot in Fig. 9. The list includes not only tier-1 and other ISPs (such as AT&T Services, Tinet, Sprint, TATA Communications, Qwest, Level 3, Hurricane Electrics), but also a rather large spectrum of OTTs such as CDNs (e.g., CloudFlare, EdgeCast), hosting (e.g., OVH) and cloud providers (e.g., Microsoft, Amazon Web Services), social networks (e.g., Twitter, Facebook, LinkedIn), and security companies that provide mitigation services against DDoS attacks (e.g., Prolexic, OpenDNS). The list also includes manufacturers (e.g., Apple, RIM), Web registrars (e.g., Verisign, nic.at), virtual roaming and virtual meeting services (Media Network Services), blogging platforms (Automattic, a publishing company hosting wordPress.com), cloud messaging (EASYLINK2 owned by AT&T Services), and web analytics (OMNITURE owned by Adobe Systems). Of course, DNS-related service providers such as root and top-level domain servers (e.g., ISC/F-root, CommunityDNS), DNS service management (e.g., UltraDNS, DynDNS), and public DNS resolvers (e.g., Google DNS, OpenDNS) also emerge in the census.

Diversity. We report a breakdown of AS classes in Fig. 11, crisply showing that DNS now represents about one third of IP anycast activities. Plots in Fig. 9 clearly illustrate the large diversity of anycast usage, under all metrics. Indeed, no correlation appear between any two metrics, illustrating the degree of freedom in anycast deployments: for instance, the geographical footprint and IP/24 footprints are largely unrelated (Pearson correlation coefficient of 0.35). Additionally, the number of open ports, and the specific port values, vary not only across deployments having an heterogeneous business model, (e.g., we observe from a minimum of 1 open port for DNS to $O(10^4)$ open ports for OVH) but also between deployments of the same kind (e.g., CloudFlare and EdgeCast CDNs have in common only port 53, 80 and 443 over the set of 22 open ports they are using, with CloudFlare using 4× more ports than EdgeCast).

Geographical footprint. We specifically study the *mean number of geographical replicas per AS* (bottom plot in Fig. 9) championed by the CDN CloudFlare in our measurement. Overall, we observe that 25 ASes

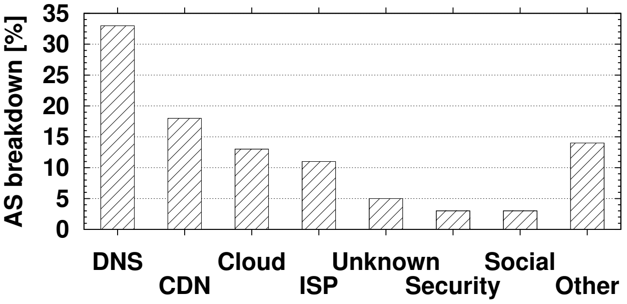


Figure 11: Breakdown of AS category (only first category is considered).

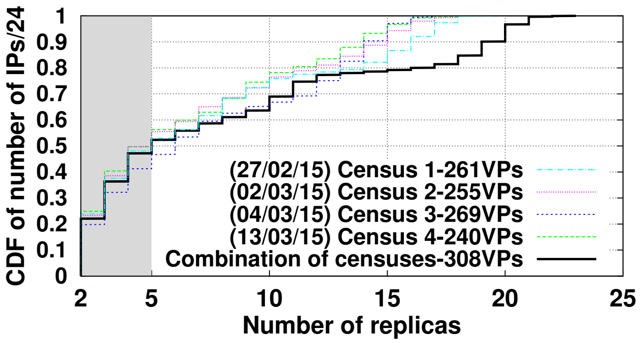


Figure 12: CDF of geographically distinct replicas per IP/24 (individual censuses and overall)

have at least 10 replicas distributed around the globe. Notice that these orders of magnitude are, significantly smaller with respect to L7-anycast deployments that can exceed $O(10^3)$ for the large providers, which is in part due to the low number of vantage points in PlanetLab (see Sec.3.2). Among those ASes, we observe 10 DNS service providers (including ISC, DNScast, and DynDNS) and 7 major CDNs (e.g., CloudeFlare, Limelight, Highwinds, Fastly, CacheNetworks, Instart Logic, CDNetworks). We also discover two cloud providers (e.g., Microsoft and Aryaka Networks), one tier-1 ISP (Hurricane Electric which has 15% of ASes in its customer cone according to CAIDA), a security company (OpenDNS, also popular for its public DNS service), a social network (Facebook) and a manufacturer (Apple).

Fig. 12 further reports the cumulative number of replicas per IP/24, depicting both results coming from the combination of censuses, as well as individual result from each census alone. Specifically, the MIS solver orders circles by increasing radius size: intuitively, the smaller the latency, the lower the number of overlaps, the better the recall of our method. This is confirmed in Fig. 12, where censuses are combined by computing the *minimum* among multiple latency measurements between the same VP and target pair, to get an estimate of the RTT latency that is as close as possible to the propagation delay. Additionally, combining measurement increases recall: about 200 more IP/24 are found to be anycast in the combination with respect to the average individual census.

In this paper, we limitedly consider results from the combination, but remark that results are quite consistent across censuses (notice that curves overlap in Fig. 12). A last comment is worth making about deployments where we observe only 2 geographically distributed replicas – which is possibly due to the low density of our VPs, but could also be tied to the wrong geolocation of some VP raising false positive replicas. While we have anecdotal evidence of some of these exiguous deployments being anycast, we prefer to defer a more detailed analysis for future work (see Sec. 5).

IP/24 footprint. In terms of the *number of anycast IPs/24 per AS* (middle plot in Fig. 9), we find that the CDN CloudFlare is by far the largest in terms of IP address ranges. Overall, we find 10 ASes that have at least 10 anycast IPs/24: 3 are CDNs (CloudFlare, EdgeCast, BitGravity), 3 are DNS providers (DNScast, WoodyNet, UltraDNS), and the remaining ASes represent multiple services (Automattic, Google, Amazon Web Services, and Prolexic). The distribution of the number of IPs/24 per AS depicted in Fig. 13 shows that about half have exactly one IP/24 (e.g., LinkedIn and AT&T Services). Yet, about 10% of the ASes employ at least 10 subnets: for instance Prolexic, EdgeCast, Google, and CloudFlare employ 21, 37, 102, and 328 anycast IP/24 respectively.

While in this work we do not provide a systematic investigation of the deployment density (i.e., how many IP/32 are alive in each IP/24), from the above discussion about diversity is not surprising that we were able to identify both very sparse (e.g., Google 8.8.8.8 is the only address alive in the 8.8.8.0/24) and very dense deployments (e.g., well over 99% of IPs are alive in most CloudFlare subnets).

Importance. The presence of ASes ranking among the top-100 in the CAIDA list, as well as CDNs serving content in the top-100k Alexa list are good indicators that anycast is used for popular and important services. Considering CDNs that are, after DNS, the most popular anycast service according to Fig. 11, we observe that 8 CDNs serve Alexa-100k websites: this set includes CloudFlare, EdgeCast, and Fastly with 188, 10, and 5 websites respectively (in addition, Highwinds, CacheNetworks, Instart, Incapsula, and BitGravity host one popular site each). In addition, 11 of the websites listed by Alexa are hosted by Google anycast IPs. Finally,

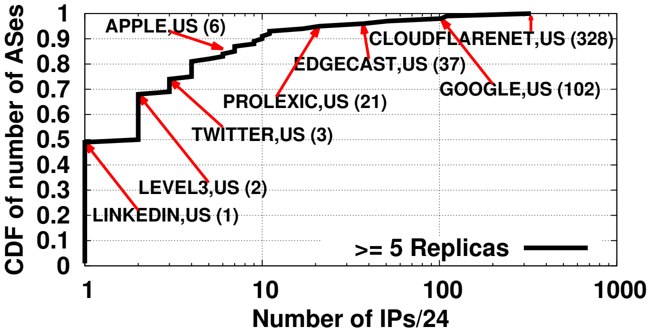


Figure 13: Number of IPs/24 per AS

10 websites are hosted on IPs that belong to Prolexic (now part of Akamai), which operates a DDoS mitigation service that receives the traffic on behalf of its client networks, redirecting only legitimate traffic to them.

4.3 Anycast Services

Portscan campaign. In reason of the historical use of anycast for DNS services, we believe it to be important to provide an up-to-date longitudinal view across services offered via IP-anycast, especially focusing on TCP. We provide a summary of the nmap probing in the top of Fig. 14. We test all anycast /24 of the top-100 ASes: picking a single IP representative per /24 we scan, at low rates, all 2^{16} TCP ports. Our results are conservative in that different IPs in the same /24 may have different open ports (which happens, e.g., for CloudFlare and EdgeCast), and since an under-estimation of the number of open TCP ports can also be the result of probe filtering by firewalls and routers along the path to the targets. Out of the 897 IP of the top-100 ASes, we find that 816 of 81 ASes have at least one open TCP port. The total number of distinct open TCP ports across is 10485, providing 449 well-known services (i.e., as indicated by TCP port classification), 170 of which over SSL. Additionally, nmap fingerprinting discovers 30 different software implementations running on the anycast replicas, that we also detail next.

Class imbalance. Given the heterogeneity of the IP/24 footprint, we argue being necessary to consider only per-AS statistics to avoid presenting results that are biased due to class imbalance. We illustrate the problem by depicting in Fig. 14 the frequency count of the top-10 open TCP ports by number of ASes (top) and IPs/24 (bottom). Notice that only port 80, 443 and 53 appear to be common to both top and bottom plots: especially, all ports in the hatched area are due to the large predominance of IP/24 owned by the CloudFlare AS, which also affect the order of common ports in the top-10. We thus focus on per AS statistics in the following.

nmap portscan statistics

IPs/32	ASes	Ports (SSL)	Well known	Software
812	81	10,499 (185)	457	30

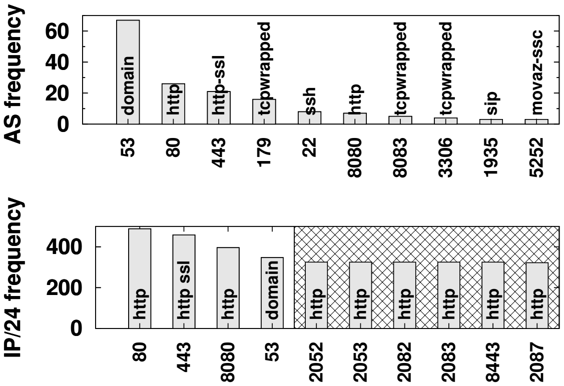


Figure 14: Overall nmap portscan statistics and Top-10 open TCP ports (per AS and per /24).

Stateful services. Fig. 15 presents the complementary CDF of the number of open TCP ports per AS. We make the following observations: (i) roughly half of the ASes have at least one open TCP port, (ii) about 10% of the ASes have at least 5 open TCP ports and (iii) the largest service footprint is represented by Incapsula and especially OVH with 313 and 10148 open ports respectively. In the latter case, while we did not investigate thoroughly, we suspect the large number of ports being due to the fact that OVH, the largest hosting service in Europe and the 3rd in the world, is significantly popular in the BitTorrent seedbox ecosystem [42]. Predominant services (beyond DNS) include fairly popular HTTP and HTTPS, used by over 20% of the ASes. Even excluding the OVH case, the list of interesting services is large. In terms of business diversity, 22 ASes have at least 4 different TCP ports open: 8 CDNs, 4 DNS, 4 ISPs including a tier-1 ISP (Tinet SpA) and Google with 9 open TCP ports. Finally, interesting (though unpopular) services worth listing include multimedia services (RTMP, Simplify Media, MythTV), and gaming (Minecraft).

Software diversity. Fig. 16 lists 30 different software that we group into three main categories: Web, Mail, and DNS. Interestingly, the list includes open source software such as popular web and DNS daemons (e.g., nginx, ISC BIND) and proprietary software (e.g., ECAcc/ECS/ECD which are web servers developed by EdgeCast). Starting with DNS software, notice that for 44 ASes using port 53 (out of 67), nmap could not identify the software version running on the remote server. Unsurprisingly, we find that ISC BIND is by far the

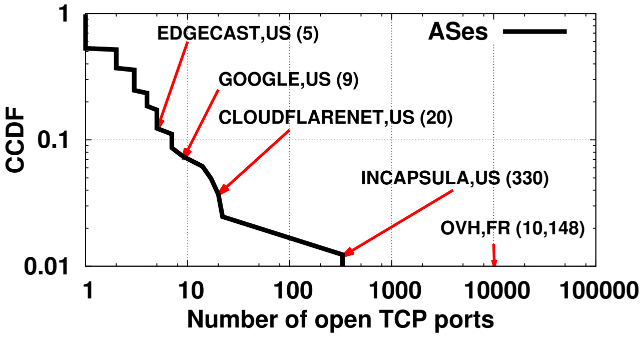


Figure 15: Complementary CDF of the number of open TCP ports per AS.

most adopted protocol to handle DNS requests over anycast. Yet, we also detect the use by 3 ASes (Apple, K-ROOT, L-ROOT) of the NLnet Labs NSD implementation, which is specifically designed to add resilience against software failures of DNS root servers.

Among web servers, the most popular are nginx (7 ASes), Apache httpd and lighttpd (ex æquo with 4 ASes). We observe the use of proprietary web servers by some CDNs (e.g., cloudflare-nginx and Panel httpd). Though our dataset has a limited size, we attempt a comparison with the relative popularity of webservers in the unicast world: the Spearman correlation of popularity rank in our dataset with webserver ranks [2] in the Alexa-10M is low (0.38). As for the DNS case, difference may arise in some peculiar features that are especially valuable in the anycast context. Finally, we detect the presence of running daemons that serve mail on anycast IPs from Google (Gmail imapd, Gmail pop3d, gsmtp) as well as of RPC (ssh, MicrosoftRPC) and databases (MySQL/Microsoft SQL).

5. DISCUSSION

We present the first census(es) of IPv4 anycast deployment, gathered through an original and robust technique implemented with an efficient and scalable system design. In spirit with the open source and data movement, results of our census are available at [21].

Our characterization show that a tiny fraction of the IPv4 space is anycasted, yet among the anycasters we recognize major players of the Internet ecosystem including top-ranking ISPs, popular Cloud, OTT and especially CDN operators. We additionally show great heterogeneity along multiple directions, and especially in terms of the offered services. Particularly, our portscan campaign of anycasted subnets reveals over 450 well-known services from over 10, 000 unique open ports. Additionally, we uncover 30 software implementations, with a relative breakdown that differs from software ranking in the unicast IP world.

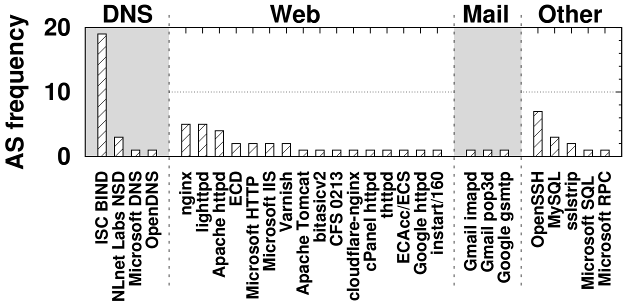


Figure 16: Breakdown of software running on anycast replicas.

Yet, this work only scratches the surface, and opens more questions than it is able to answer, as for instance:

IP-level CDN: Refining the active methodology by mapping content object (and not only frontpage) from the Alexa-100k would be needed to gather a better understanding of IP-level CDN.

Longitudinal view: Taking periodic censuses and analyzing the time evolution over longer timescales would allow to track evolution of IP anycast deployments.

Traffic volume: A missing information concerns the traffic volume served by IP anycast, that can be gathered via passive measurement, and annotated with results of our census (i.e., binary flag per anycast IP/24).

Combine measurement platforms: As we have seen, it would be interesting to exploit multiple platforms in addition to PlanetLab, such as RIPE Atlas: this would both lead to a better characterization of large deployments (e.g., increase the recall), as well as possibly assist in confirming/discarding suspicious deployments (i.e., those for which we detected 2 replicas from PL).

BGP hijacking: Detecting geo-inconsistencies for knowingly unicast prefixes is symptomatic of BGP hijacking attacks: being able to periodically and quickly scan the network to raise alarms and cross-check them with other types of data (e.g., BGP feeds, traceroute measurements) is a relevant extension of this work.

Acknowledgements

We thank our shepherd, Matteo Varvello, and the anonymous CoNEXT reviewers for their valuable feedback. This work has been carried out at LINCS (http://www.lincs.fr). The research leading to these results was supported by the European Union under the FP7 Grant Agreement n. 318627 (Integrated Project ”mPlane”) and by a Google Faculty Research Award.

6. REFERENCES

[1] http://internetcensus2012.bitbucket.org/paper.html.
[2] http://w3techs.com/technologies/overview/web_server/all.

[3] J. Abley. A software approach to distributing requests for DNS service using GNU Zebra, ISC BIND 9 FreeBSD. In *Proc. USENIX ATEC*, 2004.

[4] J. Abley and K. Lindqvist. Operation of Anycast Services. RFC 4786 (Best Current Practice), 2006.

[5] V. K. Adhikari, S. Jain, and Z. li Zhang. Youtube traffic dynamics and its interplay with a tier-1 isp: An isp perspective. In *ACM IMC*, 2010.

[6] http://www.caida.org/projects/ark/.

[7] http://www.root-servers.org.

[8] F. Baker. Requirements for IP Version 4 Routers. IETF RFC 1812, 1995.

[9] H. Ballani and P. Francis. Towards a global IP anycast service. In *Proc. ACM SIGCOMM*, 2005.

[10] H. Ballani, P. Francis, and S. Ratnasamy. A measurement-based deployment proposal for IP anycast. In *ACM IMC*, 2006.

[11] B. Barber, M. Larson, and M. Kosters. Traffic source analysis of the J root anycast instances. Nanog, 2006.

[12] I. Bermudez, S. Traverso, M. Mellia, and M. Munafo. Exploring the cloud from passive measurements: The Amazon AWS case. In *Proc. IEEE INFOCOM*, 2013.

[13] P. Boothe and R. Bush. DNS Anycast Stability: Some Early Results. CAIDA, 2005.

[14] R. Braden. Requirements for Internet Hosts - Communication Layers. IETF RFC 1122, 1989.

[15] 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.

[16] M. Calder, A. Flavel, E. Katz-Bassett, R. Mahajan, and J. Padhye. Analyzing the performance of an anycast cdn. In *ACM IMC*, 2015.

[17] D. Cicalese, D. Joumblatt, D. Rossi, M.-O. Buob, J. Augé, and T. Friedman. A fistful of pings: Accurate and lightweight anycast enumeration and geolocation. In *Proc. IEEE INFOCOM*, 2015.

[18] D. Cicalese, D. Joumblatt, D. Rossi, M.-O. Buob, J. Augé, and T. Friedman. Latency-based anycast geolocalization: Algorithms, software and datasets. In *Tech. Rep.*, 2015.

[19] L. Colitti. Measuring anycast server performance: The case of K-root. Nanog, 2006.

[20] C. Contavalli, W. van der Gaast, S. Leach, and E. Lewis. Client Subnet in DNS Queries. https://tools.ietf.org/html/draft-ietf-dnsop-edns-client-subnet-04.

[21] http://www.enst.fr/~drossi/anycast.

[22] Z. Durumeric, E. Wustrow, and J. A. Halderman. Zmap: Fast internet-wide scanning and its security applications. In *USENIX Security Symposium*, 2013.

[23] B. Eriksson, P. Barford, J. Sommers, and R. Nowak. A learning-based approach for IP geolocation. In *PAM*, 2010.

[24] B. Eriksson and M. Crovella. Understanding geolocation accuracy using network geometry. In *Proc. IEEE INFOCOM*, 2013.

[25] X. Fan, J. S. Heidemann, and R. Govindan. Evaluating anycast in the domain name system. In *Proc. IEEE INFOCOM*, 2013.

[26] http://www.ict-mplane.eu/public/fastping.

[27] A. Flavel, P. Mani, D. A. Maltz, N. Holt, J. Liu, Y. Chen, and O. Surmachev. Fastroute: A scalable load-aware anycast routing architecture for modern cdns. In *Proc. USENIX NSDI*, 2015.

[28] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-based geolocation of internet hosts. In *ACM IMC*, 2004.

[29] T. Hardie. Distributing Authoritative Name Servers via Shared Unicast Addresses. IETF RFC 3258, 2002.

[30] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister. Census and survey of the visible internet. In *ACM IMC*, 2008.

[31] Internet addresses hitlist dataset (20140829/rev4338). Provided by the USC/LANDER project (http://www.isi.edu/ant/lander).

[32] D. Karrenberg. Anycast and BGP stability: A closer look at DNSMON data. Nanog, 2005.

[33] T. Krenc, O. Hohlfeld, and A. Feldmann. An internet census taken by an illegal botnet: A qualitative assessment of published measurements. *ACM CCR*, 44(3), 2014.

[34] Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, and K. C. Claffy. Two days in the life of the DNS anycast root servers. In *PAM*, 2007.

[35] D. Madory, C. Cook, and K. Miao. Who are the anycasters. Nanog, 2013.

[36] B. M. Maggs and R. K. Sitaraman. Algorithmic nuggets in content delivery. *SIGCOMM Comput. Commun. Rev.*, 45(3), Jul 2015.

[37] K. Miller. Deploying IP anycast. Nanog, 2003.

[38] https://nmap.org.

[39] https://www.opendns.com/data-center-locations.

[40] C. Pelsser, L. Cittadini, S. Vissicchio, and R. Bush. From paris to tokyo: On the suitability of ping to measure latency. In *ACM IMC*, 2013.

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

[42] D. Rossi, G. Pujol, X. Wang, and F. Mathieu. Peeking through the BitTorrent seedbox hosting ecosystem. In *Proc. Traffic Monitoring and Analysis (TMA)*, 2014.

[43] S. Sarat, V. Pappas, and A. Terzis. On the use of anycast in DNS. In *Proc. ACM SIGMETRICS*, 2005.

[44] Y. Shavitt and N. Zilberman. A geolocation databases study. *IEEE J-SAC*, 29(10), 2011.

[45] F. Streibelt, J. Böttger, N. Chatzis, G. Smaragdakis, and A. Feldmann. Exploring edns-client-subnet adopters in your free time. In *ACM IMC*, 2013.

[46] A. Su, D. R. Choffnes, A. Kuzmanovic, and F. Bustamante. Drafting behind akamai (travelocity-based detouring). In *Proc. ACM SIGCOMM*, 2006.

[47] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao. Dissecting video server selection strategies in the YouTube CDN. In *Proc. of IEEE ICDCS*, 2011.

[48] S. Zander, L. L. Andrew, and G. Armitage. Capturing ghosts: Predicting the used ipv4 space by inferring unobserved addresses. In *ACM IMC*, 2014.