Anycast de nos jours

Danilo Cicalese

To cite this version:

Danilo Cicalese. Anycast de nos jours. Réseaux et télécommunications [cs.NI]. Télécom ParisTech, 2018. Français. NNT : 2018ENST0025 . tel-02477974

HAL Id: tel-02477974

https://pastel.hal.science/tel-02477974

Submitted on 13 Feb 2020

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2018-ENST-0025

EDITE - ED 130

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Informatique et Réseaux »

présentée et soutenue publiquement par

Danilo CICALESE

le 3 Mai 2018

Anycast de nos jours.

Directeur de thèse : **Dario ROSSI** – Télécom ParisTech, France

Jury

M. Alberto DAINOTTI, Chercheur, Center for Applied Internet Data Analysis	Rapporteur
Mme. Cristel PELSSER, Professeure, Université de Strasbourg	Rapporteure
M. Ernst BIERSACK, Professeur, Eurécom	Examinateur
M. Antonio PESCAPÈ, Professeur, Università degli Studi di Napoli Federico II	Examinateur
Mme. Renata TEIXEIRA, Directeur de Recherches, INRIA	Examinatrice
M. Timur FRIEDMAN, Maître de Conférences, Sorbonne Université	Invité

THÈSE

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

PhD ParisTech

A dissertation presented

in fulfillment of the requirements for the degree of Doctor of Philosophy of

TELECOM ParisTech

Speciality Computer Networking and Telecommunications

presented and defended in public by

Danilo CICALESE

On Mai 3 2018

Anycast Nowadays

PhD Director: **Dario ROSSI**

À mes chers parents, mon frère et ma soeur.

Abstract

The work of this thesis originates from the investigative curiosity of discovering IP anycast, a technique commonly used to share the load of a variety of global services. The adoption of this technique has increased in recent years. Once relegated to root and top-level Domain Name System (DNS) servers, anycast is now commonly used by Content Delivery Networks (CDN) and other key Internet players. In this thesis, we aim to raise awareness about IP anycast by building a comprehensive picture of it in order to demonstrate which companies are using it and how they do so.

First, we focus on the identification of a protocol-agnostic and lightweight technique used for the discovery and geolocation of anycast replicas. Other measurement techniques used for identifying and enumerating anycast replicas exist. However, they exploit specifics of the DNS protocol, which limits their applicability to this particular service. We also provide the community with open-source software and datasets. These serve to both replicate our experimental results and facilitate the development of new techniques such as ours.

This methodology enables the next step of our research: unveiling all the companies that currently use anycast for their services. We carry on multiple IPv4 anycast censuses, relying on latency measurements from a distributed platform. We collect and analyze these large-scale delay measurements. These censuses ultimately find that many major Internet companies utilize anycast. In addition to identifying the companies, updated information on anycast IPs and their geolocation can serve for many other purposes.

Hence, the decision was made to put a system in place, which is capable of performing monthly IPv4 censuses and analyzing the results of one-year measurements. Our results, data, and code are also shared with the community.

Finally to complete the picture, we investigate the users and services that anycast CDNs are serving at large on the Internet. We perform a passive characterization focusing on the services they offer, their penetration, etc. Our findings reveal that more than 50% of web users access content served by anycast CDNs during peak time. A broad range of Transmission Control Protocol (TCP) services are offered over anycast. These can include audio, video streaming, and HTTP & HTTPS, the latter of which are the most popular.

KEY-WORDS: Geolocation, Anycast, Network Monitoring, Network Measurements, IPv4

Résumé

Les motivations des recherches réalisées dans cette thèse viennent de la curiosité de découvrir IP anycast. Cette technique est couramment utilisée pour partager la quantité d'information d'une variété de services globaux. L'adoption de cette technique a augmenté au cours de ces dernières années. Une fois reléguée aux serveurs racine et domaine de premier niveau du système de noms de domaine (DNS), anycast est maintenant communément utilisé par les réseaux de distribution de contenu (CDN) et d'autres acteurs clés d'Internet. Dans cette thèse, nous visons à sensibiliser la communauté de l'utilisation d'IP anycast en composant une image complète afin de montrer quelles entreprises l'utilisent et comment elles le font.

Tout d'abord, nous nous concentrons sur l'identification d'une technique, qui ne depends pas d'un protocole specific, utilisé pour la découverte et la géolocalisation des répliques anycast. D'autres techniques utilisées pour identifier et énumérer des répliques anycast existent déjà. Toutefois, ils exploitent les spécificités du protocole DNS, ce qui limite leur applicabilité à ce service. Nous fournissons également à la communauté des logiciels open-source avec des jeu de donnée. Ceux-ci servent à reproduire nos résultats expérimentaux et à faciliter le développement de nouvelles techniques.

Cette méthodologie nous a permis de mettre en oeuvre la prochaine étape de notre recherche: dévoiler toutes les entreprises qui utilisent actuellement anycast pour leurs services. Nous effectuons plusieurs IPv4 census d'anycast, en utilisant des mesures de latence à partir d'une plateforme distribuée. Nous collectons et les analysons. Ces census révèlent finalement que de nombreuses grandes entreprises d'Internet utilisent anycast. De plus identifier les sociétés et avoir des informations à jour sur les IP anycast et leur géolocalisation peuvent servir à d'autres finalités. Nous avons donc décidé de mettre en place un système capable d'effectuer des census IPv4 mensuels et d'analyser les résultats d'un an. Nos résultats, données et codes sont également partagés avec la communauté.

Enfin, pour compléter l'étude, nous analysons les utilisateurs et les services que les CDN anycast diffusent sur Internet. Nous effectuons une caractérisation passive en mettant l'accent sur les services qu'ils offrent et leur pénétration etc. Nos résultats révèlent que normalment plus de 50% des internautes accèdent au contenu servi par les CDN anycast. Une large gamme de services TCP (Transmission Control Protocol) est offerte sur anycast. Celle-ci peut inclure des services d'audio, de streaming vidéo ou des HTTP & HTTPS, ces derniers étant les plus populaires.

MOTS-CLEFS: Geolocation, Anycast, Surveillance du réseau, Mesure de réseau, IPv4

Résumé en français

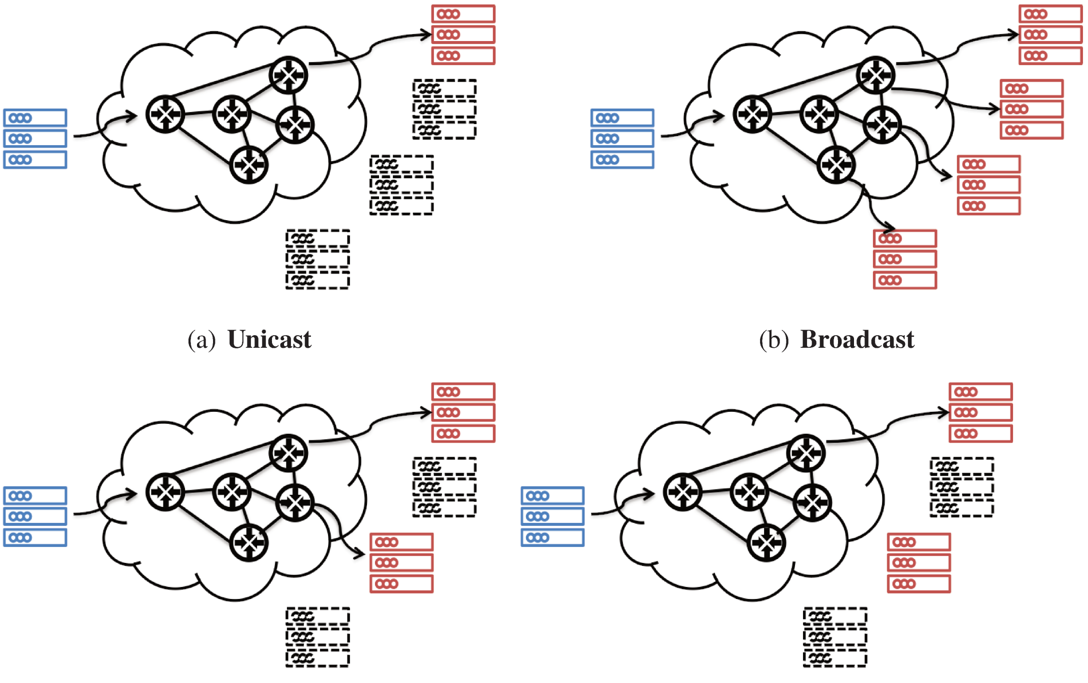
1 Introduction

L'Internet au cours de ces années a évolué et nos attentes sont changé: nous utilisons l'Internet pour des multiples activités telles que l'envoi ou lire des e-mails (par exemple Gmail), l'utilisation des moteurs de recherche pour trouver des informations (par exemple, Bing), regarder des vidéos en streaming (par exemple Netflix) et utiliser les réseaux sociaux (par exemple Twitter, LinkedIn, Facebook).

Des études dans le passé [49,99] ont souligné l'importance de la performance et de la disponibilité du contenu. D'autres études [64, 82, 109, 119, 122] montrent comment ces facteurs ont un impact économique pour les entreprises.

La latence a un impact économique: les entreprises se concentrent sur la recherche de moyens pour réduire la latence dû principalement aux surcharges de protocole et aux limites d'infrastructure, les paquets ne peuvent pas voyager à la vitesse de la lumière dans la fibre optique. Certains d'entre eux ont décidé de répliquer l'information dans des endroits géographiquement dispersé et de rediriger les utilisateurs vers le plus proche (en termes de latence).

Cette technique a été largement utilisée dans le passé par les serveurs DNS, mais dans cette thèse, nous dévoilons que dans ces dernières années de nombreux autres acteurs clés Internet ont adopté anycast et avec cette augmentation, il y a un besoin concomitant de comprendre les services anycast. Nous trouvons société de streaming



(a) **Unicast** (b) **Broadcast**

(c) **Multicast** (d) **Anycast**

Figure 1: Illustration des méthodes d'adressage Internet: à gauche, l'émetteur (bleu) et à droite, les récepteurs possibles (rouge) et autres hôtes (lignes pointillées).

vidéo comme Netflix, Content Delivery Networks (CDN) tels que EdgeCast [1], qui fait désormais partie de Verizon, et CloudFlare [16], qui annoncent de servir respectivement 1,5 milliards d'objets par heure représentant les 4% total du trafic d'Internet [10] et plus de 2 millions de sites Web [58] . Les réseaux sociaux comme Twitter [127] qui gère plus de 500 millions de tweets par jour ou moteur de recherche web comme Bing détenu par Microsoft qui sert 5 milliards de requêtes mensuelles et représente le 34% de le marché aux US [107].

2 Contributions de cette thèse

Cette thèse se concentre sur IP-anycast. Notre intérêt a été motivé par l'adoption croissante d'IP-anycast, d'une part et, d'autre part par l'absence d'études dans le domaine: dans la littérature, les chercheurs se sont concentrés principalement sur le DNS et leurs performances. [47,72,92,100,110,118,118]. Les études précédentes sont reprises dans Tab.1.

La connaissance de IP-anycast peut intéresser les chercheurs dans un large éventail des domaines: de la caractérisation, dépannage et cartographie d'infrastructure [31] mais aussi pour des tâches liées à la sécurité telles que la détection de la censure [113]. Malheureusement, les informations publiquement disponibles concernant les sociétés anycast sont souvent inconnues [104] ou, dans le cas où elles sont disponibles, telles que DNS, sont souvent obsolètes [72].

Ainsi, nous décidons de faire la lumière sur ce sujet: en essayant d'être de bons scientifiques, nous avons abordé notre recherche en essayant de répondre aux questions *Cinq Ws et un H*. Ce sont des questions qui aident à rassembler des informations: les réponses construisent une vue complète de l'IP-anycast. Dans la suite, nous listons les questions qui donnent un aperçu du sujet:

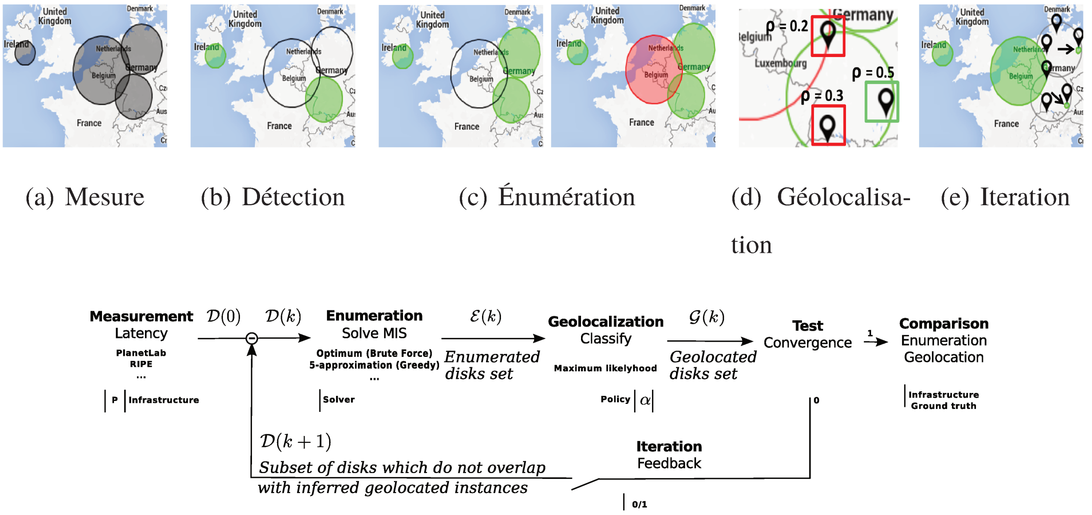
- Why did companies start to adopt anycast?
  Pourquoi les entreprises ont-elles commencé à adopter anycast?

- Where is an anycaster?
  Où est un anycaster?

- Who are all the anycasters?
  Qui sont les anycasters?

- How stable are the anycast deployments?
  Dans quelle mesure les déploiements d'anycast sont-ils stables?

- When does a company need to adopt anycast?
  Quand une entreprise doit-elle adopter anycast?



(a) Mesure (b) Détection (c) Énumération (d) Géolocalisation (e) Iteration

Figure 2: Synoptique (en bas) et illustration (en haut) de l'algorithme de détection, d'énumération et de géolocalisation iGreedy

2.1 Géolocalisation Anycast

Nous commençons notre étude en proposant une technique agnostique de protocole pour la découverte et la géolocalisation des répliques d'IP-anycast. Pourtant, d'autres techniques existent mais elles sont limitées aux déploiements DNS ou L7 anycast. Nous fournissons également à la communauté des logiciels libres et des ensembles de données pour reproduire nos résultats expérimentaux, ainsi que pour faciliter le développement de nouvelles techniques comme la nôtre.

Méthodologie

Nous avons développé un algorithme itératif capable de détecter, énumérer et géolocaliser les réplicas anycast sur l'ensemble des données. La technique, basée sur la détection des violations de la vitesse de la lumière, est illustrée en Fig. 2.

En un mot, (a) nous effectuons d'abord une mesure de latence à une adresse IP et nous les mappons à un disque centré autour d'une VP, qui, par conception, contient au

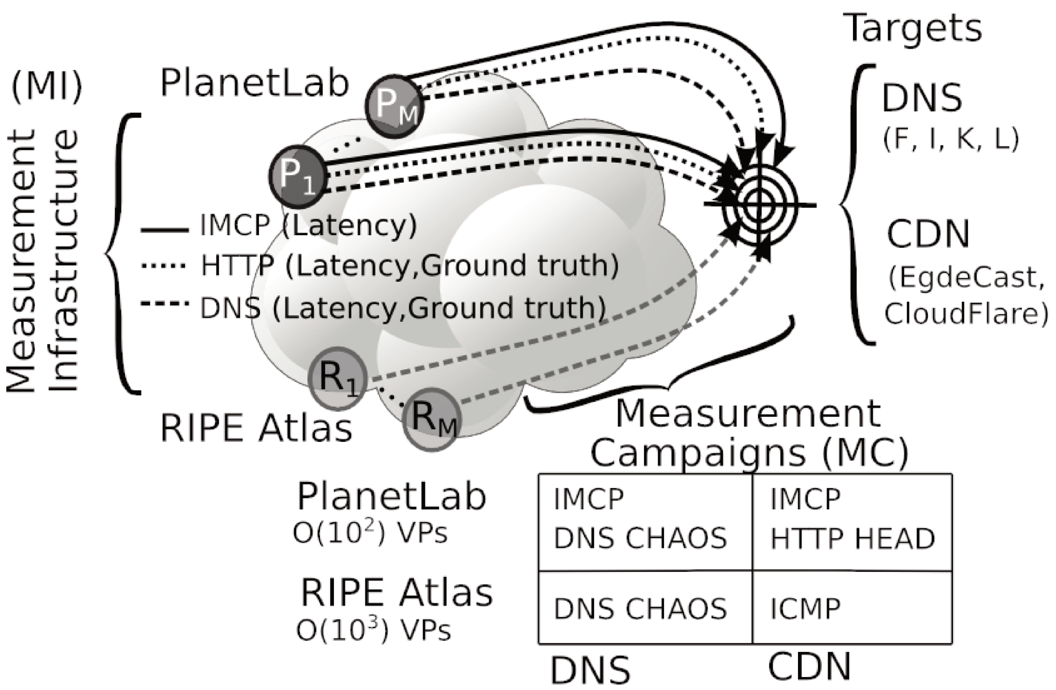
Algorithm 1 Greedy 5-approximation de MIS pour énumération d'instances anycast

```
Require: Un ensemble de disque $\mathcal{D}$
Ensure: Un ensemble de disque $\mathcal{E}$ tel que $\forall p, q \in \mathcal{E}, \mathcal{D}_p \cap \mathcal{D}_q = \emptyset$
  Initialisation: trier les disques dans $\mathcal{D}$ en augmentant la taille du rayon
  Initialisation: $\mathcal{E} \leftarrow \emptyset$
  for all disk $\mathcal{D}_d$ of $\mathcal{D}$ do
    for all disk $\mathcal{D}_e$ of $\mathcal{E}$ do
      if $\mathcal{D}_d \cap \mathcal{D}_e = \emptyset$ then
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{\mathcal{D}_e\}$
      end if
    end for
  end for
```

moins une instance anycast; (b) si deux de ces disques ne se croisent pas, nous pouvons déduire que les VPs sont en contact avec deux répliques différentes, comme c'est le cas pour les disques verts de la figure. 2(b); (c) nous fournissons une estimation prudente du nombre minimum de réplicas anycast en résolvant un problème de Set Indépendant Maximum (MIS), en utilisant un algorithme greedy 5-approximation qui fonctionne sur des disques de taille croissante de rayon comme dans la figure 2(c); (d) dans chaque disque, nous géolocalisons la réplique à la granularité au niveau de la ville avec un estimateur du maximum de vraisemblance orienté vers la population de la ville; et enfin, (e) on coalesce les disques aux villes classées, ce qui réduit le chevauchement des disques et permet l'itération de l'algorithme jusqu'à la convergence, augmentant ainsi le rappel (c'est-à-dire le nombre de répliques découvertes) le long de chaque itération.

Dataset

Nous faisons un certain nombre de *campagnes de mesure*(MC) pour fournir une mesure de la latence à partir d'un nombre relativement faible (de centaines à des milliers) des agents situés à différents points de vue autour du monde. Emplacement et

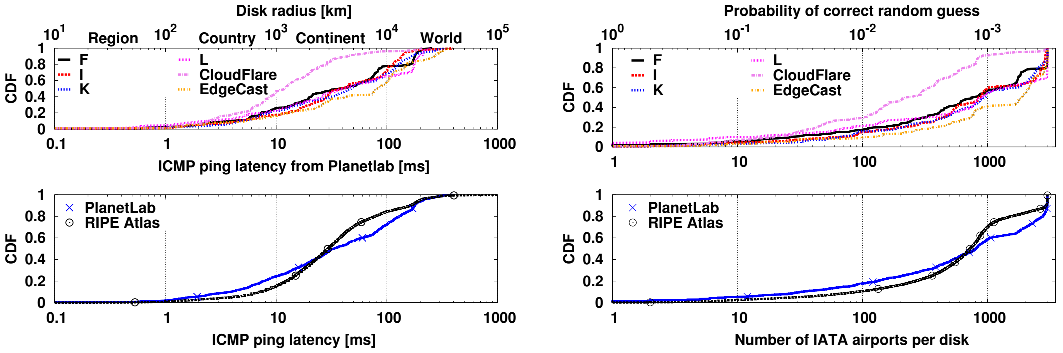


IMCP DNS CHAOS	IMCP HTTP HEAD
DNS CHAOS	ICMP

Figure 3: Synoptique du scénario de mesure anycast

nombre d'agents dépend de la *infrastructure de mesure* (MI) utilisée dans la campagne. Puisque nos MCs ciblent sont des services connus des DNS et CDN anycast, nous pouvons construire *ground truth* (GT) fiable en utilisant des informations spécifiques au protocole. En particulier, construire une GT pour le CDN anycast, à notre connaissance, est une contribution inédite, surtout depuis le GT est beaucoup plus utile en ce qui concerne *information disponible publiquement* (PAI).

Infrastructure de mesure (MI)..Dans ce travail, nous utilisons RIPE Atlas [26] et PlanetLab [25] qui sont intéressants en raison de leurs aspects de complémentarité. D'une part, RIPE Atlas a une plus grande empreinte en termes de nombre de VP (20 fois plus grand que PlanetLab) et une meilleure couverture géographique (10 fois) ou AS (5 fois), ce qui devrait nous amener à RIPE Atlas fournir une couverture plus exhaustive que PlanetLab. D'un autre côté, RIPE Atlas est plus contraint que PlanetLab dans le



(a) Statistiques dataset: Par-cible et Par-infrastructure de Cumulative distribution function (CDF) de la latence minimale sur tous les protocoles

(b) Statistiques dataset: Par-target et Par-infrastructure de Complementary CDF (CCDF) du nombre d'aéroports IATA dans chaque disque

type et le taux de mesure qui peut être effectué: par exemple, aucune mesure HTTP n'est autorisée à partir de RIPE Atlas, ce qui limite l'espace des actions dans le cas CDN. Bien que l'utilisation d'autres MI encore plus ouverts (tels que MLab, Archipelago, etc.) serait bien entendu intéressante, nous remarquons que ces plateformes ont une empreinte relativement plus petite que PlanetLab et RIPE Atlas, ce qui constitue déjà un point de départ intéressant et pertinent.

Campagne de mesure (MC). À partir des RIPE Atlas et PlanetLab MIs, nous effectuons plusieurs *campagnes de mesure* (MC) destinées à plusieurs adresses IP cibles, représentatives de différents services. Plus précisément, nous ciblons les serveurs racine DNS F, I, K et L et en plus deux adresses représentatives des CDN EdgeCast et CloudFlare.

Publicly available information (PAI). concernant nos adresses IP cibles sont généralement disponibles sur un site Web. Dans certains cas, PAI *comprend* un plus grand nombre de réplicas par rapport à ceux réellement visibles à partir des VPs. Dans d'autres cas, l'inverse est vrai: c'est-à-dire que PAI comprend un ensemble *plus petit* par rapport à l'ensemble des répliques vu de le VP, ce qui arrive chaque fois que le

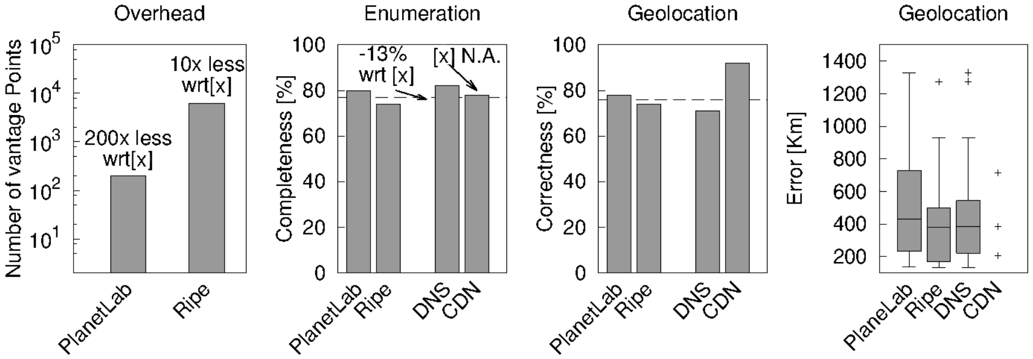


Figure 4: Récapitulatif compact des performances de géolocalisation et d'énumération d'iGreedy sur MI (PlanetLab, RIPE) et répartition des cibles PlanetLab entre les services (DNS, CDN). Comparaison avec [72] indiqué dans la Fig. avec *x*.

contenu de la page Web est obsolète.

Ground truth (GT). Pour chaque adresse IP cible dans notre dataset, nous fournissons la géolocalisation *ground truth* (GT), qui est non ambiguë et résout les problèmes mentionnés ci-dessus avec PAI. GT est assemblé en (i) effectuant des expériences supplémentaires qui exploitent des informations spécifiques au protocole et (ii) validant manuellement cette nouvelle information par rapport aux mesures PAI et de latence. Dans le cas des adresses IP des serveurs DNS racine, nous utilisons DNS CHAOS requêtes comme dans [72], alors que nous utilisons des requêtes HTTP dans le cas d'adresses CDN IP pour extraire de manière fiable des informations de géolocalisation.

Comparaison avec l'état de l'art

Nous séparons l'analyse de l'énumération et de la géolocalisation d'iGreedy comme suit. L'énumération vise à *exhaustivité*, c'est-à-dire, évaluer le nombre de disques $|\mathcal{E}|$ contactant différentes répliques que iGreedy est capable de se souvenir, indépendamment de savoir si la géolocalisation réussit. Nous normalisons le nombre de répliques au nombre de PAI obtenue avec des informations spécifiques au protocole et dénotons

Table 1: État de l'art dans la recherche de Anycast

	[104]	[72]	*This work*	[47]	[118]	[92]	[42]	[43]	[100]	[98]
Plateform(#VPs)	Renesys monitors	PL (238), Netalyzr (62k), rDNS (300k)	*PL (300), RIPE (6k)*	End-hosts $O(100)$	PL (300)	DNSMC (77)	PL (129)	rDNS (20K)	C,F,K root	Renesys monitors
Technique	BGP vs. traceroute	DNS CHAOS +traceroute	*Latency probes*	DNS CHAOS	DNS CHAOS	DNS CHAOS +BGP	DNS CHAOS	DNS queries	pcap	BGP
Targets	IPv4 prefixes	F root, TLDs, AS112	*F,I,K,L root, EdgeCast CloufFlare CDNs*	C,F-K,M root	B,F,K root, TLDs	C,F-K,M root	C,F-K,M root, AS112	F,J root, AS11		1 CacheFly prefix
Détecter	✓	✓	✓							
Énumérer		✓	✓							
Géolocaliser			✓							
Proximité					✓		✓	✓	✓	
Affinité				✓	✓		✓	✓	✓	✓
Disponibilité					✓	✓		✓		✓
Loadbalance								✓		

ce rapport $|\mathcal{E}|$/GT comme exhaustivité de l'énumération. La géolocalisation vise plutôt *exactitude*, de sorte qu'il est important d'évaluer la quantité de géolocalisation correspond correctement à la PAI, qui peut être exprimée comme le True Positive Rate or Precision = TP/(TP + FP).

Notez que les résultats de l'énumération de [72] sont *directement quantitativement* comparables, car [72] utilise F et d'autres serveurs racine comme étude de cas, bien que les délais et l'infrastructure de mesure diffèrent. Inversement, puisque iGreedy est la seule technique capable de géolocalisation anycast, nous n'avons aucune technique de comparaison directe avec dans ce cas, et rendons donc notre base de données ouverte pour promouvoir et faciliter les comparaisons futures.

D'un coup d'œil, Fig. 4 montre que, par rapport à [72], iGreedy: (a) réduit le nombre de mesure de plusieurs ordres de grandeur, (b) a un rappel d'énumération comparable. De plus, alors que la technique [72] est limitée au DNS et n'est pas capable d'effectuer la géolocalisation, iGreedy: (c) est indépendant du protocole et (d) est capable de deviner correctement l'emplacement de l'occurrence anycast sur les 3/4 des cas. Enfin, nous pouvons voir que les résultats sont (e) qualitatifs et cohérents entre l'infrastructure de service et de mesure. Comme pour (a), nous remarquons en effet que [72] utilise des 62K VPs (l'ensemble de données Netalyzr), soit environ 200× plus grand que PlanetLab et 10× plus grand que RIPE Atlas. Comme pour (b), puisque nous observons une performance d'énumération pire que celle de [72] mais néanmoins comparable, ceci signifie intrinsèquement que les jeux de données utilisés dans [72] sont hautement redondants (par exemple, incluant plusieurs essais des mêmes utilisateurs, ou affecté par la popularité de Netalyzr dans une région géographique). La même observation vaut également pour le MI que nous utilisons dans ce travail (par exemple, RIPE a plusieurs centaines de moniteurs à Paris, mais iGreedy en utilise au plus un). Comme pour (c), iGreedy est agnostique au protocole car seules des mesures de latence sont nécessaires, qui peuvent être collectées à partir de protocoles indépendants du service tels que ICMP. Comme pour (d), nous rapportons que la géolocalisation est correcte dans 78% des cas, et que la distribution d'erreur médiane des 22% restants est de 384 km.

Faire un parallèle à la géolocalisation unicast, il vaut la peine remarquer que l'amplitude d'erreur dans iGreedy est similaire à celle des techniques unicast: par exemple, sans (avec) filtrer les grandes mesures de latence, [51] signale une erreur médiane de 556Km (22Km). Une similitude supplémentaire avec mérite d'être soulignée, en citant [51] ''A disadvantage of our geolocation technique is that large datacenters are often hosted in remote locations, and our technique will pull them towards large population centers that they serve. In this way, the estimated location ends up giving a sort of logical serving center of the server, which is not always the geographic location.'':

la même prise ici pour iGreedy.

3 IPv4 Anycast Adoption et Déploiement

Dans ce qui suit, nous fournissons une image complète de l'adoption d'IP-anycast dans l'Internet actuel. Contrairement à la croyance répandue dans la communauté scientifique, l'utilisation d'anycast n'est pas limitée aux applications sur UDP (par exemple, DNS), mais inclut également les applications L7 exécutant des connexions TCP telles que les services cloud, l'web-page et atténuation des attaques DDoS. Nous présentons le premier recensement de l'utilisation de anycast IPv4 sur Internet. Plus en détail, nous menons une étude expérimentale de l'adoption anycast basée sur des mesures de délai distribué que nous recueillons à partir de multiples recensements IPv4 anycast. Nous décrivons également les défis liés à la réalisation de telles mesures à grande échelle et complétons les résultats spatiaux avec une vue longitudinale de l'évolution anycast.

3.1 Méthodologie

Campagne de mesure. Nous utilisons un logiciel distribué fonctionnant sur PlanetLab (PL) pour effectuer des recensements anycast IPv4 avec des mesures de latence ICMP. Chacun des $O(10^2)$ PL points d'observation (VP) reçoit un ensemble de cibles $O(10^7)$ IP/32 (à savoir, la liste d'occurrences IPv4 est fournie par [6]). Nous considérons que chaque IP/32 dans cette liste de résultats est représentatif du sous-réseau IP/24 correspondant et couvre ainsi l'intégralité de l'espace d'adresse IPv4.

Analyse. L'ensemble de données collectées à partir du recensement est télécharger vers un serveur central. Nous exécutons iGreedy, présenté dans le paragraphe précédent, pour détecter, énumérer et géolocaliser les réplicas anycast sur l'ensemble de données. Nous avons modifié et amélioré le code pour appliquer la méthodologie au recensement complet et nous discutons des 100 premiers cas d'anycast dans ce qui suit.

Caractérisation et validation. En plus de la détection anycast, les étapes précédentes permettent de géolocaliser les réplicas derrière chaque anycast IP / 24. Nous validons l'exactitude de la technique de géolocalisation anycast chaque fois qu'une PAI est disponible.

Alors que notre méthodologie de détection est indépendante des services, nous utilisons nmap [102] sur une liste des IP-anycast obtenus à partir du recensement pour fournir une caractérisation fine et révéler les services anycast qu'ils offrent. Étant donné qu'un portscan exhaustif (c'est-à-dire les ports 2^{16} TCP et UDP, sur toutes les répliques de tous les déploiements d'anycast) continue d'engendrer un coût de mesure prohibitif, nous limitons les mesures aux services TCP (ex., les plus inattendus) et déploiements intéressants (c.-à-d. déploiements avec de grandes empreintes géographiques). Nous trouvons plus de 10 000 ports ouverts, qui mappent à environ 500 services bien connus, et prenons une trentaine d'applications logicielles.

Census typique. Dans ce travail, nous effectuons plusieurs recensements IPv4 et analysons les résultats obtenus à partir de leur combinaison. Pour chaque VP $O(10^2)$, la magnitudo d'un recensement typique est illustrée sur la figure. 5: à partir d'une liste de cibles de $O(10^7)$, moins de la moitié envoie une réponse. Les réponses ICMP incluent des codes d'erreur $O(10^5)$ relatifs aux communications administrativement interdites: les expéditeurs de ces messages d'erreur ICMP sont ajoutés à un *greylist*, pour éviter de les réexaminer lors de recensements futurs. Enfin, en utilisant la technique de géolocalisation anycast sur les cibles $O(10^6)$ qui génèrent des messages de réponse écho ICMP valides, nous découvrons des déploiements anycast de $O(10^3)$ IP/24, correspondant à environ $0,1$‰ de l'ensemble de l'espace adresse IPv4.

3.2 Anycast /0 census

Les détails sur les résultats de nos recensements sont indiqués dans Fig. 6. Dans

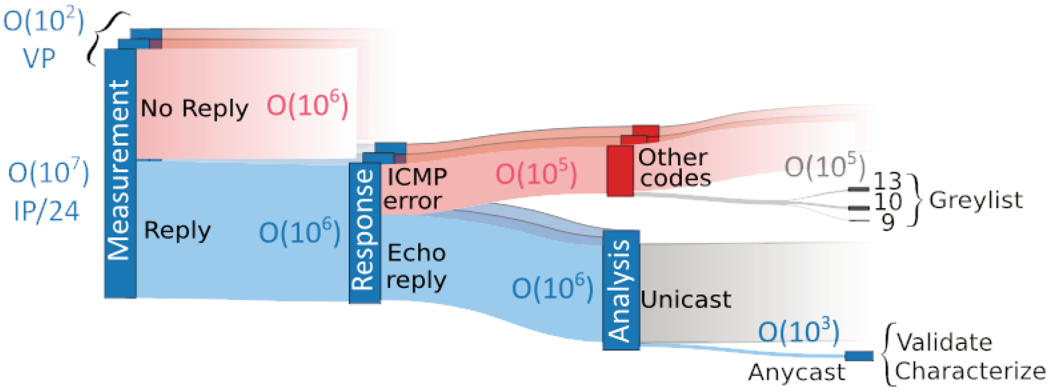


Figure 5: Ampleur typique du census anycast

l'ensemble, 1696 IP/24 appartenant à 346 AS semblent avoir plus d'une réplique anycast, alors que nous avons pu trouver seulement 897 IP/24 appartenant à 100 AS ayant au moins 5 répliques avec notre technique.

Top-100 Anycast ASes

Malgré d'abord le nombre de anycast IP/24 puisse ne pas sembler importante en raison de son empreinte exiguës, il est néanmoins très riche. De la vérification croisée avec les classements CAIDA et Alexa, nous nous attendons déjà à ce que l'utilisation d'anycast ne soit pas seulement limitée au DNS, mais couvre des ISPs et OTTs. Fig. 7 présente une vue d'ensemble de l'adoption de anycast, illustrant plusieurs informations pour les 100 AS pour lesquelles nous avons détecté au moins 5 répliques, identifiées par leur nom avec WHOIS rapporté dans l'axe des X inférieur. *Géographique et empreinte IP/24* sont indiqués dans la partie inférieure de la figure: AS sont disposés de gauche à droite, en nombre décroissant de répliques (barre inférieure graphique, avec écart-type sur IP/24 appartenant à la même AS), indiquant en plus le nombre d'anycast IP/24 pour cet AS (diagramme à barres du milieu). *Service footprint* est corrélé avec les ports TCP ouverts dans l'AS (middle scatter-plot). Ensuite, la *importance relative* de l'AS dans Internet et pour le Web sont exprimées en termes de CAIDA et d'Alexa respectivement (top scatter-plot). Enfin, une étiquette figurant sur l'axe des x supérieur catégorise

	IP/24	ASes	Villes	Pays	Répliques
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

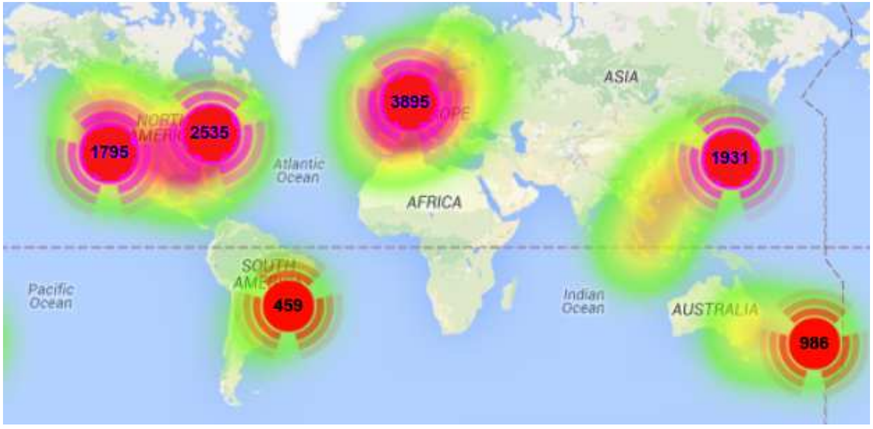


Figure 6: Anycast census results at a glance.

l'activité principale des AS d'un point de vue commercial.

Big fishes. Il est assez facile de repérer les principaux acteurs de l'écosystème Internet dans Fig. 7. La liste comprend non seulement les tier-1 et autres ISPs (tels que AT&T Services, Tinet, Sprint, TATA Communications, Qwest, Niveau 3, Hurricane Electrics), mais aussi un large éventail des OTTs tels que les CDNs (par exemple, CloudFlare , EdgeCast), hosting (par exemple, OVH) et les fournisseurs de cloud (par exemple, Microsoft, Amazon Web Services), les réseaux sociaux (par exemple, Twitter, Facebook, LinkedIn), et les sociétés de sécurité fournissant des services d'atténuation contre les attaques DDoS (OpenDNS, Prolexic). La liste comprend également des fabricants (par exemple, Apple, RIM), des registraires Web (par exemple, Verisign, nic.at), des services d'itinérance virtuelle et de réunions virtuelles (Media Network Services), des plateformes de blogs (Automattic, une société d'édition hébergeant

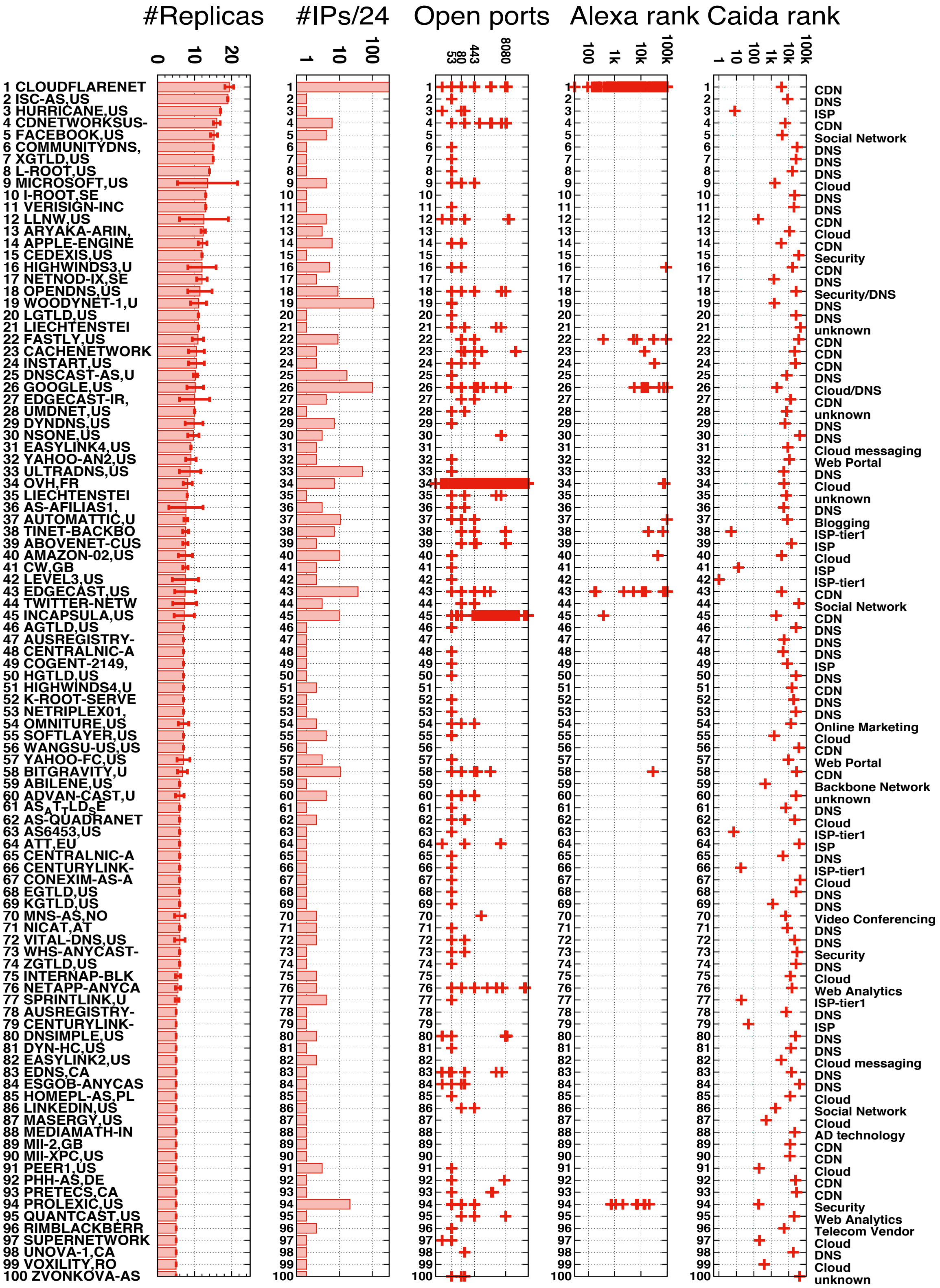


Figure 7: Bird's eye view of Top-100 anycast ASes (ranked according to geographical footprint)

wordPress.com) , la messagerie cloud (EASYLINK2 détenue par AT&T Services) et l'analyse Web (OMNITURE détenue par Adobe Systems). Bien sûr, les fournisseurs de services liés au DNS tels que les root et de premier niveau (par exemple, ISC/F-root, CommunityDNS), la gestion des services DNS (par exemple, UltraDNS, DynDNS) et les DNS publics (par exemple, Google DNS, OpenDNS) apparaissent également dans le recensement.

Services Anycast

Campagne de Portscan. En raison du mythe de longue date reléguant l'utilisation d'anycast aux services stateless, et en particulier DNS, nous pensons qu'il est important de fournir une vue longitudinale sur les services offerts par IP-anycast, en particulier en se concentrant sur TCP.

Nous testons tous les anycast IP/24 des 100 premiers AS et sélectionnons le seul IP/32 représentant par IP/24, nous scannons, à faible fréquence, tous les ports 2^{16} TCP. Nos résultats sont conservateurs car différents IP/32 peuvent avoir des ports ouverts différents (ce qui arrive, par exemple, pour CloudFlare et EdgeCast), et une sous-estimation du nombre de ports TCP ouverts peut aussi être le résultat du filtrage des sondes par les firewall et les routeurs long le chemin vers les cibles. Sur les 897 IP des 100 premiers AS, nous trouvons que 816 des 81 AS ont au moins un port TCP ouvert. Le nombre total de ports TCP ouverts distincts est de 10485, fournissant 449 services bien connus (c'est-à-dire, comme indiqué par la classification de port TCP), dont 170 sur SSL. De plus, les empreintes digitales nmap découvrent 30 implémentations logicielles différentes exécutées sur les réplicas anycast.

3.3 Évolution temporelle

Nous décidons de compléter l'image par une étude longitudinale (brève): depuis plus d'un an, nous effectuons régulièrement des recensements IP-anycast mensuels. Ainsi, nous fournissons une image large incluant tous les déploiements, ainsi qu'une vue plus détaillée en sélectionnant certains d'entre eux. Sans perte de généralité, nous nous référons à la dernière année des recensements collectés entre mai 2016 et mai 2017 et faisons tous nos jeux de données (mesures brutes de PlanetLab et RIPE Atlas), résultats (répliques anycast géolocalisées mensuelles pour tous IP/24) et code disponible à la communauté.

3.4 Campagne de mesure

Targets. Pour la sélection des cibles, nous comptons sur la liste de résultats USC/LANDER [6], fournissant une liste d'hôtes IP/32 cibles (vraisemblablement actifs) pour chaque préfixe /24. Tous les deux mois, la liste est mise à jour, et donc notre sélection cible. Nous considérons uniquement les adresses IP de hitlist qui ont été contactées avec succès (c.-à-d., notées par un score positif [6]), ce qui nous laisse environ $6,3 millions$ de cibles potentielles (sur 14,7 millions).

Vantage points (VP). Nous effectuons la sélection de VP comme suit: dans PlanetLab, où le nombre total de PV est petit (et décroissant), nous sélectionnons simplement tous ceux qui sont disponibles; Dans RIPE Atlas, où le nombre de PV est élevé et en raison de la limite de crédits, nous sélectionnons soigneusement 500 PV, en s'assurant que chaque VP est éloigné des autres d'au moins 200 km (environ 2 ms).

Censuses. Fig. 8 montre le nombre d'adresses IP uniques qui ont répondu à au moins un de nos VP PlanetLab dans chaque recensement, et l'axe des ordonnées à droite indique ce nombre comme la fraction des réponses du cibles contactées. La partie ombrée indique les mois où nous étions encore en train de mettre à jour le système. Nous pouvons voir que le nombre total d'IP uniques est toujours supérieur au nombre observé par un seul VP, et qu'il fluctue entre 2,9 millions (mai 2015) et 4,3 millions (nov 2016) de manière cohérente avec [61, 131]. Ce nombre a augmenté depuis juin 2016, date à laquelle nous avons commencé à mettre à jour régulièrement la hitlist

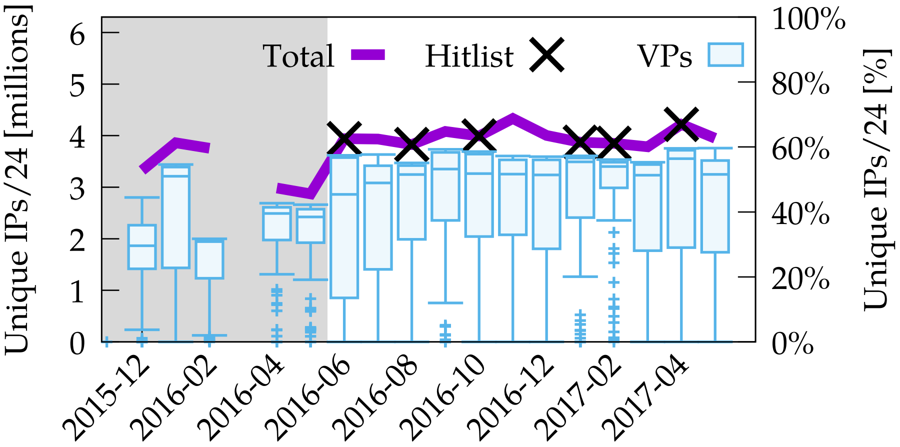


Figure 8: Campagne de mesure: box plot du nombre d'IP uniques /24 réactif sur l'ensemble des VP PlanetLab.

[6] (indiquée par des croix). Cependant, notez que même avec de nouvelles listes de résultats, toutes les cibles ne sont pas réactives qui corrèle avec le score de disponibilité des cibles: en particulier, le score moyen pour les cibles réactives (89) est supérieur au score des non-réactives (40). La figure indique également la répartition des IP sensibles par points de vue (box plots): le rappel varie considérablement par recensement, par VP et au fil du temps, certains VP pouvant recueillir seulement quelques centaines de réponses ICMP. Heureusement, bien que le nombre de VP PlanetLab diminue, le nombre médian de cibles contactées dépasse systématiquement 3 millions.

3.5 Résultats: broad view

Vue longitudinal. Premièrement, nous évaluons la variabilité des déploiements anycast. Nous commençons par considérer une granularité IP/24, et illustrons sur la Fig. 9 l'évolution du nombre de déploiements anycast IP / 24. La figure montre que dans nos recensements, le nombre de déploiements a légèrement augmenté (+10%) l'année dernière, culminant en avril 2017 à 4729 IP/24 appartenant à 1591 préfixes BGP acheminés et 413 AS. Au cours des six derniers mois, le nombre de déploiements d'anycast

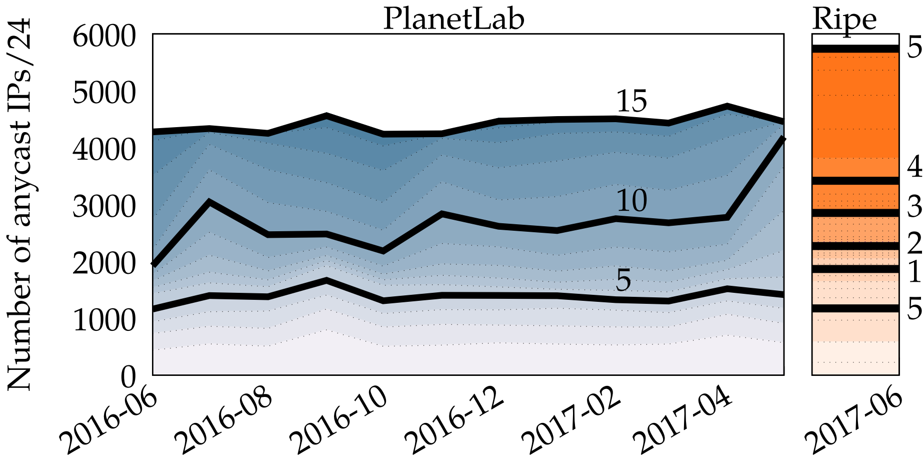


Figure 9: Vue longitudinale de l'évolution d'anycast: Nombre de déploiements IP/24 anycast (axe Y) et répartition de leur empreinte géographique (heatmap et courbes de niveau) dans PlanetLab (à gauche, au cours de la dernière année) vs RIPE Atlas (à droite, le mois dernier).

n'a jamais été inférieur à 4500 alors qu'en juin 2016, lorsque nous avons commencé les recensements régulièrement, nous n'avons trouvé que 4297 IP/24, 1507 préfixes BGP et 379 AS. Par rapport à l'étude spatiale de mars 2015 présentée précédemment, cela représente une augmentation de 2,5 fois des occurrences anycast détectées sur une période de 2 ans.

3.6 Résultats: focused view

Top-10 déploiements. Nous fournissons maintenant une vue plus détaillée de quelques AS sélectionnés. Tab. 2 affiche les détails concernant les 10 premiers déploiements (nom et type de société, numéro AS et nombre de préfixes BGP annoncés par cet AS), l'empreinte spatiale (ie, le nombre S_A de IP/24 par AS et sa variabilité temporelle) et l'empreinte géographique (c'est-à-dire le nombre G_A de répliques distinctes et sa variabilité temporelle). Considérant l'empreinte spatiale IP/24, Cloudflare (AS13335) a un rôle prépondérant: il est présent dans tous les recensements avec plus de 3000

Empreinte de déploiement:				Spatial		Géographique	
Compagnie	AS	Type	BGP	S_A^+	CV_S	G_A^+	CV_G
Cloudflare	13335	CDN	206	3016	0.04	49	0.07
Google	15169	SP	16	524	0.38	30	0.08
Afilias	12041	TLD	218	218	0.15	6	0.10
Fastly	54113	CDN	34	175	0.09	20	0.07
Incapsula	19551	DDoS	146	146	0.23	15	0.17
Cloudflare	13335	CDN	206	3016	0.04	49	0.07
L root	20144	DNS	1	1	0	47	0.13
F root	3557	DNS	2	2	0	40	0.19
Woodynet	42	TLD	132	133	0.02	39	0.12
Verisign	26415	Reg.	2	2	0	36	0.20

Table 2: Vue focalisée: variabilité de l'empreinte des cinq premiers déploiements géographiques (en haut) et top-5 (en bas).

IP/24 appartenant à environ 200 préfixes annoncés (principalement /20 mais aussi des préfixes moins spécifiques, comme /12 ou /17), et nous n'avons pas observé de variation significative au fil du temps.

Variabilité temporelle. Nous inspectons maintenant la variabilité temporelle à un grain plus fin. Nous commençons par illustrer dans Fig. 10 l'évolution temporelle de l'empreinte spatiale, normalisée sur le maximum observé pour ce déploiement (ie, $S_A(t)/max_t S_A^+(t)$) pour les top 5 (Afilias, Google) ainsi que pour d'autres acteurs Internet clés (Microsoft, Akamai, Netflix, Windstream). Les évolutions représentent un échantillon de ce que on peut trouver dans nos recensements: par exemple les deux AS rapportés dans l'image appartenant à Akamai (AS21342) et Neflix (AS40027)

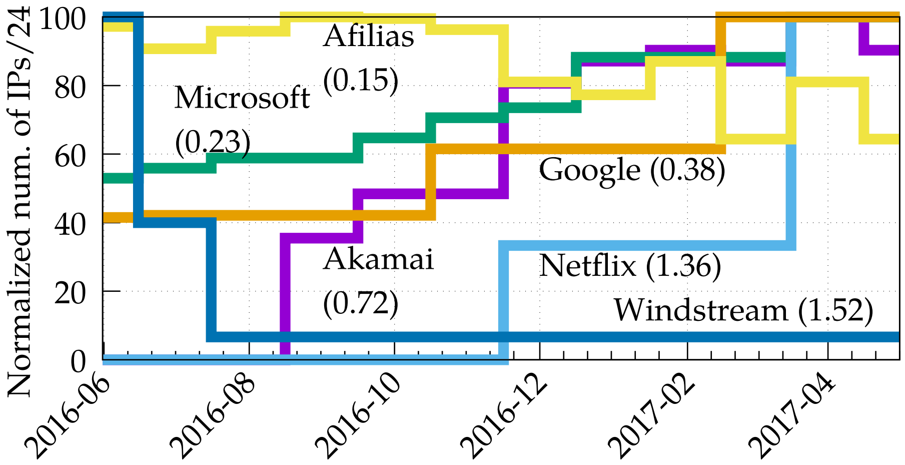


Figure 10: Évolution spatiale: nombre d'IP/24 pour les déploiements Anycast AS sélectionnés (PlanetLab).

commencent à être annoncés comme anycast pendant notre période d'observation et soit systématiquement (Akamai) soit abruptement (Neflix) augmentent la quantité d'anycast IP / 24 réactif au fil du temps. Google (AS15169) et Microsoft (AS8068) ont tous deux une présence importante au début de la période d'observation, avec environ 50% de l'IP/24 déjà utilisé, et environ le double de l'IP/24 utilisé à la fin de la période d'observation. la période d'une manière lisse (Microsoft) ou abrupte (Google). Enfin, près du début de notre période d'observation, Windstream réduit considérablement son empreinte spatiale anycast, ne conservant qu'une seule IP/24 non diffusée. Si ces observations ont une valeur anecdotique et ne peuvent expliquer la raison des changements dans le déploiement, elles confirment toutefois que les déploiements d'anycast ont une évolution temporelle plutôt vive, dont l'ampleur est captée par le coefficient de variation. Il convient de rappeler que le déploiement individuel présente de grandes variations, mais l'agrégat reste assez stable dans le temps (rappel Fig. 9).

4 Caractérisation du trafic Anycast IPV4 à partir de traces passives

Dans la partie précédente, nous proposons une large caractérisation des déploiements anycast. Cependant, les études actives présentées ne sont pas en mesure de fournir certaines informations, telles que la popularité des services anycast, le volume de trafic qu'ils attirent et quelles applications ils servent, cela ne peut être recueilli que par une mesure passive. Dans ce qui suit, nous effectuons une étude de caractérisation passive et nous nous concentrons sur les principaux acteurs Internet qui ont commencé à adopter cette technologie pour diffuser du contenu Web via des CDN activés par Anycast (A-CDN).

4.1 Méthodologie

Listes de sous-réseaux Anycast. En un mot, notre flux de travail extrait d'abord le sous-ensemble des flux dirigés vers des serveurs anycast, puis caractérise le trafic qu'ils échangent avec les utilisateurs actuels d'Internet en tirant parti des mesures passives. À partir de la liste anycast, nous extrayons un ensemble plus *conservatif* de 897 sous-réseaux, ayant au moins cinq emplacements distincts.

Moniteur passif. Nous avons instrumenté une sonde passive sur un PoP d'un réseau opérationnel dans un ISP national. La sonde exécute Tstat [74], un outil de surveillance passif qui observe les paquets circulant sur les liens reliant le PoP au réseau backbone ISP. Tstat reconstruit en temps réel chaque flux TCP et UDP en temps réel, et lorsque le flux est interrompu ou après une temporisation inactive, il enregistre plus de 100 statistiques. Pour cette étude, nous etudions l'ensemble de données recueillies pendant tout le mois de mars 2015. Il se compose de 2 milliards de flux TCP surveillés, pour un

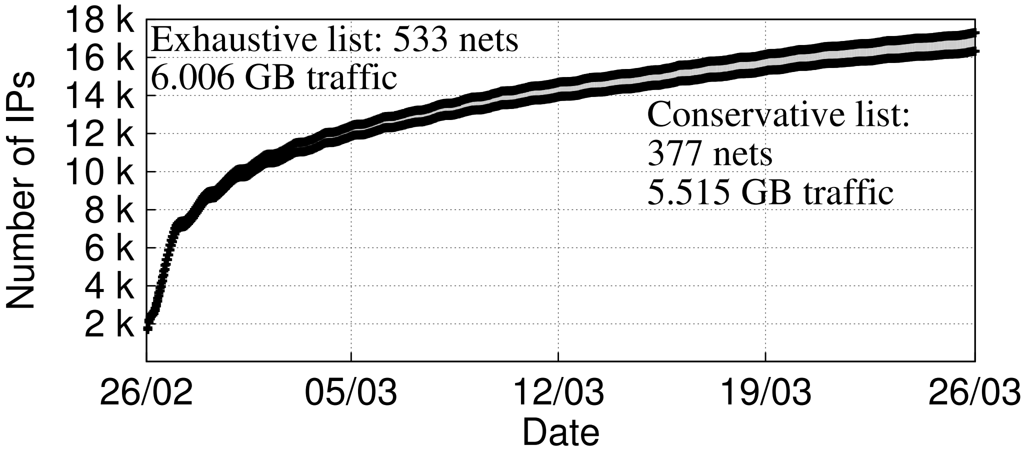


Figure 11: Nombre cumulé de serveurs distincts rencontrés au cours du mois.

total de 270 TB de trafic réseau. 1,4 milliard de connexions sont dues au web (HTTP ou HTTPS) générant 209 TB de données. Parmi les nombreuses mesures fournies par Tstat pour chaque flux TCP, nous nous concentrons uniquement sur: (i) l'adresse IP du serveur; et (ii) l'adresse IP du client anonymisée; (iii) le temps de parcours aller (RTT) minimal entre la sonde Tstat et le serveur; (iv) La quantité d'octets téléchargés; (v) le protocole de couche d'application (par exemple, HTTP, HTTPS, etc.); (vi) Le nom de domaine complet du serveur que le client contacte tel qu'il a été renvoyé par DN-Hunter.

5 Results: broad view

Résultats. Les résultats confirment l'empreinte du trafic anycast, et A-CDN en particulier, aux heures de pointe la probabilité de contacter au moins un serveur anycast est supérieure à 50% et nous observons plus de 16 000 adresses anycast IP distinctes.

Fig. 11 indique le nombre d'adresses uniques IP-anycast cumulé observées dans le temps, toujours pour les listes conservative et exhaustive. En résumé, la plupart des /24 hôtes quelques serveurs, qui sont en général assez proches du PoP (RTT <10 ms), utilisent 2 ou au plus 3 protocoles (HTTP ou HTTPS principalement). La diversité commence à apparaître dans le nombre de noms de domaine complets (FQDN), certains d'entre eux étant utilisés pour une poignée de services, tandis que d'autres en desservent plusieurs milliers. Le volume servi varie considérablement. De même, alors que seulement la moitié des débits dépasse 50 kB, et la plupart sont inférieurs à 1 MB, la taille du flux atteint plusieurs centaines de MB. Vu la popularité, certains /24 sont utilisés par plusieurs milliers d'utilisateurs finaux, d'autres par moins de 10.

6 Conclusion

Internet fonctionne grâce à un écosystème d'entreprises, d'organisations et de communautés. C'est un environnement en constante évolution utilisé pour de nouvelles choses chaque jour.

De nos jours, les utilisateurs accèdent à une grande variété de contenus (par exemple, logiciels, vidéos et images) via Internet et attendent une certaine qualité d'expérience. Pour répondre à ces demandes, les organisations impliquées utilisent des technologies de pointe pour mettre en œuvre différentes solutions. Un exemple est la duplication de contenu à divers points du globe pour une récupération ultérieure. Ce mécanisme permet de réduire le trafic du réseau et de garder les utilisateurs satisfaits.

Cette thèse sensibilise et met en lumière un autre de ces mécanismes: anycast. Dans la plupart des déploiements anycast, les contenus sont répliqués sur plusieurs serveurs. Cela comporte de nombreux avantages, par ex. améliorer la fiabilité et résoudre les problèmes d'évolutivité. Certaines entreprises utilisent anycast comme première couche de défense contre les attaques par déni de service distribué (DDoS).

La première contribution de cette thèse est la conception d'une méthodologie. La méthodologie est capable de découvrir, d'énumérer et de géolocaliser des instances IP-anycast. Nous lançons sa mise en œuvre en tant que logiciel open source, iGreedy.

La deuxième contribution est la conception et la mise en œuvre d'un système qui permet de découvrir les acteurs IP-anycast à travers divers recensements sur Internet. Leur évolution est présentée avec une étude longitudinale. Les mises à jour mensuelles sont publiées via une liste d'adresses IP anycast, y compris leur empreinte et leur géolocalisation. De plus, les résultats sont intégrés dans le projet RIPE Atlas, OpenIPmap [31].

Enfin, une étude passive a été menée pour observer à la fois l'utilisation et la stabilité des CDN anycast, dont les résultats démontrent à la fois la popularité et la fiabilité des services anycast.

Contents

Chapter 1

Introduction

Back in the 1989, I was born in a small town in Italy and at that time the Internet was fully prospering: few months before the first dedicated transatlantic satellite communication was established and half million of people were connected to the Internet thanks to the first Internet Service Provider.

Ten years later, I received the first dial-up modem, a beatiful orange apparatus that uses the telephone network to provide Internet access and allowed me to navigate at the incredible speed of 56Kbs. Unforgettable its glorious sound protocols, starting with dialing, that has millions of views on YouTube [5].

Nowadays, the situation is evolved and our expectations are changed: we use the Internet for multiple activities such as send or read emails (e.g. Gmail), use search engines to find information (e.g. Bing), watch video in streaming (e.g. Netflix) and use social networks (e.g. Twitter, LinkedIn, Facebook). Yet, all the Internet players have to keep up using cutting-edge technologies to provide a certain Quality of Experience (QoE) and if only one of them is affected by an outage, it might affect the entire Internet ecosystem.

There are many organizations that make the Internet works as depicted in Fig. 1.1, among all, there are companies that provide network infrastructure services. They are

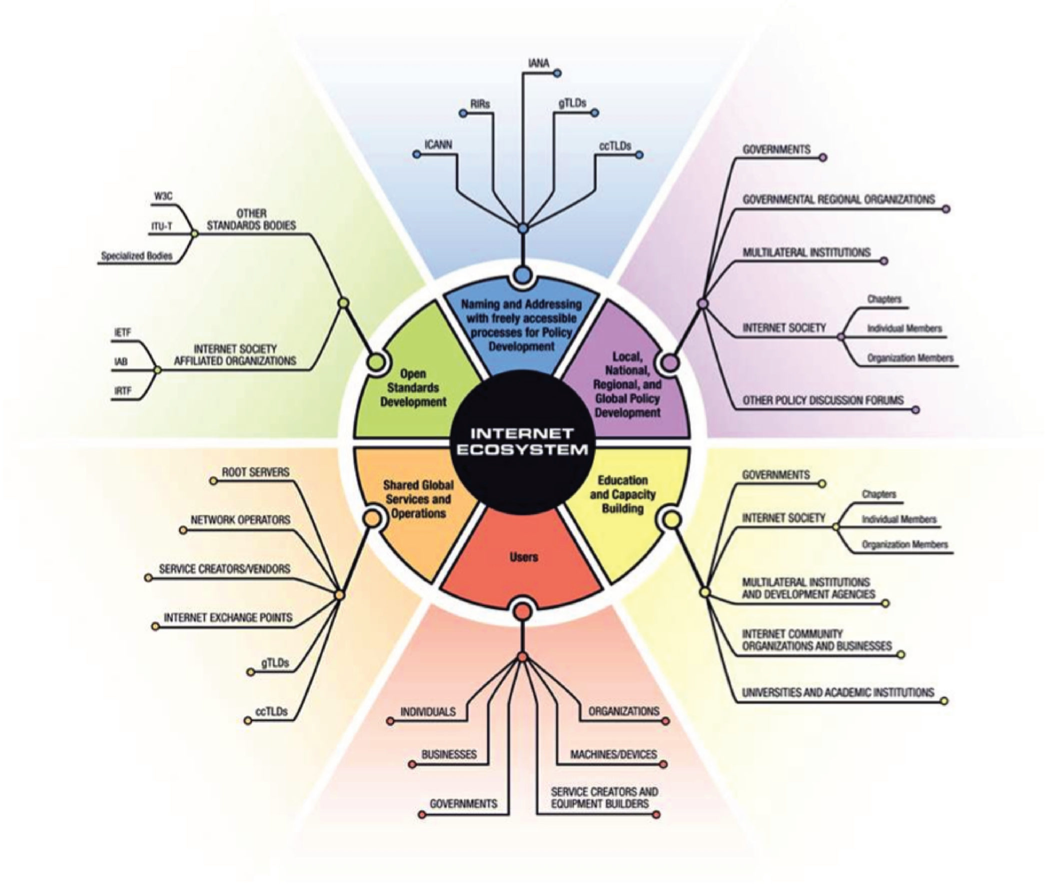

IANA
RIRs
gTLDs
ICANN
ccTLDs
W3C
ITU-T
Specialized Bodies
OTHER STANDARDS BODIES
IETF
IAB
IRTF
INTERNET SOCIETY AFFILIATED ORGANIZATIONS
Naming and Addressing with freely accessible processes for Policy Development
Open Standards Development
Local, National, Regional, and Global Policy Development
INTERNET ECOSYSTEM
Shared Global Services and Operations
Education and Capacity Building
Users
GOVERNMENTS
GOVERNMENTAL REGIONAL ORGANIZATIONS
MULTILATERAL INSTITUTIONS
INTERNET SOCIETY
Chapters
Individual Members
Organization Members
OTHER POLICY DISCUSSION FORUMS
GOVERNMENTS
INTERNET SOCIETY
Chapters
Individual Members
Organization Members
MULTILATERAL INSTITUTIONS AND DEVELOPMENT AGENCIES
INTERNET COMMUNITY ORGANIZATIONS AND BUSINESSES
UNIVERSITIES AND ACADEMIC INSTITUTIONS
ROOT SERVERS
NETWORK OPERATORS
SERVICE CREATORS/VENDORS
INTERNET EXCHANGE POINTS
gTLDs
ccTLDs
INDIVIDUALS
BUSINESSES
GOVERNMENTS
ORGANIZATIONS
MACHINES/DEVICES
SERVICE CREATORS AND EQUIPMENT BUILDERS

Figure 1.1: The Internet ecosystem [9]

responsible of daily operations of the Internet:

- the Domain Name Service (DNS), the Internet's phone book that translates domain names to IP addresses;
- network operators manage Autonomous Systems (AS) and Internet Service Providers (ISP) providing Internet access to their clients;
- cloud and content delivery network providers (CDN) host and distribute third party contents;

- Internet Exchange Points (IXPs) that are points where ASes and CDNs exchange Internet traffic between their networks.

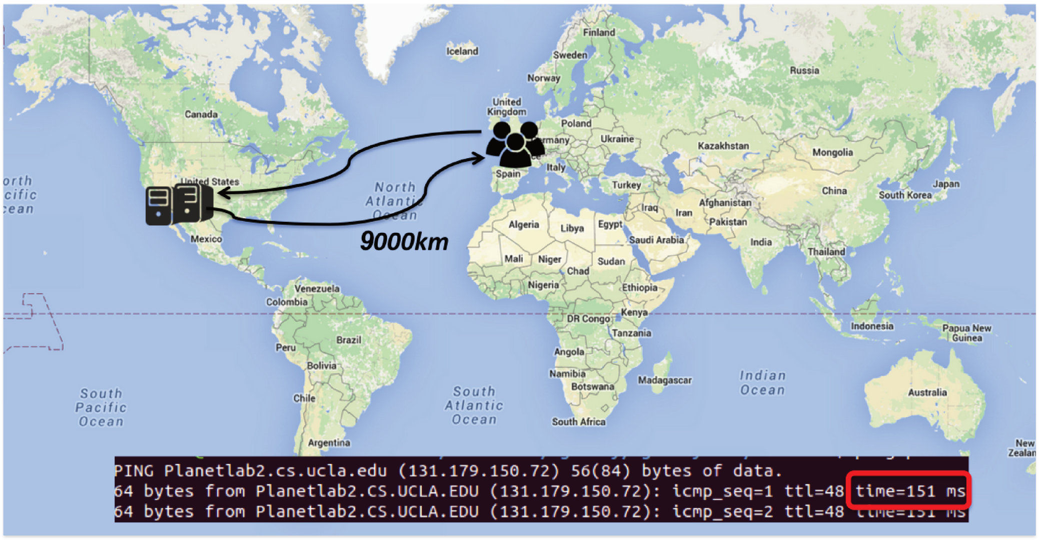
All of them contribute to the QoE of the users that we should not underestimate: studies in the past [49, 99] have highlighted the importance of the performance and availability of the contents. Other studies [64, 82, 109, 119, 122] show how these factors have an economic impact for the companies.

1 The importance of latency

The user-perceived latency, in particular, has an important role: Amazon claims that 100ms latency penalty implies a 1% sales loss [99], Google that an additional delay of 400ms in search responses reduces search volume by 0.74% [49] and Bing that 500ms of latency decreases revenue per user by 1.2% [101].

Thus, companies focus their attention on finding ways to reduce the latency that is mainly due to protocol overheads and infrastructural limits, packets cannot travel at the speed-of-light in the fiber channel. A recent study [121] investigates this phenomena and found the today's Internet to be more than an order of magnitude slower than the speed-of-light. Fig. 1.2 shows a real example of round trip time between a client located in Paris and a host in Los Angeles, the packets takes 150ms, a huge latency for the companies.

Hence, the companies need to place the information closer to the users, but while they are spread all over the world, the only solution for them is to replicate the information in location geographically dispersed and redirect the users to the closest one (in terms of latency).



```
PING Planetlab2.cs.ucla.edu (131.179.150.72) 56(84) bytes of data.
64 bytes from Planetlab2.CS.UCLA.EDU (131.179.150.72): icmp_seq=1 ttl=48 time=151 ms
64 bytes from Planetlab2.CS.UCLA.EDU (131.179.150.72): icmp_seq=2 ttl=48 time=151 ms
```

Figure 1.2: Real latency between a client and a host located respectively in Paris and in Los Angeles in an academic network.

2 Anycast

The Internet offers different addressing methods that we present in Sec. 1, and one of these, *anycast* [112], meets the needs of companies: at the network level with *IP-anycast*, multiple hosts share the same IP address and the client contacts one of the anycast replica delegating the choice to the network, usually the closest in terms of BGP distance. The common way to deploy IP-anycast is to announce an IP prefix from multiple points using the same AS [87], that we refer to Single Origin AS (SOAS). Another way is to announce the IP prefix using multiple ASes, usually referred as Multiple Origin ASes (MOAS) prefixes.

IP-anycast is an attractive solution: it is very cheap, simple to deploy (i.e., avoiding the need to manage some custom and complex application-layer solution), provides users with enhanced QoE (e.g., for services where we want to cache content close to the user) and brings several advantages for the service provider (i.e., load balancing among replicas, robust to DDoS attacks in reason of geographic traffic confinement, etc.).

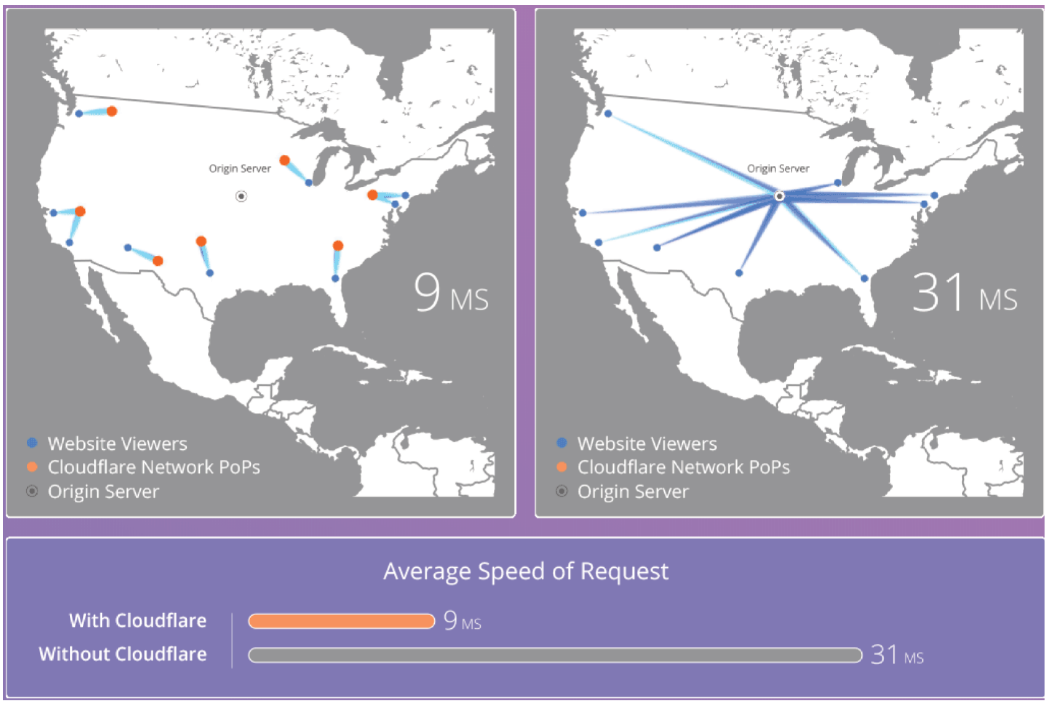


Figure 1.3: Cloudflare CDN: bringing contents closer to visitors [58].

This technique has been widely used in the past by the DNS servers, yet in this thesis, we unveil that in the recent years many other Internet key players adopted anycast and with this rise, there is a concomitant need to understand anycast services. We find video streaming company as Netflix, Content Delivery Networks (CDN) such as EdgeCast [1], now part of Verizon, and CloudFlare [16], that advertise to serve respectively 1.5 billion objects per hour representing the 4% of the whole Internet traffic [10] and over 2 million Web sites [58]. Social networks as Twitter [127] that manages more than 500 million tweets per day or web search engine as Bing owned and operated by Microsoft that serves 5 billions monthly request and represents the 34% share in the U.S. desktop search market [107].

Chapter	Subject	Publication
Chap. 3	Anycast detection, enumeration and geolocation	[C1, J1, D1]
Chap. 4	Anycast IPv4 censuses and active characterization	[C2, M1]
Chap. 5	Passive anycast characterization	[W1]

Table 1.1: Synopsis of the thesis

3 Contributions of this thesis

This thesis focuses on IP-anycast, we resume our contributions in Tab. 1.1. Our interest was motivated by the increasing adoption of IP-anycast on one hand, and, on the other by the lack of studies in the field: in the literature the researchers mainly focused on DNS and their performance [47, 72, 92, 100, 110, 118, 118]. The knowledge of IP anycast can be of interest to scientists in a broad span of fields: from characterization, troubleshooting and infrastructure mapping [31] but also for security-related tasks such as censorship detection [113]. Unfortunately, the publicly available information regarding anycast companies are often unknown [104] or in case they are available, such as DNS, are often outdated [72].

Thus, we decide to shed light on this topic: trying to be good scientists, we addressed our research trying to answer the *Five Ws and one H* questions. These are questions that help in gather information: the answers build a complete view of the IP-anycast. In the following, we list the questions that provide an overview of the topic.

3.1 Why did companies start to adopt anycast?

Critical Internet infrastructure such as DNS, needs to provide reliable services and be ready for any failure or attack. One of the solution they adopted to reduce the risk is to replicate their servers in multiple physical sites announcing the same IP using

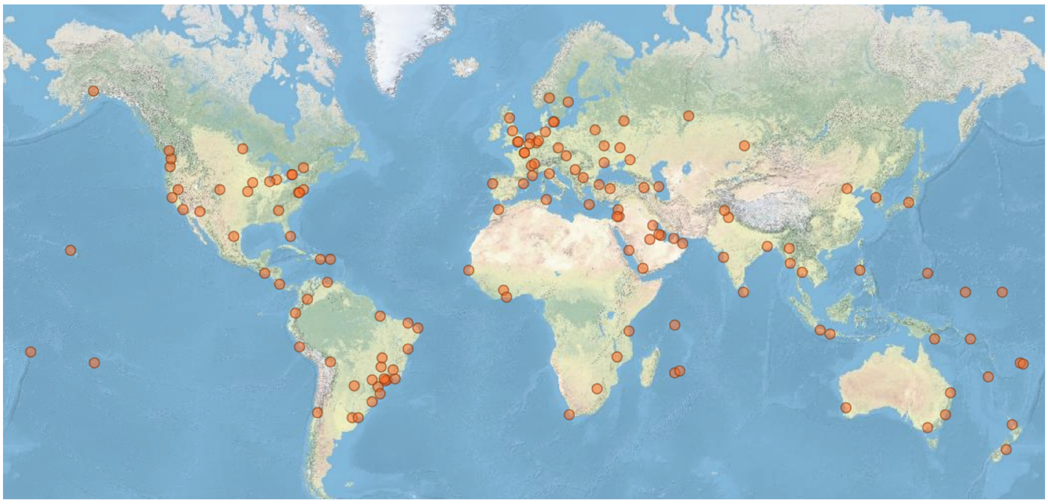


Figure 1.4: L-root DNS server locations [21]

anycast. On the other hand, one of the major threat of the Internet are the distributed denial-of-service (DDoS) attacks that try to overwhelm the target with requests or other traffic from multiple vantage points. In the past, the root DNS, 13 independent services present in more than 500 location, have been the target of numerous DDoS attacks [2,89,110]: they created some damages but thanks to anycast, they didn't disrupt the service.

IP-Anycast not only offers a defensive method against the DDoS attacks, but it offers an easy deploy of geographically distributed replica server, easy management delegating the replica selection to the BGP routing, solves scalability problems distributing the load among the different replicas and reduce, on average, the user-perceived latency. These are the reasons behind the IP-anycast adoption by other companies: CacheFly [50], a CDN provider, was among the CDN anycast pioneer. It started to use it in the 2002 announcing prefixes from three different continents.

In Chapter 2, we provide more details on the Internet addressing methods and more insights on the different anycast implementations.

3.2 Where is an anycaster?

Commercial IP geolocation database services such as MaxMind are very popular. They can be accurate in popular countries (e.g. US), less precise in other countries (e.g. third world countries) [115], inaccurate for infrastructure IPs [104] and fail with IP-anycast. They report only one location for an IP and, as we explained before, the anycast IPs are announced in multiple location by definition. Fig. 1.4 shows the different L-root server locations, the same IP is announced from more than hundred different places.

One explanation for the defecit in the anycast geolocation is that current services are oriented towards geolocation of client machines, not servers. Content distributors might geolocate clients in order to respect contractual constraints that limit them to sending certain content to certain countries; banks might geolocate clients for security purposes and as part of their due diligence concerning knowledge of their customers; advertisers might geolocate clients in order to profile populations and deliver targeted messages.

Prior work on identifying, enumerating, and geolocating anycast services [72] has mainly focused upon DNS. Thus, the techniques used are specific to DNS and are not generalizable. Others [50, 104] investigate the adoption of anycast, however their technique is limited to the only detection of anycast instances. They discover the anycast deployments detecting speed-of-light violations via latency measurements: i.e., as packets travel slower than speed of light, an US and EU host probing the same target cannot both exhibit excessively low latency (e.g., few milliseconds), as this would violates physical laws.

With our work, presented in Chapter 3, we make steps forward improving the technique: we propose a new method for exhaustive and accurate enumeration and city-level geolocation of anycast instances, requiring only a handful of latency measurements from a set of known vantage points.

We also provide the community open-source software, iGreedy [22], and datasets allowing others to replicate our experimental results, potentially facilitating the development of new techniques such as ours.

3.3 Who are all the anycasters?

Previously, most of the study on IP anycast focused on DNS [42,100,118], which was the main application of IP-anycast. However, few companies [1, 16, 50] were advertising to use IP-anycast for non-DNS services and this was a signal that the situation was changing.

Nevertheless, little was known about the IP-anycast players and thus we decide to discover who are the anycasters and where they place their instances doing IPv4 Internet anycast censuses. This has required to face several challenges: we had to scale the technique to perform Internet censuses; coordinate a large number of vantage points avoid hitting ICMP rate limiting and avoid hurting the destinations; collect and analyse a large amount of measurements.

In our results, presented in Chapter 4, we find the situation significantly changed. In particular, major players of the Internet ecosystem including ISPs (e.g. Level 3, Hurricane Electrics, ATT), Over-The-Top media services (e.g. Microsoft, Incapsula, CloudFlare), and manufacturers (e.g. Apple, RIM) provide a diversity of services with IP-anycast (e.g., content distribution, cloud services, web hosting, web acceleration, DDoS protection). Of course, we also find DNS root and top-level domain servers and other DNS service management. Finally, we find that the companies put the anycast replica where people live. This follows from the fact that the decision to add an anycast replica, follows from the goal of ameliorating performance for a large fraction of users, which live in large cities (interestingly, this was already used to bias geolocation of *unicast* addresses [69]).

3.4 How stable are the anycast deployments?

Once we found all the companies that uses IP-anycast, we wondered if they are using it for a long period of time. Thus, we decide to re-engineer our system, improving performance and reliability, and start to perform monthly anycast censuses. We automated the whole process from the Internet scanning to the data analysis publishing the geolocation results through a web interface. Collaborating with RIPE NCC, we also integrated our dataset in their geolocation service [27].

We continue to perform monthly censuses, but in our study, we focus on the anycast evolution in the period between June 2016 and May 2017. The total number of ASes that uses anycast is stable, yet if we look closer we can observe a heterogeneous situation: new players appear (e.g. Netflix) others leave (e.g. NetDna), some companies increase their geographical footprint (e.g. Google, Microsoft) and other reduce the number of anycast IPs (e.g. Windstream). We give more insights on this topic in Chapter 4.

3.5 What are the services they provide?

The companies use anycast to provide different services. Historically, IP-anycast has been used for services over User Datagram Protocol (UDP) such as DNS, root and top level domain servers, 6-to-4 relay routers, multicast rendezvous points, and sinkholes. When considering stateful services, the usage of IP anycast has been discouraged primarily due to its lack of control and awareness for the server and network load. Indeed, IP-anycast relies on BGP routing, meaning that any routing change could re-route the traffic to another server. This could break any stateful service, e.g., causing the abortion of TCP connections, and could cause the dropping of any application layer state. Moreover, the relatively slow convergence of routes and the purely destination based routing in IP make difficult design reactive systems where traffic can be arbitrarily split among multiple surrogate nodes. Over the years however, several studies proposed techniques to overcome these issues, showing that it is possible to leverage anycast for connection oriented services [36, 37, 75]. Thus, we leverage passive measurements, and offer a characterization of traffic served. We find a non-negligible part represented by HTTP/HTTPs services and a tiny part that uses anycast for other purposes such as multimedia streaming, email protocols or Peer-to-Peer traffic. We provide the details in Chapter 5, where we collect and analyze the traffic served in real networks.

3.6 When does a company need to adopt anycast?

It is not possible to answer this question in few lines. We hope the reader will get an opinion reading the thesis, we report further details in the discussion in Chapter 7.

Chapter 2

Background

In this chapter, we introduce the terminology used throughout the thesis. We start reviewing the different Internet addressing methods (Sec. 1), we then focus on the different anycast implementations (Sec. 1.1) and underline their benefits and limits (Sec. 1.2). We then describe the Internet measurements platforms focusing on their footprint and coverage (Sec. 2). Finally, in the last section we provide a broad view of the previous and ongoing community efforts in the anycast field (Sec. 3).

1 Internet addressing methods

The Internet offers a plethora of services and to meet their requirements, it provides different addressing methods. Each communication is characterised by the number of senders and receivers. Fig. 2.1 shows the four main addressing methods that we describe in the following.

Unicast: send to this one address. This is the most used addressing method in the Internet. The communication is between only one sender and one receiver. The packet is sent from a single source to a specified destination. Each receiver is identified by a unique destination address.

Broadcast: send to all the addresses. The communication is between one sender and multiple receivers. The packet is sent from a single source to all the possible destination at the same time. In IPv4 there is a special address used as broadcast destination. However, in the past the broadcast address has been exploited to perform DoS attacks and for this reason it is not present in IPv6.

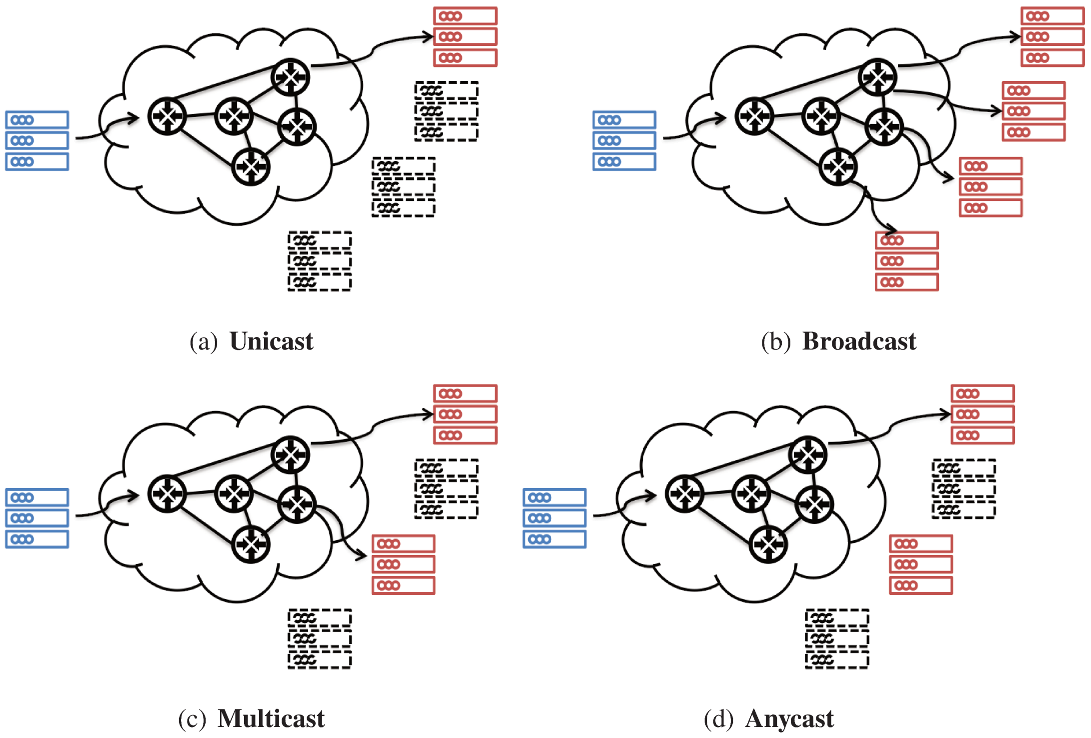
Multicast: send to every member of this group. The communication is between one or more senders and a group of receivers. In this case, there may be one or more senders and the information is distributed to a set of receivers. However, it also happens that unicast packets are sent incorrectly to multicast addresses and malicious attackers have been exploited this to perform DoS attacks and achieve packet amplification.

Anycast: send to any one of the member of this group. The communication is between one sender and one receiver, as in the unicast case. The difference is that, in this case, all the receivers share the same identifier. The most common used implementation are at network level (L3-anycast) and application level (L7-anycast).

1.1 L3 and L7 Anycast

At the network level, we need to distinguish between IPv4 and IPv6 implementations. In both cases, we refer to it as L3-Anycast or *IP-Anycast* [32]. In IPv4, a group of hosts, usually geographically distributed, share a common IP address. When a client contacts an IP-Anycast server, the packets are thus routed at the network layer to the closest address according to the BGP policies. The policies route packets sent to this address to the nearest replica according to BGP metrics, notably (though not only) the AS business relationships and the number of AS hops. This method still works in IPv6 [85], further, for each subnet prefix, there is a set of reserved anycast addresses [91].

An alternative is the implementation of anycast at the application level. *L7-anycast* relies on IP unicast and exploits DNS and HTTP redirection techniques to direct traffic



(a) **Unicast**

(b) **Broadcast**

(c) **Multicast**

(d) **Anycast**

Figure 2.1: Illustration of the Internet addressing methods: on the left, the sender (blue) and on the right, the possible receivers (red) and other hosts (dashed lines).

from a client to any server in a group of geographically dispersed servers. In both cases, the server (DNS or HTTP) tries to map the user to the closest server using some geolocation databases (e.g.,, maxmind) and perform load balancing among nearby replicas.

1.2 Advantages and limits of anycast implementations

The adoption of anycast, as stated before, introduces several advantages: it reduces the distance between the user and the destination, balance the load between the servers and increases the reliability. However, IP-anycast and L7-anycast are two different implementations and both options have advantages and disadvantages that we discuss in the following.

IP-anycast is natively supported by IP. It is easy to deploy, only needs to have a set of servers announcing the same IP, and allows to control the replica visibility through the *BGP No-Export* option that ensures the announcement to not get advertised outside the AS. This is counterbalanced by its limits such as the delegation of the server selection to BGP. The BGP policies determine which replica a host contacts (a single host might become overloaded with user traffic), which is difficult to control and therefore, does not guarantee server affinity (e.g., connection-oriented services).

L7-anycast solves these issues using the DNS (or HTTP) redirection: it offers a fine-grained control over the anycast server selection. Since it allows to decide which replica the user contacts, it is also easier to distribute the load between the servers. On the other hand to perform a good selection, the DNS resolver has to collect numerous metrics (e.g., list of servers up, latency, load) and the deploying requires more engineering work.

In conclusion, 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 [105]. An example of L7-anycast is Akamai [34] that has a very dense deployments: it claims to have more than 200 thousands servers spread across the Internet and to serve more than 30% of the Internet traffic.

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 BGP routing. CacheFly [50] was among the IP anycast pioneers followed then by other companies including EdgeCast [1], CloudFlare [16] and Fastly [20] that prefer to have fewer locations with more powerful servers.

Finally, there are other companies that have started to adopt both, L3 and L7, anycast deployments. Microsoft recently proposed an architectural improvements that

Table 2.1: Internet measurement infrastructures: footprint and geolocation meta-data

Infrastructure	**Footprint and coverage**			**Geolocation**			**Used in**
	VP	**AS**	**Country**	**Information**	**Granularity**†	**Validation**	**this work**†
RIPE	10k	2k	150	(lat,lon)	I	MaxMind [24]	✓
PlanetLab	250	180	30	(lat,lon)	I	Spotter [96] (for EU)	✓
Archipelago	208	100	63	City-level	I	Oral communication	⋆
M-Lab	1.5k	35	30 (1/2 in US)	(lat,lon)	I	Single owner	⋆
Dasu	100k	1k	150	Continent	A	n.a.	×
SamKnows-FCC'13	190k			US State	A	n.a.	×
SamKnows-OFCOM'14	2k			Rural vs Residential	I	n.a.	×

†Legend: used in this work (✓); open so possible in principle (⋆); proprietary so not possible (×); Granularity: Individual VP (I); Aggregated (A)

combine both approaches to address the performance shortcomings of IP-anycast in terms of scalability and server selection [75].

2 Measurement Infrastructures

In Chapter 3, we introduce our technique to detect, enumerate and geolocate the anycast deployments. We rely on latency measurements collected from different *measurement infrastructures* (MI) that allow us to use multiple distributed *vantage points* (VPs). A fairly large number of MI exist in the current Internet. Our technique can be applied from any such MI, it is thus interesting to analyze the MI candidates, which we list in Tab. 2.1. Given our aim, we put special emphasis on the MI footprint and coverage, and additionally annotate with meta-data concerning the geolocation accuracy of individual VPs.

The table reports a number of open (e.g., RIPE Atlas, PlanetLab, Archipelago, MLab, etc.) and some proprietary (e.g., Dasu, SamKnows, etc.) measurement infrastructures. Due to obsolescence[1] of such information, we prefer to "round" numbers collected from the respective pages, to imply that only orders of magnitude are (at time of writing) or will be accurate (at time of reading). Notice also that while in this thesis we are mostly concerned with geolocation accuracy of MI vantage points, a recent and more complete survey of measurement infrastructures is also available for the interested reader [40].

The geolocalization of each vantage point in such MIs is typically reported by the person who hosts the measurement agent, and these reports can be verified through unicast geolocalization services [69,70,80,115,120], to check for initial accuracy and to catch cases when an agent is moved to another location. For instance, PlanetLab Europe performs an additional validation of the VP location with Spotter [96]. Geolocalization of RIPE Atlas VPs is checked through MaxMind [24], and tags additionally report information about the accuracy of geolocalization for (a growing number of) VPs. In few cases we spotted inconsistent location of VPs, that we validate via manual inspection[2].

In this thesis, we use RIPE Atlas [26] and PlanetLab [25] which are interesting due to their complementarity aspects. On the one hand, RIPE Atlas has a larger footprint in terms of number of VP (over 40 times larger than PlanetLab) as well as a better geographical (10 times) or AS (5 times) coverage, from which we expect RIPE Atlas to provide a more exhaustive coverage than PlanetLab. On the other hand, RIPE Atlas is more constrained than PlanetLab in the type and rate of measurement that can be performed: for instance, no HTTP measurement are allowed from RIPE Atlas, limiting the space of actions. While the use of even further open MI (such as MLab,

[1] Quoting CAIDA's Measurement Infrastructure Comparison Webpage [15] "*This list was compiled in 2004 and is no longer being maintained. This page is made available for historical purposes*"

[2] As a side effect of this work, we contributed to correct the geolocalization of some of them contacting the infrastructure maintainers. Annotations about inconsistent vantage points locations are available at [12]

Table 2.2: State of the Art in Anycast Measurement Research

	[104]	[72]	*This work*	[47]	[118]	[92]	[42]	[43]	[100]	[98]
Platform(#VPs)	Renesys monitors	PL (238), Netalyzr (62k), rDNS (300k)	*PL (300), RIPE (6k)*	End-hosts $O(100)$	PL (300)	DNSMC (77)	PL (129)	rDNS (20K)	C,F,K root	Renesys monitors
Technique	BGP vs. traceroute	DNS CHAOS +traceroute	*Latency probes*	DNS CHAOS	DNS CHAOS	DNS CHAOS +BGP	DNS CHAOS	DNS queries	pcap	BGP
Targets	IPv4 prefixes	F root, TLDs, AS112	*F,I,K,L root, EdgeCast CloufFlare CDNs*	C,F-K,M root	B,F,K root, TLDs	C,F-K,M root	C,F-K,M root, AS112	F,J root, AS11		1 CacheFly prefix
Detect	✓	✓	✓							
Enumerate		✓	✓							
Geolocalize			✓							
Proximity					✓		✓	✓	✓	
Affinity				✓	✓		✓	✓	✓	✓
Availability					✓	✓		✓		✓
Loadbalance								✓		

Archipelago, etc.) would be of course interesting, we remark these platform to have a comparatively smaller footprint than PlanetLab and RIPE Atlas, which thus already constitute an interesting and relevant starting point.

3 State of the art

Knowledge of IP-anycast is instrumental not only for characterization, troubleshooting and infrastructure mapping [31,65] but also for security-related tasks such as censorship detection [113]. Moreover, IP-anycast deployments evolve over time and it is important to have this information up-to-date.

However, while latency-based unicast geolocation [70,80] is well studied, the same does not hold for anycast, where triangulation technique locating unicast instances at the intersection of several measurement do not apply. We now review the state of the art, organized along several sub-categories related to our work.

3.1 Unicast geolocation

Unicast geolocation is a well investigated research topic: numerous techniques based on latency measurements [69, 70, 80] and databases [115, 120] have been proposed for unicast geolocation. Yet, database techniques are not only unreliable with unicast [115], but also with anycast, since they advertise a single geolocation per IP. Similarly, latency-based techniques [69, 70, 80] 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 we will illustrate in Chapter 3.

3.2 Unicast infrastructure mapping

Unicast infrastructure mapping studies, the last in line being [51, 123], leverage EDNS-client-subnet (ECS) extension 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 over-the-top operator. However, ECS support is becoming widespread to enhance the user online experience but is not yet pervasive, and additionally ECS technique fails with anycast (notice indeed that [123] reports only a single location for Edgecast, which instead has a few tens of PoPs, geographically dispersed worldwide).

3.3 Anycast characterisation

Research on anycast has so far prevalently focused either on architectural modifications [42, 75, 76, 93] or on the characterization of existing anycast deployments, which is close to our work and that we compactly summarize in Table 2.2. Overall, a large fraction of these studies quantify the performance of anycast in current IP anycast deployments in terms of metrics such as proximity [42, 43, 59, 100, 118], affinity [42–44, 47, 98, 100, 118], availability [43, 92, 98, 118], and load-balancing [43]. Only [44, 100] base their investigations on passive measurement methodologies. This complementary view is important to offer a characterisation from the end-user point of view such as the anycast popularity and the characteristic of their traffic. In both cases, authors' collection point is located at the anycast servers, so that they obtain a complete view of all user requests served by specific single server. Interestingly, while the body of historical work targets DNS, more recent work [52, 75, 98] has tackled investigation of anycast CDN performance (e.g., client-server affinity and anycast prefix availability for the CacheFly CDN).

3.4 Anycast infrastructure mapping

Fewer techniques instead exists that allow to detect, enumerate or geolocate anycast replicas, and that are thus closest to this work. In terms of methodologies, as reported summarized in Table 2.2, anycast infrastructure mapping has so far employed the following measurement techniques: (i) issuing DNS queries of special class (CHAOS), type (TXT), and name (host-name.bind or id.server [130]), (ii) BGP feeds, and finally active (iii) traceroute or (iv) ICMP latency measurement. Specifically, [72] employs (i) and (iii), while [104] leverages (ii) and (iii) and our proposal [56] uses exclusively (iv).

3.5 Anycast detection

In particular, *detection of anycast prefixes* is tackled in [104] by leveraging latency measurement (from distributed traceroute agents) and BGP routing information (from looking glass and public routers). BGP information is helpful since when the transit tree of an IP prefix has multiple domestic ISPs that are located in disjoint geographic locations, this IP prefix is likely anycasted. Latency measurement complement this information inferring if an IP prefix is anycast by detecting speed-of-light violations, as in this work. Techniques used in [104] have the merit of being protocol-independent, yet they do not to enumerate (let alone to geolocate) replicas as we do in this work.

3.6 Anycast enumeration

Protocol specific information is instead leveraged in [72] to *enumerate DNS anycast replicas.* As pointed out in [72], unfortunately not even all anycasted DNS servers reply with their identifier to CHAOS-class queries, and in case they do, the replies do not always follow a common naming standard. Additionally, [72] show that in-path proxies may modify such replies and therefore propose two new techniques to distinguish among servers within an anycast group. These techniques consist in either augmenting DNS CHAOS TXT queries with traceroute or modifying existing anycast servers to reply to special DNS IN TXT queries. Despite being a valuable tool for the domain name system, the main drawback of DNS-based technique is their narrow field of applicability with respect to [56, 104].

3.7 Anycast geolocation

Finally, with the exception of our iGreedy proposal that we present in Chap. 3, *no other technique exists that is capable of lightweight and protocol-independent anycast replicas geolocation.* The technique is lightweight as it relies on a handful of latency measurement, and is reliable and robust in spite of noisy measurement: indeed, latency measurement are used for detection and enumeration purposes, whereas geolocalization depends on reliable side-channel information (e.g., such as city population).

3.8 Internet Censuses: Unicast vs Anycast

In the past, numerous studies focus their attention on scaling active scanning techniques to provide broad spatial surveys of the Internet infrastructure. Given the lack of high-rate scanning tools such as [23, 67, 97], at that time researchers have studied sample of the Internet-space [86] or have splitted the IPv4 space over multiple vantage points [8, 83, 84] or completed the scans in an extended period of time. Since 2006, authors [83] measures periodically the population of visible Internet edge hosts (at IP/32 level) from eight different vantage points in two different locations, providing an IPv4 hitlist (one likely alive IP/32 target per IP/24). In 2008, authors in [60] scanned the Internet to find DNS servers that provide incorrect resolutions. In 2010, the IRLscanner tool allowed to scan the IP/32 Internet in about 24 hours, and results from 21 Internet-wide scans using 6 different protocols have then been presented in [97]. In 2012, the Carna Botnet [8] has used 420k insecure embedded devices to build a distributed port scanner to scan all IPv4 addresses using nmap [102]. It is also worth pointing out that an independent analysis [95] 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, [95] 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.

In the recent years, the situation has drastically changed with the advent of new network scanner tools as ZMap [33] and Masscan [23], able to achieve scan rates in excess of 10 Gbps, which let a IP/32 scan complete in less than five minutes. This has led to a huge increase of sporadic and regular scans, including the malicious ones: as documented in [66], using a network telescope, authors detected over 10 million scans from about 1.5 million hosts during January 2014. These are mainly regular scans, with daily [4] or lower frequency [108, 117]. Despite only a tiny fraction of these scans target more than 1% of the monitored IPv4 address space, they generate the majority of the unsolicited traffic hitting the darknet.

In these unicast censuses, the set of targets can be split among VPs for scalability reasons and the location of VPs with respect to targets is irrelevant. 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 only few hundreds, 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 smaller than that of an unicast censuses.

Chapter 3

Anycast Geolocation

We start our investigation proposing a protocol-agnostic technique for IP anycast replicas discovery and geolocation. Yet, other techniques exist but they are limited to DNS deployments or L7 anycast. We also provide the community with open source software and datasets to replicate our experimental results, as well as facilitating the development of new techniques such as ours. In particular, our proposed method achieves thorough enumeration and city-level geolocalization of anycast instances from a set of known vantage points.

This chapter is organized as follows. We first specify our objectives in Sec. 1 and describe our dataset, including the ground-truth in Sec. 2. Our own solution of the anycast geolocation problem, namely iGreedy, is described in Sec. 3. Results based on the dataset and software that we release at [12] are shown at a glance in Sec. 4 and completed by a thorough sensitivity analysis in Sec. 5, after which conclusive remarks are gathered in Sec. 6. Finally, we report in App. B example of usage of the tool, which has since been extended to perform live measurement from RIPE Atlas.

1 Problem definition

Our open source test suite allows anyone not only to (i) perform anycast measurement with the iGreedy technique, that represents the current state of the art, but also to (ii) try out his or her anycast identification algorithm on ground truth data that we have established, and compare the performance of their algorithm against iGreedy. This section frames the problem at high level, overviewing at the input available to these algorithms, and presenting useful design guidelines to advance the state of the art. We defer details about the dataset (Sec. 2), our proposed algorithm (Sec. 3), its performance (Sec. 4) and sensitivity (Sec. 5) to later sections.

1.1 Latency-based algorithms

The algorithms that can be evaluated in our framework suite are ones that solve the problem using latency-based measurements. As depicted in Fig. 3.1, for any application of the algorithm there will be some number M of measurement agents launching latency measurement towards each of the target addresses (e.g., with ICMP ping or other protocols). We reported the details about the measurement infrastructure in Sec. 2 and we present in Sec. 2.1 the measurement campaigns, while their impact of iGreedy performance is the object of Sec. 5.3.

Measurement agents are located at a different vantage point somewhere in the world, at a known (and reliable) position expressed as latitude and longitude (lat, lon). As early introduced, the anycast identification problem consists of three subproblems: (i) given a target IP address, t, *detect* if it is anycasted, (ii) *enumerate* the replicas that are offering the anycasted service, and (iii) *geolocalize* those replicas. We now separately consider each subproblem.

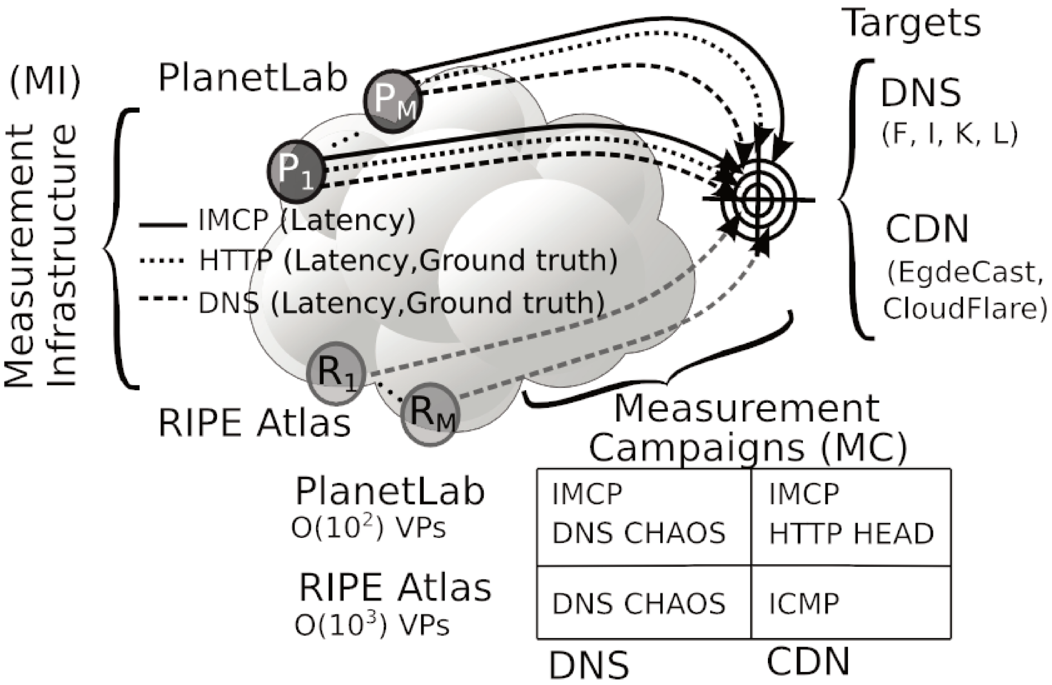


Figure 3.1: Synoptic of anycast measurement scenario

1.2 The anycast detection subproblem

We assume that any latency-based detection algorithm will generate a number $2 \leqslant N \leqslant M$ of disks for a given target IP address t. Each disk is a circle that is centered on a vantage point in which, according to speed of light calculations, t must lie. If there is any pair of disks that do not overlap, this is proof that t is an anycast address, as illustrated in Fig. 3.2 and introduced more formally in Sec. 3.1. On the other hand, if there is a unique area in the world on which all of the disks overlap then, while t *might* be anycast, we cannot prove this with the evidence at hand and so we assume t to be unicast.

For a given address that truly is anycast, loosely speaking the more disks that are generated, the more chances there are to correctly determine this fact. The choice of vantage points (their locations and the spacing between them) will also matter. It might be that for a given set of vantage points, even conducting measurements from all of the vantage points will not be sufficient to determine that some anycast addresses are such (e.g., when vantage points are few and close, or when the latency noise is large so that all disks overlap). But if there are non-overlapping disks, it suffices to correctly choose two vantage points in order to make the detection. A similar technique is employed by [104] to detect anycast replicas.

1.3 The anycast replica enumeration subproblem

Intuitively, if the observation of a pair of non-overlapping disks allows to detect an address as anycast, the observation of several disks that do not overlap among them allows to further enumerate distinct replicas. Given N disks, there are multiple ways to choose a set of $K \leqslant N$ non-overlapping disks such that the addition of any of the disks outside of this set would result in an overlap. Our enumeration technique is discussed in Sec. 3.2

Ultimately, the enumeration problem is better framed in terms of an *optimization problem*, in which an optimal solution consists in identifying the largest number (all if possible) of replicas. The enumeration subproblem may use the same disks as used for the detection problem (where only a pair suffices to trigger detection), or additional disks (aiming for a full enumeration). At a minimum, a number of disks equal to the number of anycast replicas is required in order to enumerate all of them. In practice, it might not be possible to enumerate all replicas depending on the set of vantage points available in the measurement infrastructure. Sensitivity to the measurement infrastructure is assessed in Sec. 5.3.

1.4 The anycast replica geolocalization subproblem

For each of the K non-overlapping disks generated in the previous step, a replica must lie somewhere within that disk. The geolocalization problem can thus be thought as a

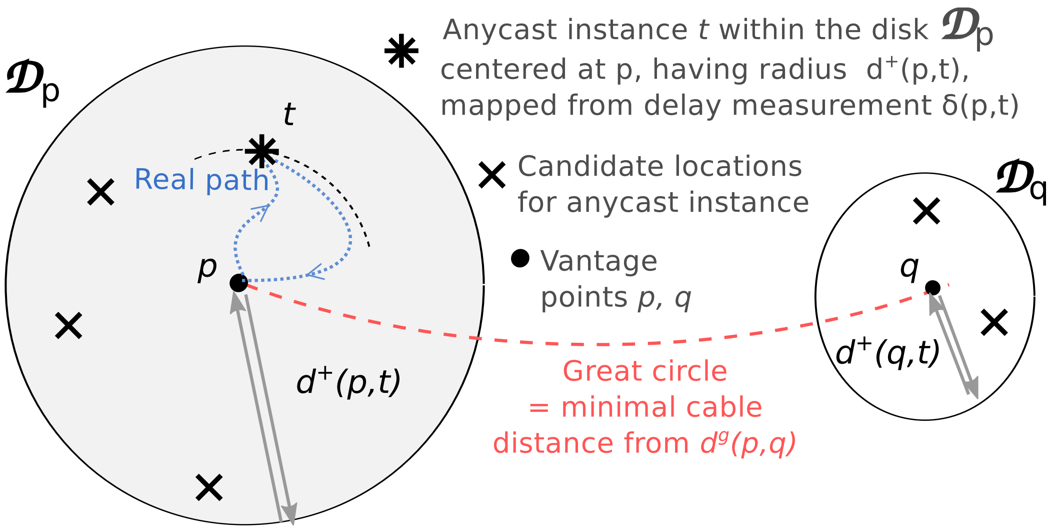

$\mathcal{D}_p$
t
Real path
p
$d^+(p,t)$
Anycast instance t within the disk $\mathcal{D}_p$ centered at p, having radius $d^+(p,t)$, mapped from delay measurement δ(p,t)
Candidate locations for anycast instance
Vantage points p, q
Great circle = minimal cable distance from $d^g(p,q)$
$\mathcal{D}_q$
q
$d^+(q,t)$

Figure 3.2: Synoptic of anycast instance detection via latency measurements

classification problem, in which we select, from a set of discrete locations within each disk, the most likely position of the anycast replica in that disk. Of course, in this stage not only latency measurement, but also any pertinent information can be leveraged by algorithms, such as, for instance, known landmass areas (replicas are unlikely to be out at sea) and locations of major metropolitan areas (replicas might be situated close to these, in order to promote low latencies to large numbers of users).

In Fig. 3.2 such discrete locations are indicated with crosses within the disk: we discuss how we build a reliable ground truth for the position of such crosses in Sec. 2.4. Our geolocation technique is described in Sec. 3.3 and a sensitivity of its performance is presented in Sec. 5.1.

1.5 Beyond the state of the art

Desirable properties of algorithms outlined above can be summarized as (i) reliable detection, (ii) complete enumeration, (iii) accurate geolocation, (iv) protocol independence and (v) low-overhead. Properties (i)-(iii) are specific to each algorithm, and this paper illustrates especially iGreedy enumeration and geolocation performance. The remaining properties are instead intrinsic to our problem formulation, since algorithms are given just (iv) RTT latency measurement (v) in the order of 100-1000 vantage points.

Notice especially that the (iv) protocol independence requirement suggests the use of ICMP to obtain latency samples. Indeed, given that no a priori information on the service running on an anycast host can be assumed, soliciting response on specific transport layer ports (e.g., UDP 53 for DNS or TCP 80 for CDN) would likely only obtain service-specific response (i.e., conditioned to the availability of that anycast service on the target under test). Conversely, ICMP based latency measurement are not affected by this per-service bias. We further discuss quality of latency samples in Sec. 2.1 and the impact that latency noise has on iGreedy in Sec. 5.3.

Finally, in terms of (v) overhead, we remark that the amount of probe traffic in our datasets is much lower to what considered in recent studies employing from 20k vantage points [43], to soliciting responses from about to 300k recursive DNS resolvers plus 60k Netalyzr datapoints [72]. In our framework, algorithms employs as few as 1/100 of the Netalyzr (or 1/1000 of the recursive DNS) data points: while challenging, our results show that fairly complete enumeration and correct geolocation are achievable even with few latency samples.

2 Datasets

As summarized in Fig.3.1, we run a number of *measurement campaigns* (MC) to provide latency measurement from a relatively low number (hundreds to thousands) of agents situated at different vantage points around the world. Location and number of agents depend on the *measurement infrastructure* (MI) used in the campaign. Since our MCs target known DNS and CDN anycast services, we can build reliable *ground truth* (GT) using protocol-specific information. In particular, building a ground truth for the anycast CDN service is, to the best of our knowledge, a novel contribution on its own – especially, since GT is highly more valuable with respect to *publicly available information* (PAI).

This section discusses the MC, PAI and GT datasets that we provide in our test suite, and that we used to obtain iGreedy results discussed in the reminder of the paper.

2.1 Measurement campaigns (MC)

As illustrated in Fig. 3.1 from the RIPE Atlas and PlanetLab MIs, we perform several *measurement campaigns* (MCs) destined to multiple target IP addresses, representative of different services. Specifically we target DNS root servers F, I, K and L and additionally two representative addresses of the EdgeCast and CloudFlare CDN.

For each vantage point p and target t, what can be easily measured is the round trip time delay $RTT_i(p,t)$ that the i-th packet sample took to travel from p to the closest instance of t and back to p. As algorithms require the one-way propagation delay, we estimate it as $\delta_i(p,t) = RTT_i(p,t)/2$: halving the round trip time, we make the worst case assumption of maximal distance from the vantage point (since forward and backward paths are not necessarily symmetric).

It is also well known that measured latency samples can vary from packet to packet, due to different paths [39], queuing delay [94], protocols [114] or even flow-id for the same protocol [114]. To partly compensate for these potential sources of bias we (i) use a minimum operation $\delta(p,t) = min_i RTT_i(p,t)/2$ over multiple P samples to removing noise due to variable RTT components (e.g., queuing delay) and (ii) use RTT samples gathered from different protocols (e.g., ICMP, DNS/UDP and HTTP/TCP). Concerning the latter point, we leverage different campaigns over different application layer protocols, such as DNS and HTTP, which are anyway needed to build the ground truth described in Sec. 2.4. In the case of DNS, $RTT_i(p,t)$ samples include the response time of DNS servers (yet another small but variable component that the minimum operation attempts at filtering). In the case of HTTP, $RTT_i(p,t)$ represent the TCP three-way handshake time.

We shall see in Sec. 5.3 that the number of P measurement, or the protocol they are gathered with have no noticeable impact on the performance of the algorithm we propose. To understand why this happens, it is worth recalling that (i) at the speed-of-light, packets travel about 100Km in 1ms and that (ii) recent measurement work [94] has shown that access links can queue several seconds worth of delay, well in excess of the Earth to Moon distance [53]. It follows that geolocation algorithms must cope with noise (due to protocol bias or individual samples variability) as otherwise even slight inaccuracies of the latency estimation can translate into fairly large errors for the geolocation problem. iGreedy thus prefer to leverage side-channel information (such as city population) to factor out latency measurement noise.

2.2 Publicly available information (PAI)

Publicly available information (PAI) about our target IP addresses is generally available through some Website. While PAI is of course valuable, it however adds additional challenges and ambiguities. In some cases, PAI *comprises* a larger set of replicas with respect to those actually visible from the VPs, which happens for instance in countries with low MI vantage point densities (e.g., China and African continent). In this case, discrepancies between an algorithm results and the PAI are tied to the measurement infrastructure, as opposite to the algorithm. Considering the PAI as reliable would in this case mistakingly increase the amount of False Negative classifications for the algorithm.

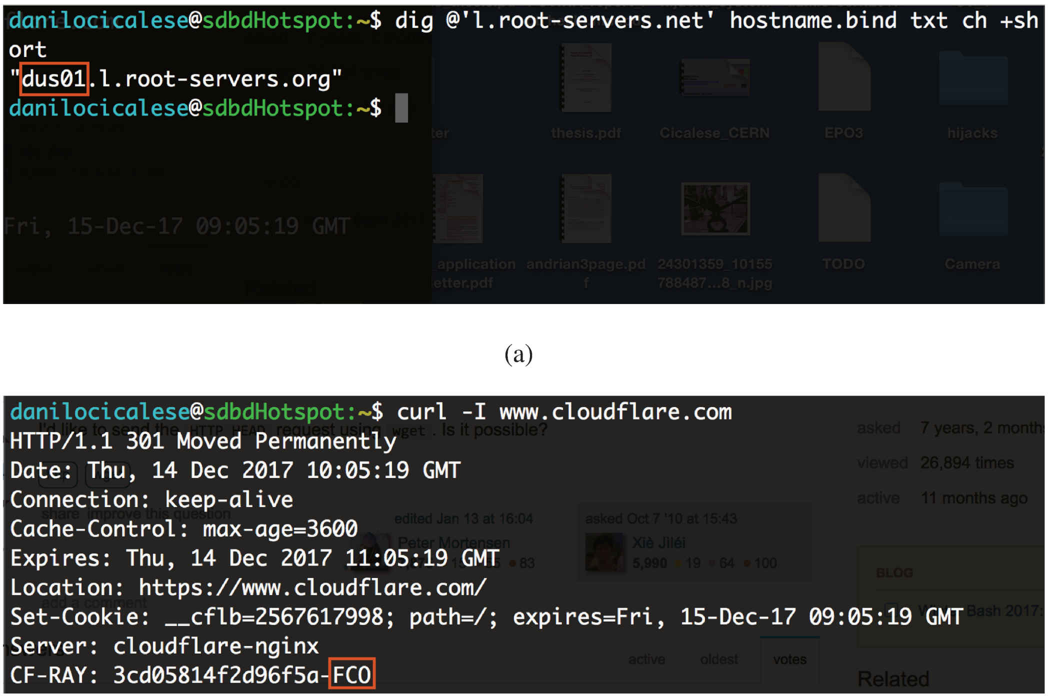
In other cases, the opposite is true: i.e., PAI comprises a *smaller* set with respect to the set of replicas actually seen from the VP, which happens whenever the Webpage content is outdated. One example is worth making to anecdotally assess the amount of discrepancy between PAI and measurement in the case of DNS root servers. DNS operators maintain an official website [38] with maps annotated with the number and geographic distribution of deployed sites around the world. According to [72], in 2013 PAI of root server E was advertising a single (unicast) location, despite their DNS state of the art method was able to enumerate 9 distinct locations. In 2014, at the time of our [56] experiments 12 anycast locations were advertised, but our PlanetLab and RIPE measurements were able to collectively discover over 40 replicas. Considering the PAI as reliable would in this case mistakingly increase the amount of False Positive classifications for the algorithm. A telling example of the manual validation concerns root server L, advertising a replica at SGW, which corresponds to an airport in Canada. However, this (spurious) instance was incoherent with the measurement from over 100 vantage points, that were (correctly) locating the vantage point in Singapore. PAI information do not report any airport in Canada but does in Singapore, confirming the hypothesis that protocol specific information is configured by humans and still possibly subject by errors, although unfrequented they can be.

These conflicting situations cannot of course be determined by solely relying on PAI and rather call for more accurate alternative methods.

2.3 Ground truth (GT)

For each target IP address in our dataset, we provide geolocation *ground truth* (GT), which is non ambiguous and solves the aforementioned issues with PAI. GT is assembled by (i) performing additional experiments that exploit protocol specific information and (ii) manually validating this new information against PAI and latency measurement. In the case of IP addresses of root DNS servers, we use DNS CHAOS requests as in [72], whereas we use HTTP requests in case of CDN IP addresses to reliably extract geolocation information as described in the following and represented in Fig. 3.3.

This is somewhat similar to the case of traffic classification, where protocol specific information (e.g., Deep Packet Inspection) is used to build a ground truth against which classification algorithms that *do not* rely on such protocol specific information can



```
danilocicalese@sdbdHotspot:~$ dig @'l.root-servers.net' hostname.bind txt ch +sh
ort
"dus01.l.root-servers.org"
danilocicalese@sdbdHotspot:~$ 
```

(a)

```
danilocicalese@sdbdHotspot:~$ curl -I www.cloudflare.com
HTTP/1.1 301 Moved Permanently
Date: Thu, 14 Dec 2017 10:05:19 GMT
Connection: keep-alive
Cache-Control: max-age=3600
Expires: Thu, 14 Dec 2017 11:05:19 GMT
Location: https://www.cloudflare.com/
Set-Cookie: __cflb=2567617998; path=/; expires=Fri, 15-Dec-17 09:05:19 GMT
Server: cloudflare-nginx
CF-RAY: 3cd05814f2d96f5a-FCO
```

(b)

Figure 3.3: Example of DNS and CDN ground truth

be tested [128]. However, it is useful to stress that collection of protocol specific information is used only in the validation phase, but is not necessary for the correct execution of the iGreedy algorithm.

CDN Ground truth. To collect CDN ground truth, we issue HTTP HEAD requests towards CloudFlare and EdgeCast to solicit a reply from the destination servers from PlanetLab. Unfortunately, it is not possible to issue HTTP HEAD requests from RIPE since the RIPE API does not provide HTTP support due to legal reasons (e.g., RIPE Atlas could be otherwise used as a proxy for accessing content restricted in some countries). It follows that from RIPE we are only able to issue ICMP measurements, whereas from PlanetLab we perform both ICMP and HTTP queries.

After manual inspection of the HTTP headers, we find that the HTTP reply headers contains meta-information about the servers location. Specifically, we observe that CloudFlare uses a custom `CF-RAY` header that uniquely identifies the server answering the HTTP request, whereas EdgeCast encodes such information in the standard `Server` header. Pairing such measurement data with PAI allows to reliably determine GT information.

CloudFlare encodes the server name directly as IATA airport codes, whose status is published at [18]. At the time we run our measurement campaign, EdgeCast used instead a mix of IATA codes and pseudorandom string for server names, publishing the servers list along with their geographical locations and the strings used to identify them at [11]. Currently, the page URL [28] as well as the information format has changed: while this is not surprising, it also testify that the collection effort of GT (and PAI) synchronously with MCs is far from being a trivial tasks. This further stresses the values of dataset sharing, which let the community capitalize on the effort of individual groups.

DNS Ground truth. To build a reliable DNS ground truth, we issue distributed IPv4 DNS queries of class CHAOS, type TXT, and name hostname.bind [130] to DNS root servers F, I, K, and L that are operated by ISC, Netnod, RIPE NCC, and ICANN respectively. We use both RIPE and PlanetLab to collect DNS replies to our queries: in the case of PlanetLab, we issue new ICMP and DNS measurement, whereas we rely in the case of RIPE of DNS measurement performed by the full set of vantage points that are already published by RIPE.

As in the previous case, pairing protocol-specific replies with PAI allows to reliably determine GT information in the DNS case. Indeed, despite CHAOS replies do not follow a standard format, GT is relatively easy to manually validate since some operators name servers in their infrastructure after IATA airport codes (e.g., AMS, PRG in root severs F and L respectively) or IXPs short names (e.g., AMS-IX, BIX, MIX in root

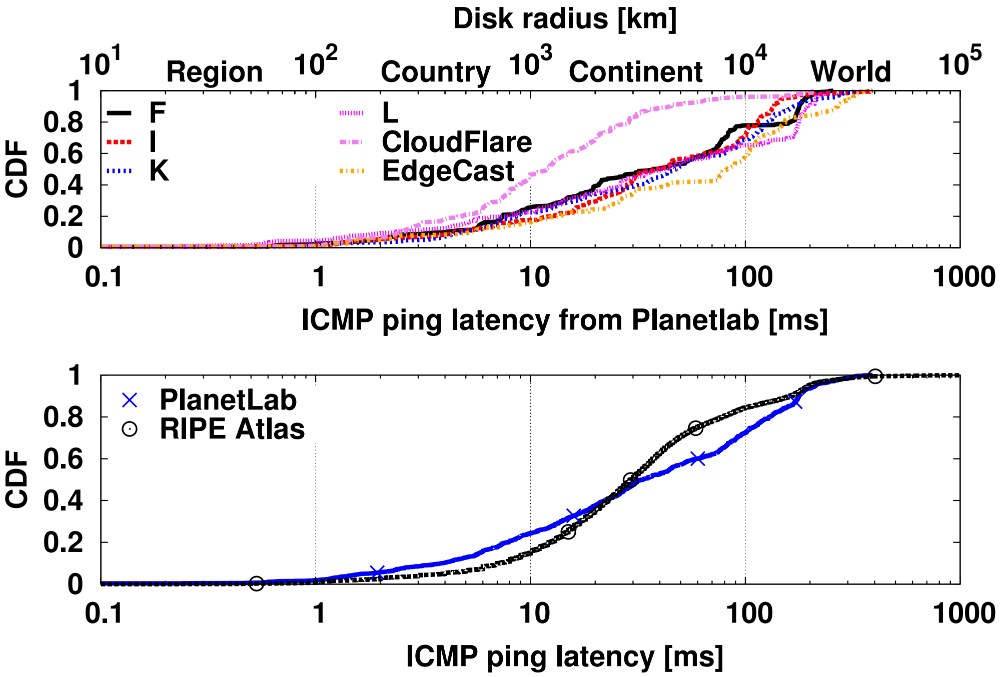


Figure 3.4: Dataset statistics: Per-target and Per-infrastructure of Cumulative distribution function (CDF) of minimum latency over all protocols

server K). In other few cases (e.g., root server I), operators use arbitrary codes, but make publicly available a list that maps site codes to locations. In sporadic cases, multiple CHAOS names are located in the same city: as we are interested in locating geographically distinct replicas, as opposite to enumerating the number of physical or virtual servers operating on a site, we coalesce all replicas located in the same site.

2.4 Dataset at a glance

We depict some relevant properties of our dataset in Fig.3.4 and Fig. 3.5. Notably, we report the Cumulative Distribution Function (CDF) of the latency samples coalescing ICMP, DNS and HTTP protocols altogether (left) for different targets (top) and measurement infrastructure (bottom). Since each latency sample translate into a disk in the

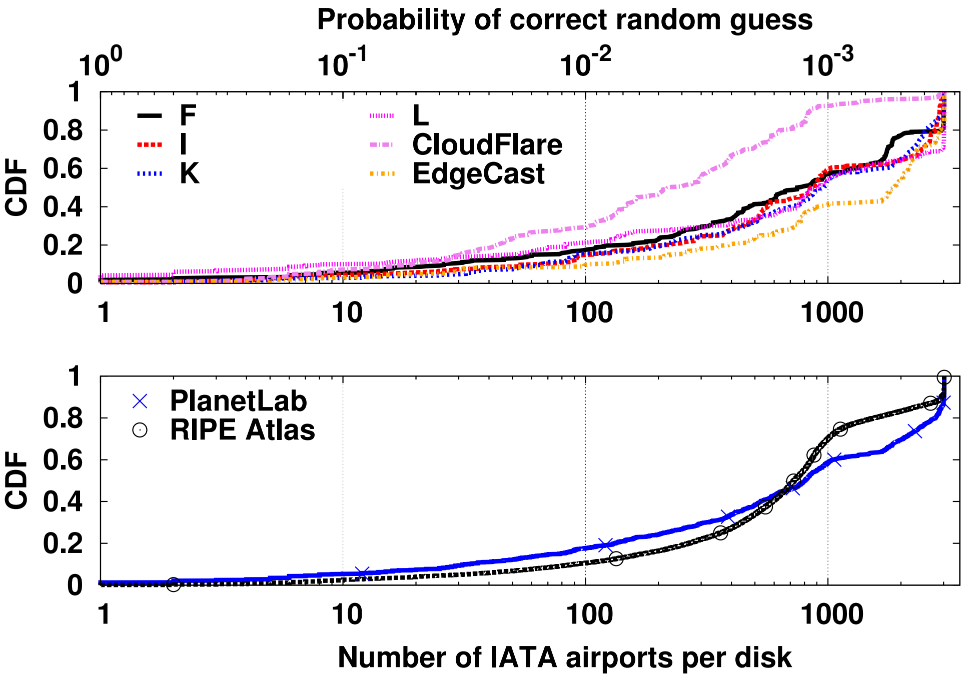


Figure 3.5: Dataset statistics: Per-target and Per-infrastructure of Complementary CDF (CCDF) of the number of IATA airports in each disk

Earth surface, we report the disk radius in the top-x axis for reference: it can be seen that the large majority of latency samples, being larger than 10 ms, map to disks that are big enough to cover a whole country.

As we have seen in the previous section, ground truth for both DNS and CDN is expressed by the very same DNS and CDN operators in terms of IATA airport codes. We report the CDF of the number of IATA airports per disk in the plot of Fig.3.5. Intuitively, the number of airports inside such disks can give a rough estimation of the difficulty of finding the correct replica location. for reference, top x-axis reports the probability of a random guess, which is inversely proportional to the number of airports in the disk. From the CDF in the bottom plot of Fig.3.5, one can notice that only about 10% of all disks contain less than 100 airports, or otherwise stated naïve guess has lower than 1/100 chances of success in roughly 90% of the cases: it follows that *anycast geolocation algorithms should thus be designed to be inherently robust to noise in latency measurement.*

A final point is worth clarifying. Our dataset also includes a set of over 3,000 airports, out of the about 7,000 available airports with a IATA code. The dataset associate airports with several properties, such as its latitude and longitude, the name of the city it serves, the population of the city, the airport "type" (taking values such as Small/Medium/Large Airport, Heliports, Closed Airports, Seaplane Bases, Balloonports, etc.) available in open databases. Of the about 7,000 airports with a IATA code, only about 500 airports worldwide fall in the large category – hence, the wide majority of airports in our dataset are of the "medium" type.

Notice that the type attribute correlates to the size of the airport, not to the size of the city population. Additionally, multiple airports of several type can be associated to the same city. Taking Paris as an example, the datasets[1] lists several airports (namely CDG, ORY, PAR, BVA and LBG). While Paris is a city with a given population (of about 2 million inhabitants) these airports have a variable size. Moreover, while CDG is very often use in naming severs, both CDG and ORY are "large" airports, that are sometimes found as well. PAR is a virtual airport, i.e., a label for any Paris airport to simplify searching flights. There are also less known airports such as BVA (essentially, low cost flights) and LBG (famous for international airshow every two years) airports, categorized as "medium“. The mapping between city and IATA is therefore a many-to-one mapping: given our target city-level geographic accuracy in this paper, this is not problematic, but shall deserve further attention to provide finer-grain geolocation (i.e., sub 100km radius or building-level [79]).

[1] The airport dataset is available along with our provided software in `./igreedy-1.0/datasets/airports.csv`

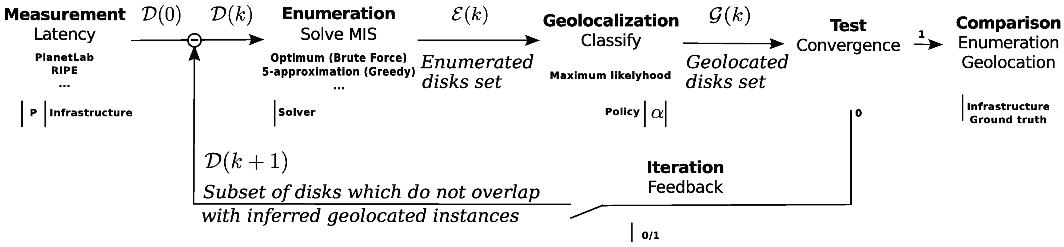


Figure 3.6: Synoptic iGreedy methodology

3 iGreedy Methodology and Design

Our test suite also comprises an open source implementation of the iGreedy technique we originally proposed in [56], of which we illustrate the inner working with the help of Fig. 3.7. The implementation can operate on historical data (i.e., the dataset early described, corredated with ground truth) or issue new measurement from RIPE Atlas (in which case the provided GT is no longer up-to-date).

In a nutshell, (a) we first perform latency measurement to a given IP and map them to a disks centered around the VP, that by design contains at least one anycast instance; (b) if two such disks do not intersect, we can infer that VPs are contacting two different replicas, as is the case for the green disks in Fig. 3.7(b); (c) we provide a conservative estimate of the minimum number of anycast replicas by solving a Maximum Independent Set (MIS) problem, using a greedy 5-approximation algorithm that operates on disks of increasing radius size as in Fig. 3.7(c); (d) in each disk, we geolocate the replica at city-level granularity with a maximum likelihood estimator biased toward city population; and finally, (e) we coalesce the disks to the classified cities, 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. We now describe the steps in more details.

3.1 Detection

At a logical level, prior to enumerating anycast instances, we must detect whether there are indeed anycast replicas behind a given unicast IP address. This can be done so by detecting speed-of-light violations in our dataset by comparing latency measurements δ to the expected propagation time due to speed-of-light considerations as in [104]. As early illustrated in Fig. 3.2, we consider pairs of latency measurements $\delta(p,t)$ and $\delta(q,t)$ for the same target. We map measurements to disks $\mathcal{D}_p$ whose radius equals $d^+(p,t) = \delta(p,t)/3$, where the constant 3 lumps altogether several factors (such as the speed-of-light in a fiber medium, the fact that fiber deployment is subject to physical constraints, etc.), of which a good overview is given in [121].

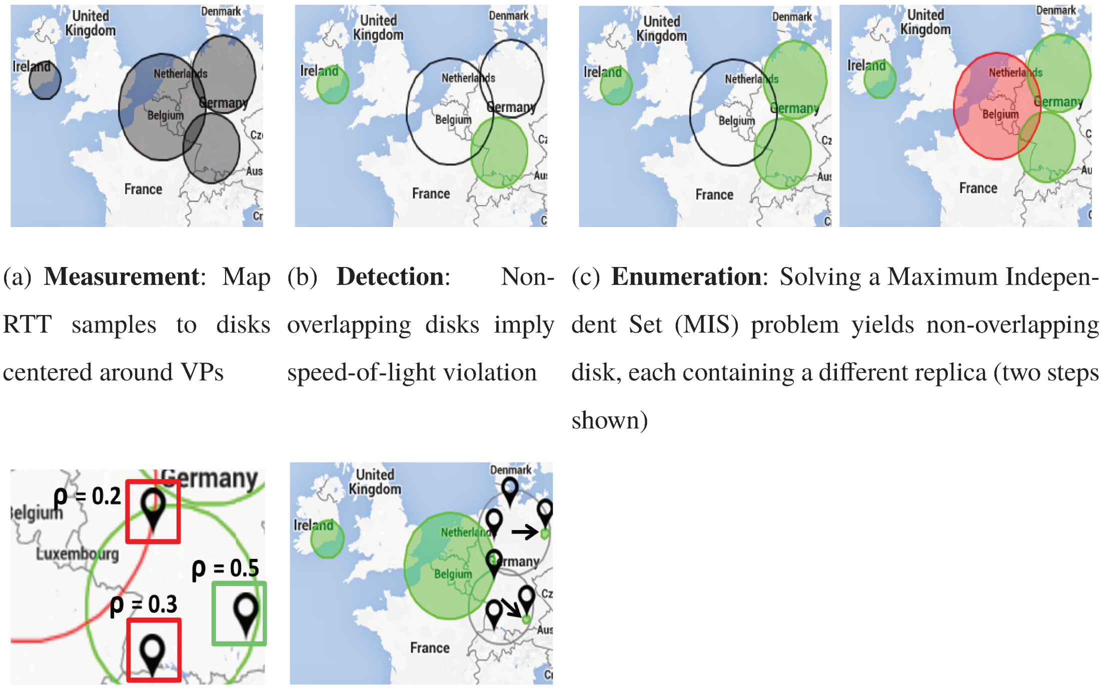
Specifically, given two vantage points p, q we compute their geodesic distance $d^g(p,q)$ according to Vincenty's formulæ. As packets cannot travel faster than light, if

$$d^g(p,q) > d^+(p,t) + d^+(q,t) \geq d(p,t) + d(q,t) \tag{3.1}$$

then disks $\mathcal{D}_p$ and $\mathcal{D}_q$ do not overlap, indicating that p and q are in contact with two separate anycast replicas.

Some remarks are in order. First, (3.1) compares distances with homogeneous dimensions, that are however gathered with different techniques. Note that considering the geodesic distance $d^g(p,q)$ between vantage points yields a *conservative lower bound* to the expected propagation time between p and q, as a packet will not travel along a geodesic path but will follow a path shaped by physical and economic constraints (i.e., the geography of fiber deployment, optoelectronic conversion, BGP routing, etc.). Instead, since latency is not only due to propagation delay, $d^+(p,t)$ conversely *aggressively upper bounds* the distance that a packet may have traveled during $\delta(p,t)$.

As the the inequality is violated only when the conservative lower bound exceeds the upper bound, it follows that (3.1) is conservative in detecting anycast instances and, assuming correct geolocation of VPs, by definition avoids raising false positive anycast instances (i.e., flagging as anycast a truly unicast target). Notice that in the iGreedy



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

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

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

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

(e) **Iteration**: Collapse disks around geolocalized replicas until convergence

Figure 3.7: Illustration of the iGreedy anycast detection, enumeration and geolocalization algorithm

implementation, the detection step illustrated in Fig. 3.7(b) is a side product of the enumeration, and is thus not explicitly accounted for in the workflow of Fig. 3.7: i.e., in case two or more anycast replicas are enumerated, this also implies true positive anycast detection. Of course, while false negatives are possible on a single inequality (i.e., flagging as unicast a truly anycast target), their odd decreases using multiple vantage point pairs.

3.2 Enumeration

While the detection criterion expressed by (3.1) is not particularly novel [104], we are the first to leverage a full set of distributed measurements in the study of anycast deployment and its geographical properties as illustrated in Fig. 3.7(c). Distributed measurement allow indeed to infer more sophisticated properties than mere binary detection, such as the count of anycast replicas, or their position.

It is possible that multiple anycast instances may be located within a given disk. Although the aim of anycast is to offer services from distinct locations, the locations may be distinct from an IP routing point of view but not distant geographically from each other. Therefore, our technique can only provide a lower bound of the number of anycast instances that correspond to our observations.

MIS formulation. To achieve our enumeration goal and simplify the geolocalization step, we model the problem as a *Maximum Independent Set (MIS)*. Our aim is to find a maximum number of vantage points (and corresponding disks) for which we are confident they contact distinct anycast instances (an instance being included in the disk). To do so, we select a maximum subset of disks $\mathcal{E} \subset \mathcal{D}$ such that:

$$\forall \mathcal{D}_p, \mathcal{D}_q \in \mathcal{E}, \qquad \mathcal{D}_p \cap \mathcal{D}_q = \emptyset \tag{3.2}$$

The enumeration problem is thus solved by the subset $\mathcal{E}$, whose cardinality $|\mathcal{E}|$ corresponds to the minimum number of instances that avoid latency violations, and which represents thus a plausible explanation to our observations. Notice that $|\mathcal{E}|$ is a lower bound on the number of anycast instances, since due to the conservative definition of (3.1) we might have removed disks that overlap due to noisy measurements. Additionally, each disk of $\mathcal{E}$ yields a coarse location of anycast replicas useful starting point for refinement in the geolocalization step.

Efficient MIS solution. Although the MIS problem is NP-hard, it can be solved in finite time for small number of vantage points with a brute force approach. This allows us to compare the solution of known greedy approximate solutions: while a simple greedy strategy has poor performance in general ($(M-1)-$approximation) the situation improves by simply sorting disks in increasing radius size (5-approximation), with complexity $O(Mlog(M) + M(M-1)/2)$ determined by the comparisons. The greedy pseudocode is trivial, but reported here for completeness.

Algorithm 2 Greedy 5-approximation to MIS for anycast instances enumeration

```
Require: A set of disk $\mathcal{D}$
Ensure: A set of disk $\mathcal{E}$ such as $\forall p, q \in \mathcal{E}, \mathcal{D}_p \cap \mathcal{D}_q = \emptyset$
  Initialization: sort disks in $\mathcal{D}$ by increasing radius size
  Initialization: $\mathcal{E} \leftarrow \emptyset$
  for all disk $\mathcal{D}_d$ of $\mathcal{D}$ do
    for all disk $\mathcal{D}_e$ of $\mathcal{E}$ do
      if $\mathcal{D}_d \cap \mathcal{D}_e = \emptyset$ then
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{\mathcal{D}_e\}$
      end if
    end for
  end for
```

We point out that more refined solutions do exist [71,90], that achieve $(1-\frac{2}{k})-OPT$ performance at the price of a non marginal computational complexity $M^{O(k^4)}$, where k is a tunable parameter. Roughly, the main idea in [71,90] is to slice the original dataset into several layers, with disks into layer ℓ having diameter d satisfying $1/(k+1)^{\ell} \geq d > 1/(k+1)^{\ell+1}$. The more the layers, the closer the approximation to the optimum, the longer the computational time. Notice also that k must be greater than 2 to apply the slicing.

As we will show later, the greedy solution often performs well in practice and is often comparable to the brute force solution, which means that more refined solution are not worth for this problem. Since computational time of a greedy approximation for O(100) vantage points is in the order of O(100ms), whereas brute force solution is O(1000sec), a simple greedy MIS solver has an undoubt practical appeal.

Alternative formulation. A final remark is worth making. Had our original goal been limited to the *enumeration* of anycast instances a *Boolean satisfiability (SAT)* formulation would have been more appropriate. A geometric interpretation of our problem would be then as follows: if we represent an anycast instance using a cross, SAT consists in placing a minimum number of crosses such that all the disks $\mathcal{D}_p$ contains at least one cross. A benefit of this formulation is that SAT upper bounds the number of instances returned by MIS, since MIS removes overlapping disks from the set: hence, a SAT formulation would increase the recall of anycast instances. At the same time, the SAT formulation would make the geolocalization harder and is thus not adapted to our goals: indeed, as several realizations having exactly the same number of anycast instances could satisfy this problem, SAT does not easily allow to determine in which circles intersection the anycast instances are located. Releasing datasets as open source should facilitate implementation of alternative techniques for replicas enumeration such as the one just illustrated.

3.3 Geolocalization

Our aim being to provide geographic locations at city granularity, we need to refine the preliminary location that is output by the enumeration algorithm. We opt for city granularity for two reasons. First, recall that a 1 ms noise in latency measurement corresponds to an increase in the disk size by 100 km. It follows that great trust should be put in latency measurements to achieve finer-grained geolocalization. Second, notice that the ground truth provided by DNS and CDN measurement is already provided as IATA airport codes. City-level granularity naturally allow to assess the correctness of our geolocalization.

Classification formulation. As opposed to classical approaches that operate in the geodesic (or Euclidean) space by constructing density maps of likely positions (see references in [70]), or assessing target location to be the center of mass of multiple vantage points [51], we transform the geolocalization task into a classification problem as in [69]. Specifically, since our output is a geolocalization at city level granularity, we shift the focus from identifying a geographical locus $(lat, lon) \in \mathcal{D}_p \subset \mathbb{R}^2$ to identifying which among the cities $c \in C \subset \mathbb{N}$ contained in the disk $(lat_c, lon_c) \in \mathcal{D}_p$ is most likely hosting the anycast instance. This focus shift greatly simplifies the problem in two ways: first, it significantly reduces the space cardinality ($\mathbb{R}^2$ to $\mathbb{N}$); second, it allows us to further leverage additional information with respect to delay or distance measurements, namely the city population.

Geolocalization step outputs IATA airport codes as shorthand for cities. For each of the non-overlapping disks of the enumeration phase, some of the over 7,000 airports codes may be contained in the disk. Aside from the trivial case where a single airport is contained in the disk, in the general case multiple airports $\{A_i\} \in \mathcal{D}_p$, represented as a crosses in Fig. 3.2, are contained in any given disk. The output of the geolocalization phase can thus be expressed with disk-airport pairs $\mathcal{G} = \{(\mathcal{D}_i, A_i)\}$ according to the notation of Fig. 4.1.

Classification criterion. To guide our selection of the most likely location of a site, we employ two metrics, namely: (i) the distance between the airport and the disk border $d(p, t) - d(p, A_i)$ and (ii) the population c_i of the main city c_i that the airport A_i serves. Our intuition in using (i) the distance between an airport and the border of a disk is that the larger the distance, the nearer is the airport from the vantage point. Here, we use geographic proximity from the vantage point as a proxy for topological proximity in the routing space. Our reasoning for (ii) extends previous work [69], which argues that IPs are likely to be located where humans are located: in other words, due to the distribution of population density, large cities represent the likely geolocalization of single-host unicast IP addresses. We further argue that, since anycast replicas are specifically engineered to reduce service latency, they ultimately have to be located close to where users live: hence the bias toward large cities is again likely to hold for server side anycast IPs as well. Hence, we include only airports that are categorized as "medium" and "large" in the database, but exclude airports categorized as "small", because we are only interested in locations that are densely populated. Our IATA dataset contains a total of over 3000 airports.

For a given disk $\mathcal{D}_p$ we compute the likelihood of each airport $\{A_i\} \in \mathcal{D}_p$ for all airports in the disk, illustrated in Fig. 3.7(d) and defined as:

$$p_i = \alpha \frac{c_i}{\sum_j c_j} + (1-\alpha) \frac{d(p,t) - d(p,A_i)}{\sum_j d(p,t) - d(p,A_j)} \tag{3.3}$$

where $\sum_i p_i{=}1$ follows from the normalization over all airports $\{A_i\} \in \mathcal{D}_p$ of the c_i contribution (population of the main city served by airport A_i) and of the $d(p,t) - d(p, A_i)$ contribution (the distance of the airport i from the disk border). The parameter $\alpha \in [0,1]$ tunes the relative importance of population vs distance in the decision, in between the distance-only ($\alpha = 0$) vs city-only ($\alpha = 1$) extremes. Likelihood for three cities is exemplified in Fig.3.7(d). Based on the p_i values, we devise two maximum likelihood policies that return either (i) a single $A_i = \text{argmax}_i p_i$ or (ii) all locations (A_i, p_i) annotated with their respective likelihoods. These policies involve a trade off, as returning all locations increases the average error (since in case $\text{argmax}_i p_i$ is correct, it pays the price of incorrect answers for $1 - p_i$), whereas returning a single location possibly involves a larger error. As we shall see, the city population has sufficient discriminative power alone, so that the simplest criterion of picking the largest city is also the best:

$$A_i = \text{argmax}_i c_i / \sum_j c_j \tag{3.4}$$

3.4 Iteration

Recall that the enumeration step lower bounds the number of instances, due to the possibility of overlapping disks. Now, consider that the geolocalization decision in effect transforms a disk $\mathcal{D}_p$, irrespective of its original radius, into a disk $\mathcal{D}'_p$ centered around the selected airport with arbitrarily small radius (in this work, we conservatively shrink disks to a 100Km radius). Hence, we argue that, provided the geolocalization technique is accurate, it would be beneficial to transform the original set of disks $\mathcal{D}$ by (i) remapping $\mathcal{D}_p$ to $\mathcal{D}'_p$ and (ii) excluding from $\mathcal{D}$ those disks that contain any of the geolocalized cities $\mathcal{D}'_p$. This case is illustrated in Fig. 3.7: the red-shaded disk overlapping in the previous enumeration step Fig. 3.7(c), no longer overlaps after geolocalization of the two circles in Fig. 3.7(d), so that it can be considered in the next iteration Fig. 3.7(e). Denoting with $\mathcal{A}(k)$ the subset of airports geolocalized up to step k, and with $\mathcal{G}(k)$ the geolocalization at step k (considering a single airport selected per disk for the sake of simplicity), defined as $\mathcal{G}(k) = \{(\mathcal{D}_i, A_i) \in \mathcal{E}(k) \times \mathcal{A}(k))\}$, we have that the dataset $\mathcal{D}(k+1)$ as input to the numeration problem at step $k+1$ can be written as $\mathcal{D}(k+1) = \mathcal{D}(k) \backslash \{\mathcal{D}_i : \exists (\mathcal{D}_i, A_i) \in \cup_{i=1}^{k} \mathcal{G}(i)\}$. Iterations continue until no further disk can be added that does not overlap. At each iteration, the set of geolocalized cities grows, so that the set of disks that no longer overlap diminishes, which keep the running time reasonably bounded. Note that iterative operations can be employed irrespectively of the underlying solver (i.e., brute force, greedy, etc.). Notice also that iteration "couples" the analysis of the geolocalization and enumeration performance, as the input to the latter is modified by the former. We analyze coverage benefits and additional complexity of iterative workflow in Sec. 5.2.

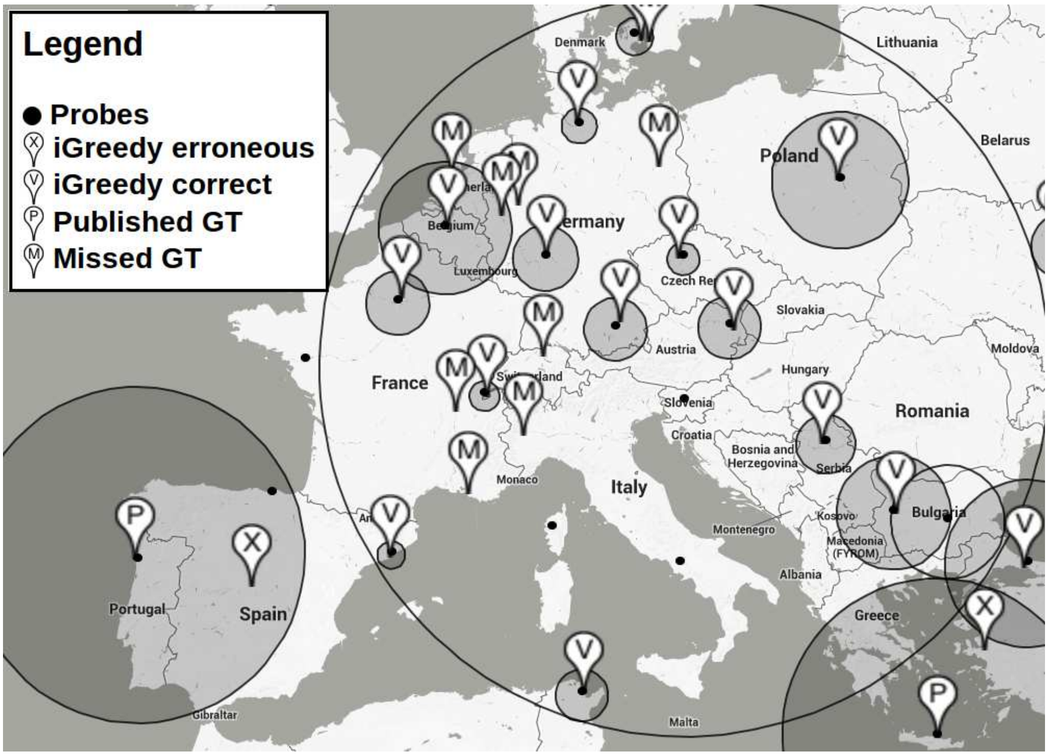


Figure 3.8: iGreedy result: Illustrative example of root server L

4 Results at a glance

We now run the open source iGreedy implementation on the dataset[2] provided to the community, reporting at a glance illustrative examples and key performance indicators. Results in this section are gathered with the "largest city" criterion of (3.4) – we defer justification of this choice to Sec. 5, where we perform a thorough assessment of iGreedy robustness.

4.1 Example of results

Geographical maps (using a GoogleMaps interface) are among the outputs directly available in the open source iGreedy implementation. For the sake of illustration,

[2] It is worth recalling that datasets are released in the same tarball of the iGreedy code.

Fig. 3.8 depicts an Euro-centric view of geolocalization of root server L replicas: the map reports vantage points (black dots) and the results of iGreedy as shaded disks that contain either correct ✓ (True Positive, TP) or erroneous × (False Positive, FP) geolocalization markers (and in the last case the location of the ground truth P as well) according to the earlier discussed ground truth. The map additionally reports missed M instances (False Negative, FN), whose position is known through public available information. Such instances were missed either because (i) they are not observed in our measurement, which is the cause of the large majority of this misses or because (ii) they are observed in disks that overlap (represented as circles with no shading). Additionally, notice an example of vantage points that our iterative workflow allows to include: i.e., disks of the Bruxelles and Paris vantage points intersect. Finally, observe that population bias yields to misclassification for the point located in Porto, Portugal: this vantage point exhibits a relatively large latency (6 ms) to hit a target also located in Porto, so that the disk is large enough to include Madrid (population of 3.3M) which is an order of magnitude more populated than Porto (population of 250K). The distance between Madrid and Porto is 420Km, which is just 10% above the median geolocation error of iGreedy.

4.2 Comparison with the state of the art

We separate analysis of iGreedy enumeration and geolocation as follows. Enumeration aims at *completeness*, i.e., assessing the number of disks $|\mathcal{E}|$ contacting different replicas that iGreedy is able to recollect, independently whether the geolocation succeed. We normalize the number of replicas to the number in the ground truth obtained with protocol specific information and denote this ratio $|\mathcal{E}|$/GT as enumeration completeness. Geolocalization instead aims at *correctness*, so that it is important to assess the amount of geolocation that correctly matches the ground truth, which can be expressed as the True Positive Rate or Precision = TP/(TP+FP).

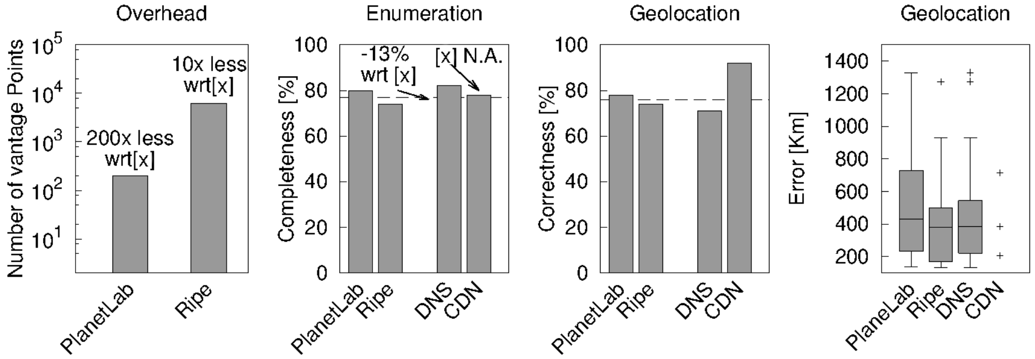


Figure 3.9: Compact summary of geolocation and enumeration performance of iGreedy across MI (PlanetLab, RIPE) and breakdown of PlanetLab targets across services (DNS, CDN). Note that only 3 replicas are at erroneous location in the PlanetLab CDN case, illustrated as points in the geolocalization error boxplot. Comparison with [72] indicated in the Fig. with x.

Notice that the enumeration results of [72] are *directly quantitatively* comparable, as [72] employs F and other root servers as a case study, albeit the measurement timeframe and infrastructure differ. Conversely, since iGreedy is the only technique able of anycast geolocation, we do not have any candidate technique to directly[3] compare with in this case, and thus make our dataset open to promote and facilitate future comparisons.

At a glance, Fig. 3.9 shows that, with respect to [72], iGreedy: (a) reduces the measurement overhead by several orders of magnitude, (b) has comparable yet lower enumeration recall. Additionally, while [72] technique is limited to DNS and does

[3] In a sense, [72] also provides geolocation in CHAOS replies, which is the very same information we use to build the ground truth: however, its use is limited to DNS and is conditioned to having location information encoded in server names, which we have seen to apply to only part of DNS root servers. Similarly, our previous work [56] compares geolocalization results of the *unicast* Client-Centric Geolocalization (CCG) [51] (which are however *only qualitatively* comparable, as they additionally target the Google infrastructure). As qualitative comparison may lead to misinterpretation, we prefer to avoid reporting it here (which is available to the interested reader in [56])

not allow geolocalization, iGreedy: (c) is protocol protocol agnostic and (d) is able to correctly guess anycast instance location about 3/4 of the time. Finally, we can see that results are (e) qualitative consistent across service and measurement infrastructure. As for (a), we indeed notice that [72] employs 62K vantage points (the Netalyzr dataset), i.e., about 200× larger than PlanetLab and 10× larger than RIPE Atlas. As for (b), since we observe enumeration performance that are worse than that of [72] but anyway comparable, this intrinsically means that the datasets used in [72] are highly redundant (e.g., including multiple trials from the same users; or affected by popularity of Netalyzr in a given geographical region). The very same observation also holds for the MI we use in this paper (e.g., RIPE has several hundreds monitors in Paris, but iGreedy uses at most one of them), so that we explore this issue further in Sec. 5.3. As for (c), iGreedy is protocol-agnostic because only latency measurements are needed, that can be gathered from service-independent protocols such as ICMP. As for (d), we report that geolocation is correct in 78% of the cases, and that the median error distribution of the remaining 22% of the cases is 384 Km.

Making a parallel to unicast geolocation, it is worth noticing that error magnitude in iGreedy is similar to that of unicast techniques: e.g., without (with) filtering large delay samples, [51] reports a 556Km (22Km) median error. An additional similarity with is worth stressing, quoting [51] “A disadvantage of our geolocation technique is that large datacenters are often hosted in remote locations, and our technique will pull them towards large population centers that they serve. In this way, the estimated location ends up giving a sort of logical serving center of the server, which is not always the geographic location.”: the very same hold here for iGreedy.

5 Sensitivity analysis

We now thoroughly validate iGreedy (e.g., classification in Sec. 5.1, solver in Sec. 5.2) and assess its robustness to measurement campaigns and infrastructures (Sec. 5.3).

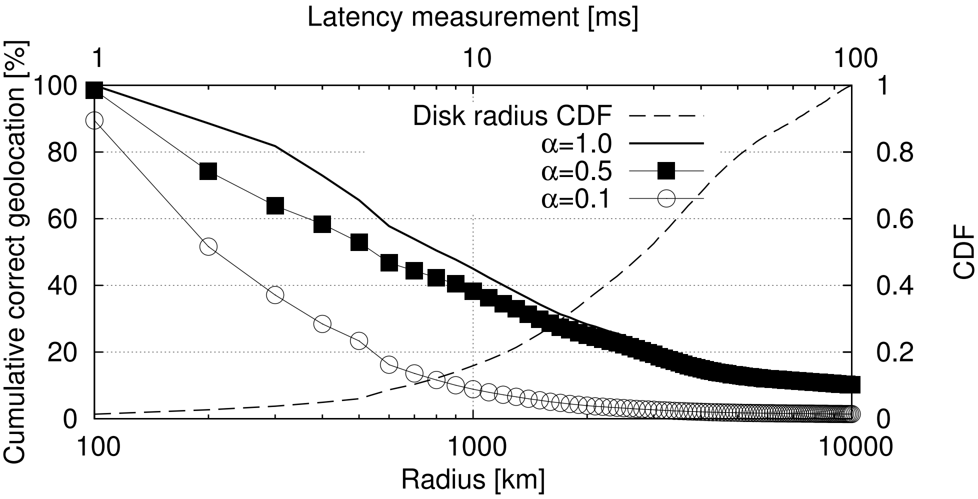


Figure 3.10: Tuning iGreedy classification: cumulative true positive geolocalization over all disks(RIPE, PlanetLab), with argmax selection for different weighting factor α.

5.1 Impact of classifier

Weighting factor α. It could be argued that, at least for small disks, latency information could bring useful information to improve classification accuracy. To assess this, we first consider individual disks $\mathcal{D}_p$, where we apply the geolocation criterion interpolating distance and population information via $\alpha \in \{0, 0.5, 1\}$ in (3.3). As we consider all disks irrespectively if they will be selected in the iGreedy enumeration phase, this information is valuable for all algorithms. Fig. 3.10 reports geolocation accuracy over all measurement platforms, targets and protocols: in particular, the plot shows the average correct geolocation ratio, cumulated over all disks up to a given size. As it can be seen from Fig. 3.10 the probability of correct geolocation is upper-bounded by $\alpha = 1$: this suggests that city population has discriminative power useful for any anycast geolocation algorithm in general, and ultimately corroborates the simple "largest city" criterion of (3.4).

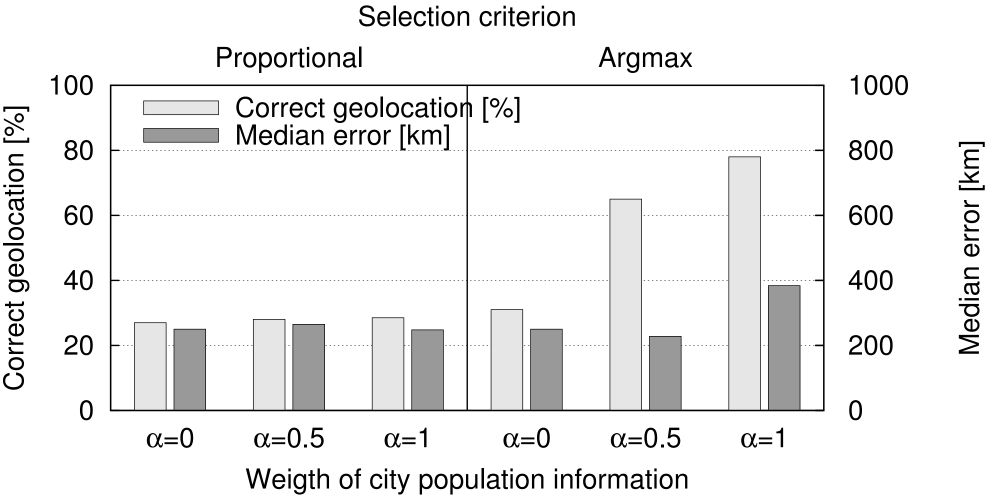


Figure 3.11: Tuning iGreedy classification: performance for different selection policies and weighting factor settings.

Given that geolocation is still erroneous in about 20% of the cases, this is an area for future improvement, using either complementary measurement (e.g., traceroute) or side-channel information (e.g., Internet exchange points maps [79]). Of course, we believe that improving geolocation accuracy of the classification step is an important point, as there is much room for improvement from the coarse-grained 100km-radius accuracy iGreedy aims at providing, towards the fine-grained building-level accuracy provided by [79]. Indeed, on the one hand one could argue that an anycast geolocation service is not useful given the accuracy does not reach acceptable levels (e.g., say 95%). On the other hand, one could also argue that even small improvements can accumulate over time and provide eventually an acceptable solution. Our hope is that this work, along with the software and the dataset we provide, can facilitate this second option to let the community collectively refine the existing service to an acceptable level.

Selection policy. We now turn our attention to iGreedy, where only a subset of all the above disks are selected by the algorithm: over this set of disks, we study combination of the selection policy (i.e., argmax vs proportional) and weighting factor ($\alpha \in \{0, 0.5, 1\}$). For any given disk $\mathcal{D}_p$, let us denote with A^{GT} the airport code given by the ground truth, and further denote with A_i the different airports that are located in $\mathcal{D}_p$. Considering the argmax policy, in case $A^{GT} = A_i$ (with i such that $\text{argmax}_i\, p_i$), the classification is accounted as correct and does not count in the error statistics. In case $A^{GT} \neq A_i$, then the classification is erroneous, and off by a distance $Err = d(A_i, A^{GT})$. In the proportional policy instead, the classification is accounted as correct only for p_i (i.e., proportionally to the percentage of time the correct instance would be selected). The geolocalization error for this instance is then computed over all airports inside the disk, and weighted according to the respective likelihood of each airport $Err = \sum_j d(A_j, A^{GT})p_j$.

In short, the tradeoff is here between optimistically returning a single location and possibly have large maximum error (argmax) vs conservatively bounding the maximum error but increase the average error (proportional). Results are reported in Fig. 3.11, from which it is easy to gather that, given that iGreedy preferably select smaller circles, and in reason of the good discriminative power of the city population, argmax is largely preferable with respect to the proportional policy. As early noticed performance are already similar for $\alpha > 0.5$, although $\alpha = 1$ confirms to be the best setting, leading furthermore to a very simple geolocalization criterion.

5.2 Impact of input data and solver

Latency measurement provides an input that is fed to a MIS solver for the enumeration phase. Given the large uncertainty in accurate geolocalization of replicas in large disks, it could be tempting to filter out latency measurement exceeding a configurable threshold. This is interesting not only to bound the maximum error, but also since a smaller dataset can reduce the computational time spent in the solution of the MIS.

Filtering input data. We thus start by filtering latency measurements fed to the MIS,

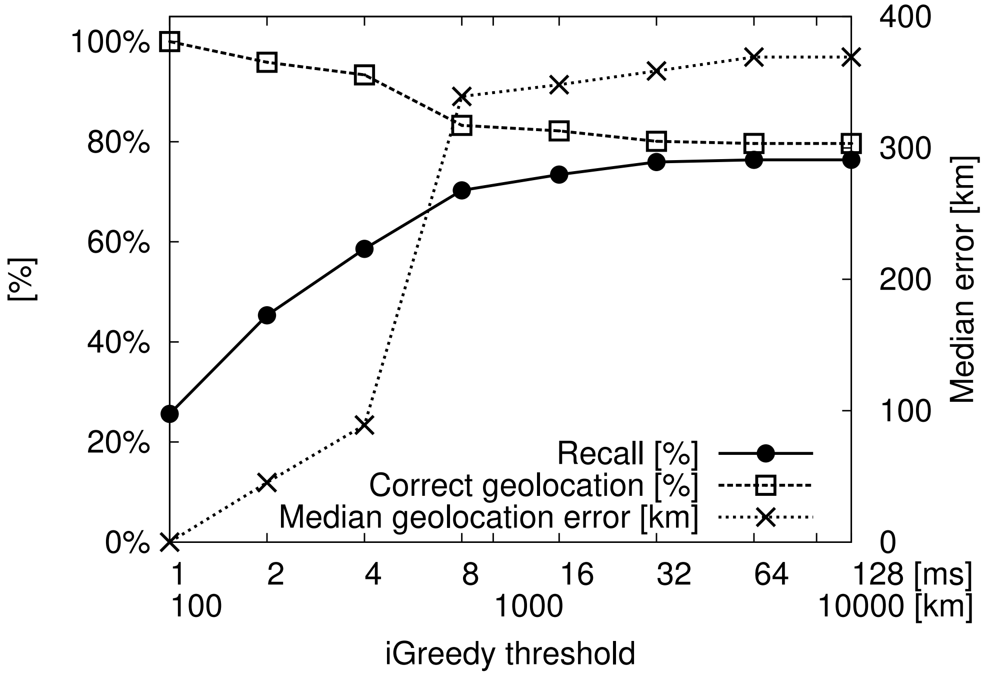


Figure 3.12: iGreedy Sensitivity: Thresholding input data (PlanetLab and RIPE)

by setting a maximum threshold: i.e., measurement larger than the threshold will be discarded, which explicitly upper-bounds the maximum error. Intuitively, this yields to a tradeoff between accuracy and completeness: indeed, each disk by definition contains at least an anycast instance, and despite the larger the disk, the harder the geolocation, however discarding large disks potentially discards useful information. Fig. 3.12 show average performance over all dataset, as a function of thresholding: clearly, small thresholds negatively affect recall, which is undesirable; instead, even for very large thresholds (e.g., over 100ms or 10000Km radius where geolocalization almost certainly fails), the completeness saturates, the correct geolocation stabilizes and the geolocation error does not increase (despite iterations). This desirable property follows from the fact that iGreedy *order disks by size* and thus *implicitly filters* out the largest circles from the solution. Hence, we do not recommend thresholding measurements to be fed

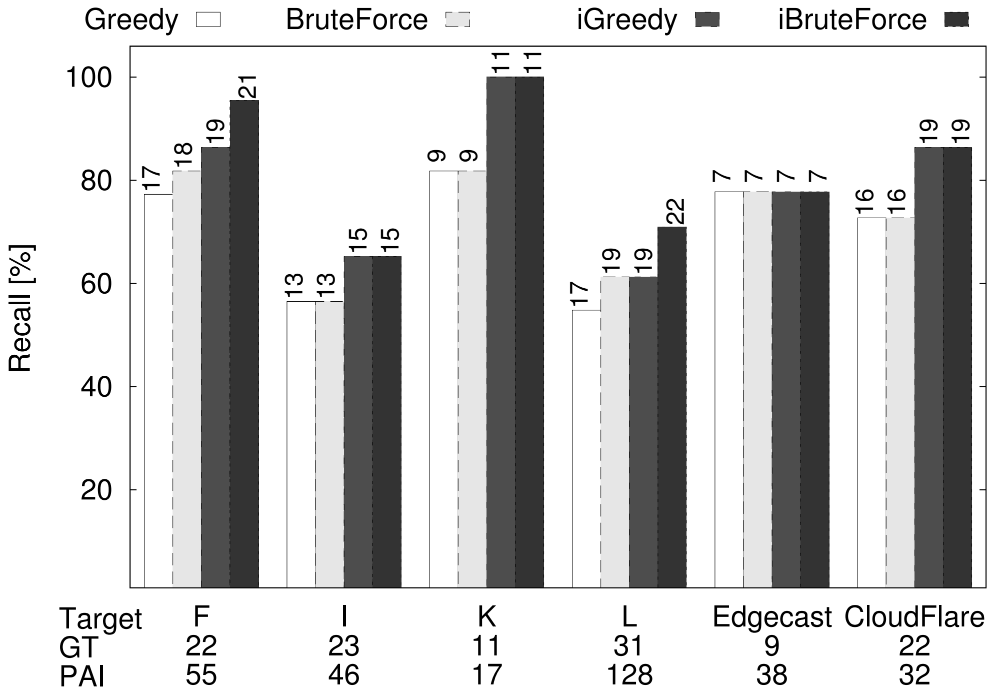


Figure 3.13: iGreedy Sensitivity: Impact of MIS Solver (PlanetLab)

to iGreedy, which is inherently robust to outliers – irrespectively of their nature such as BGP path inflation, queuing delay, etc. [121].

MIS Solver. Fig. 3.13 reports statistics about the enumeration of our targets from PlanetLab with different solvers. For each solver, the picture reports the completeness $|\mathcal{E}|$/GT; each bar annotates the number $|\mathcal{E}|$ of anycast replicas found, and the x-label is annotated with GT and PAI information for that target. It can be seen that the greedy solution is as good as that of the brute force solution (I, K, EdgeCast, CloudFlare) or anyway comparable (F, L). More interestingly, the iterative workflow produces benefits that are sizeable and consistent across datasets and solvers. Note also that most of the replicas are found in the first iteration, which keeps the number of iterations (and associated computational complexity) limited.

Solver selection is also affected by computational complexity: under this perspective, the running time of iGreedy (*hundreds of milliseconds*) is orders of magnitude smaller with respect to that of the brute force approach (*hundreds to thousands of seconds*). Indeed, that while we were able to obtain brute force solution on the PlanetLab dataset, its cost is prohibitive for the larger RIPE Atlas datasets. Thus, it also follows that, while refined solutions do exist [71, 90], they are not appealing due to the good enough performance and short running time of the iGreedy solver.

Iteration. As depicted in Fig.3.6, there are two loops: an inner loop (to solve a single MIS enumeration step) and an outer loop (the iterative feedback of iGreedy or iBruteForce). Clearly, while the complexity of the inner loop depends on the solver it is interesting to analyze the impact of the outer loop iteration. Notice that a partial answer to this question is already available from Fig. 3.13: indeed, comparing a MIS solver with its iterative iMIS version (e.g., Greedy vs iGreedy or BruteForce vs iBruteForce) one can gather an *upper bound to the number of loops* that are executed.

Notice that the picture reports a label on top of each bar, which represents the number of replicas discovered: since *at least one* replica is discovered in a new loop (else the loop terminates), then *at most* a number of loops equal to the number of newly discovered replicas through the iteration can be executed. At the same time, this estimation *upper bounds* the number of loops since it is also possible that after the classification steps (and the disk shrink), multiple non-overlapping disks are found in a single iteration. Notice further the upper bound is anyway pretty low (e.g., 0 in Edgecast, 2 for I and K and 3 for CloudFlare for both Greedy and BruteForce). It follows that the number of outer iterations is typically upper-bounded by 2-3.

To estimate the additional complexity due to iteration, we have to consider that the duration of the overall workflow is not dominated by the first iteration, and that rather each iteration can be considered to have an approximately fixed time, since all disks

Table 3.1: Overhead of iterative workflow

Target	Elapsed time (s)		Iteration
	Greedy	iGreedy	overhead
F	0.10	0.15	+50%
I	0.08	0.12	+50%
K	0.07	0.09	+20%
L	0.09	0.15	+66%
EdgeCast	0.09	0.09	+0%
CloudFlare	0.11	0.15	+36%

need to be considered at each new outer iteration in the MIS solution. We point out that albeit an optimization is possible (i.e., removing the already geolocalized disks) its impact is expected to be pretty low: indeed, this would amount at removing some few tens of disks out of several hundreds (PlanetLab) or several thousands (RIPE Atlas) vantage points, so that we do not implement it.

Overall, we can safely upper-bound the running time of iGreedy to about twice as much the running time of the simple MIS greedy solver. However, in practice average runtime increase is significantly more contained. Considering the PlanetLab infrastructure for the sake of simplicity, Tab.3.1 reports the running time (in seconds, averaged over 10 repetition) for non-iterative (Greedy) vs iterative (iGreedy) versions of the algorithms executed on a Intel i7 running 64-bits Linux OS with kernel version 3.17. The relative runtime (last column, denoted as overhead) is also reported with respect to the Greedy baseline. It can be seen that, in practice, iteration increases computational complexity by less than 50% on average.

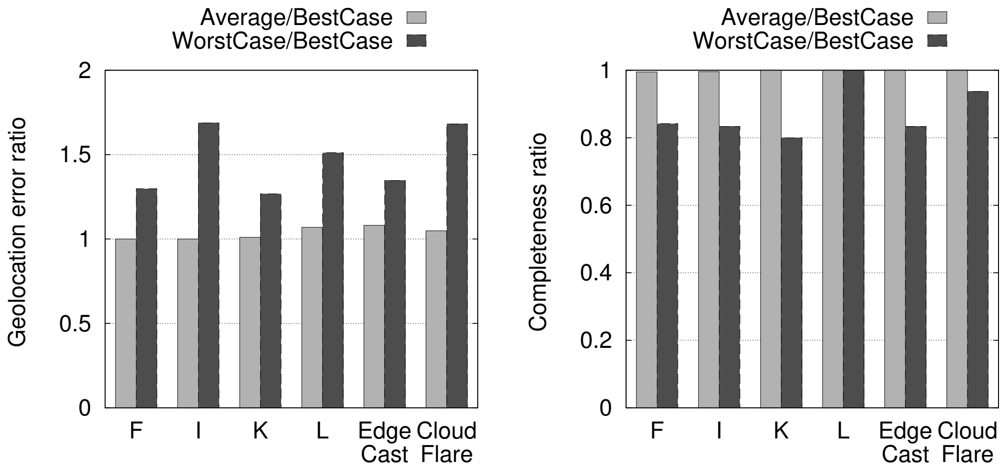


Figure 3.14: Sensitivity of iGreedy to the number of samples per vantage point: best, average and worst-case results with $P = 100$ measurement per vantage point.

5.3 Impact of measurement infrastructure and campaign

We assess the impact of MI and MC by (i) considering different combinations of vantage points from multiple MIs, and by performing latency measurement (ii) with different protocols as well as (iii) with a varying number of samples for the same protocol.

Vantage points. RIPE Atlas and PlanetLab have largely different characteristics concerning AS and country coverage, as well as vantage points footprint. We thus expect MI to play a paramount role in determining the results. Indeed, lack of VPs in an area where anycast instances lay will likely result in false negative (in case of disks overlap) or false positive (in case of a single large disk covering the area). To reduce the intrinsic VP redundancy, we also consider smaller subsets in the case of RIPE Atlas, since as previously noticed, there is an intrinsic redundancy in VP placement (e.g., we count over several hundreds RIPE Atlas hosts in Paris, albeit at most one can be useful for our purposes).

We thus consider (i) PlanetLab, (ii) a selection of 200 RIPE Atlas VPs that are at

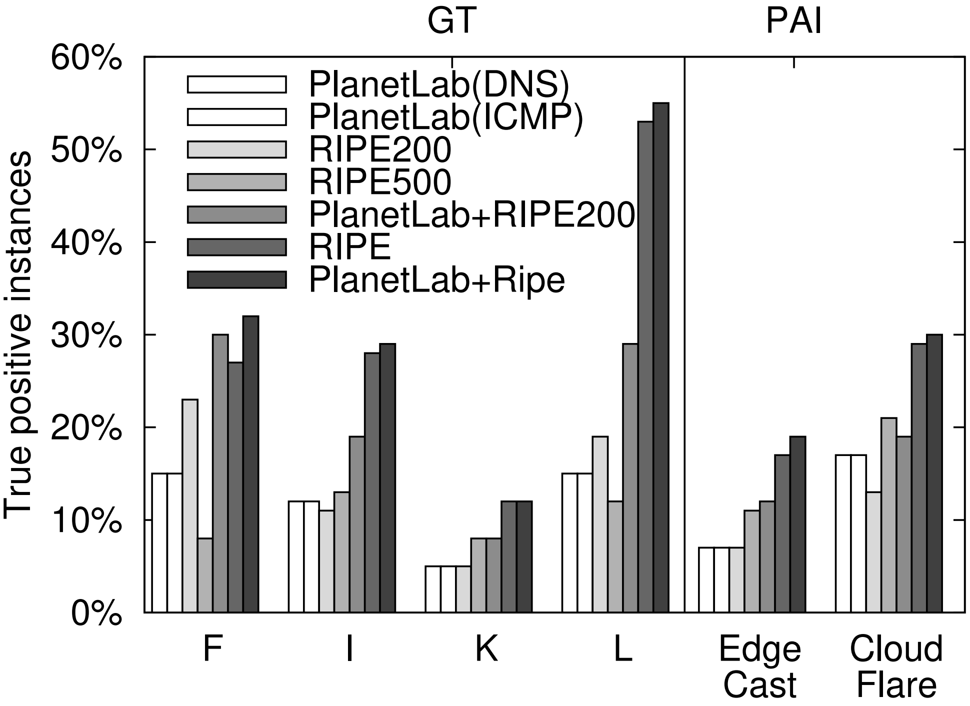


Figure 3.15: Sensitivity of iGreedy to the measurement infrastructure and campaigns. Impact of vantage point combinations and network protocols

least 100 km distant from each, (iii) a random selection of 500 RIPE Atlas VPs, (iv) the combination of PlanetLab and RIPE200, (v) the combination of PlanetLab and the full RIPE dataset. Notice that, when combining MIs, our methodology seamlessly combine DNS (from RIPE) and ICMP (from PlanetLab) measurement. Given that HTTP measurement (hence, GT information) is not available from RIPE, when combining PlanetLab and RIPE CDN measurements we compare results with the PAI.

Some remarks are in order. First, as expected, the set of VP plays a paramount role: increasing the number of VPs increase the number of valid instances that iGreedy is able to find. Second, notice that even simple selections that avoid replicas in the same city (RIPE200) are better than random selection (RIPE500). Third, combining PlanetLab and RIPE increases the enumerated replicas – which holds even considering the full set of RIPE VPs and despite PlanetLab has about 20× less VPs than RIPE. Combination of VPs is done offline: since the APIs of these MIs are totally different, it is not easy to control a single experiment jointly using VPs from both platforms. Yet, these results suggest that it would be desirable to leverage protocols such as LMAP [103] or mPlane [125] to more systematically exploit both platforms in a unified interface.

Network protocol. RTT latency can be obtained with several protocols other than network-layer ICMP measurements. As a matter of fact, a side-product of our DNS and HTTP ground-truth measurement campaigns, we obtain delay measured with application-layer protocols. Notice that for DNS CHAOS requests, RTT latency include the server response time, whereas for HTTP HEAD we rely on the TCP three-way handshake duration (as including the server response, would overestimate the distance by about a factor of two). Of course, GT campaign only offer a single latency sample, which is however not expected to have a major impact as just outlined. By running iGreedy over latency samples gathered by these different protocols, *we do not observe any quantifiable difference in the results* (for completeness, this absence of variability is illustrated in Fig.3.15 in the PlanetLab DNS vs ICMP case, but we experimentally confirm it to hold over all PlanetLab targets and protocols combination). While robustness stem from the fact that iGreedy do not geolocalize based on latency measurement, this reinforces the soundness of ICMP choice to jointly achieve protocol-independence and correctness at the same time.

Varying number of samples. Results with varying number of P latency measurements are reported in Fig.3.15. Specifically, we run a measurement campaign gathering 100 ICMP samples per VP-target pair, and then consider: (i) the best case where $\delta^{BC}(p,t) = min_i \delta_i(p,t), \forall p,t$; (ii) the worst case where $\delta^{WC}(p,t) = max_i \delta_i(p,t), \forall p,t$; (iii) the typical case where $\delta^{TC}(p,t)$ are extracted at random from the $\delta_i(p,t)$ population.

We next run iGreedy over such sets, and compare the performance over these sets: notice that while best and worst cases are deterministic, for the typical case we consider 100 such sets, so to obtain performance that are statistically representative of an average case. To quickly give an idea of the robustness of iGreedy performance to *real* input data noise, Fig.3.15 reports the mean error (left) and the completeness (right) for the worst and average cases, normalized over the best case as a reference. As it can be seen, performance deteriorates slightly even in the unlikely worst case where noise is maximum for all vantage points. More interestingly, performance of the *average case are almost indistinguishable from the best case*: this suggests that *even less than a handful of measurements per vantage point* is sufficient to correctly geolocalize anycast replicas with iGreedy.

6 Summary

Use of anycast has increased in the last few years, venturing out of the DNS realm and revealing a sizeable footprint in, e.g., CDN services. At the same time, measurement techniques for anycast infrastructure discovery are either protocol agnostic but limitedly offer detection capabilities [104], or offer enumeration capabilities but are limited to DNS [72].

The contributions of this chapter are to provide the researchers and practitioners communities with (i) iGreedy, a tool able of lightweight, protocol-agnostic anycast replicas enumeration and geolocation on the one hand, and with (ii) a suite of datasets and software, to facilitate the development, validation and comparison of new techniques on the other hand.

The key to iGreedy performance is the formalization of the *enumeration step* as a Maximum Independent Set problem to maximize the replica coverage; and formalization of the *geolocalization step* as a classification problem, that leverages city population to factor out noise in the latency measurements. iGreedy leverages latency measurement from any protocol, and can seamlessly integrate measurements from heterogeneous protocols. Being inherently robust to outliers make the technique extremely lightweight: even a single latency sample per vantage point, from a few hundreds vantage points suffices to provide satisfactory enumeration and geolocation performance. In reason of its lightweight and its low computational complexity, the technique is amenable to continuous large scale measurements, which is hopefully helpful in refining our understanding of the Internet.

Chapter 4

IPv4 Anycast Adoption and Deployment

In this chapter, we provide a comprehensive picture of IP-layer anycast adoption in the current Internet. We also reveal that, unlike common belief in the scientific community, anycast usage is not restricted to single request-response applications over UDP (e.g., DNS), but also includes L7 applications running over stateful TCP connections such as cloud services, web hosting, and mitigation of DDoS attacks.

Indeed, while valuable research efforts (Sec. 1), started with seminal work such as [83] and culminated with [8, 67] in more recent times, focus on *unicast censuses*, this work presents the first census of the use of IPv4 *anycast* in the Internet. In more details, we conduct an experimental study of anycast adoption based on distributed delay measurements that we collect from multiple anycast IPv4 censuses (Sec. 2). We also describe the challenges in performing such large-scale measurements (Sec. 3) and complement the spatial results with a longitudinal view of the anycast evolution (Sec. 4). To summarize our main contributions:

- We conduct and combine delay measurements from numerous censuses, based on which we find about $O(10^3)$ IP/24 subnets to be anycasted.

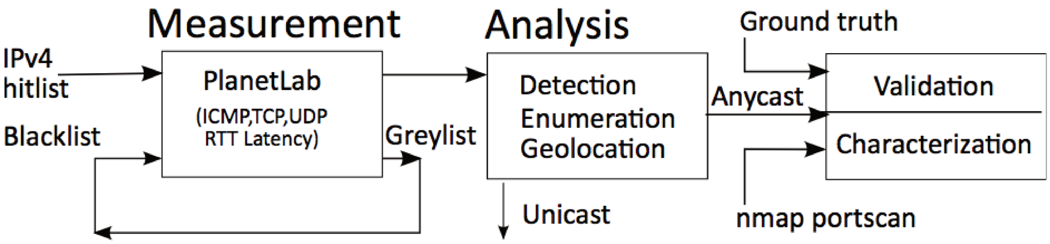


Figure 4.1: Anycast census at a glance: Workflow

- 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 quantitatively debunk the long-standing myth that IP anycast is relegated to stateless services such as DNS over UDP, and show that a large diversity of stateful services run on top of TCP.
- We describe our distributed system design, able to perform and analyze one census in few hours.
- We provide a brief longitudinal study of the anycast ecosystem.
- We make all our datasets (raw measurement from PlanetLab and RIPE Atlas), results (monthly geolocated anycast replicas for all IP/24) and code available to the community.

1 Methodology Overview

In this section, we present an overview of our work (Sec. 1.1). For the sake of readability, we describe our complete workflow with the help of Fig. 4.1.

1.1 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 [6]). 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 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 the run iGreedy, presented in Chap. 3 to detect, enumerate, and geolocate anycast replicas over the dataset. We modified and improved the code to apply the methodology to the full census in Sec. 2.1 and discuss the top-100 anycast ASes in Sec. 2.2.

Characterization and Validation. In addition to anycast detection, the previous steps allow to geolocate the replicas behind each anycast IP/24. We validate the accuracy of the anycast geolocation technique whenever a ground truth is available (as in Sec. 3.4 for CDNs such as CloudFlare and Edgecast, complementary to the validation over DNS in [56]).

While our detection methodology is service-agnostic, we use nmap [102] on a list of anycast IPs obtained from the census to provide a fine-grained characterization and reveal the anycast services they offer. 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. 2.3, 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 Chap. 3, 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 earlier) 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 $O(10^4)$ [43] to $O(10^5)$ [72] vantage points to achieve ≈90% recall [72]. 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 multiple IPv4 censuses and analyse the results obtained from their combination. For each of the $O(10^2)$ VPs the magnitudo of a typical census is illustrated in Fig. 4.2: starting from a hitlist of $O(10^7)$ targets, less than half send a reply (Sec. 3.1). ICMP replies include $O(10^5)$ error codes relating 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

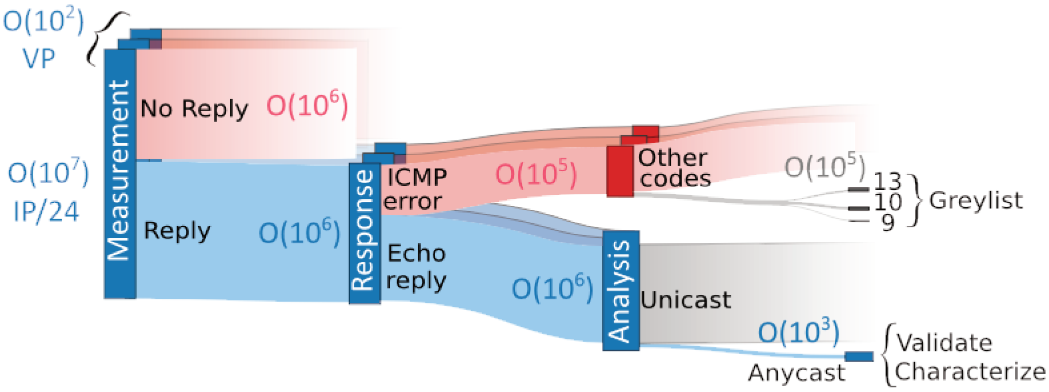


Figure 4.2: Anycast census at a glance: typical census magnitude

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 Anycast /0 census

This section presents results of the first Internet-wide anycast study. We start by aggregated statistics (Sec. 2.1) and then incrementally refine the picture by providing a bird's-eye view of the most interesting deployments (Sec. 2.2) over which we perform an additional portscan campaign to reveal their running services (Sec. 2.3).

2.1 At a glance

Details about the output of our censuses are reported in Fig. 4.3. 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 [63], offering aggregated (as in Fig. 4.3) or per-deployment (cf. Sec. 3.2) visualizations.

	IP/24	**ASes**	**Cities**	**Countries**	**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

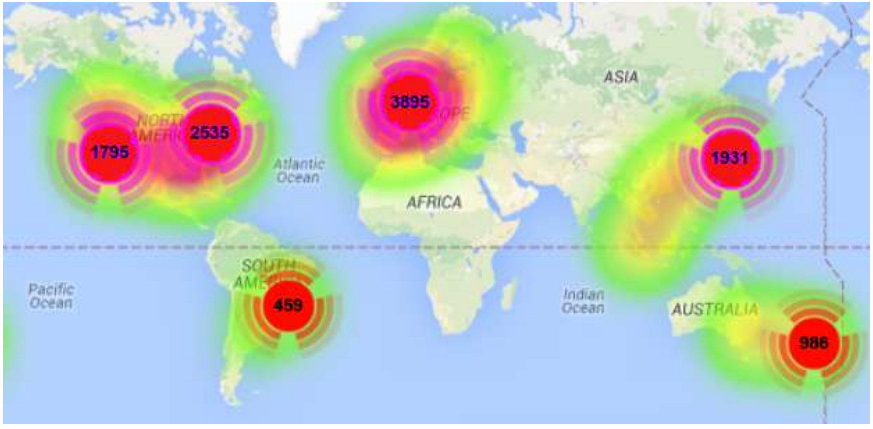


Figure 4.3: Anycast census results at a glance.

Several remarks 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 new anycast replicas but they will not be considered in the solution of the MIS problem (recall enumeration phase in Chap. 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. 4.3, show that 19 IP/24 of 8 ASes belong to the list. Similarly, we investigate the Alexa rank, to cross check how many webpages in the top-100k rank are hosted[1] by ASes using IP-anycast: here again, we find 242 IP/24 of 15 ASes. We thus infer that, contrary to common belief relegating its usage to DNS, IP-anycast is used

[1] 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.

by ASes that are play a central role in the Internet, as well as by ASes that are among major players of the Web. Finally notice that only Tinet (AS3257) appears in both CAIDA-100 and Alexa-100k lists: as none of these two rankings alone could possibly provide an exhaustive coverage of IP-anycast, it follows that the workflow we propose in Sec. 1 is needed to gather a comprehensive view such as the one provided here.

2.2 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. 4.4 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 bottom x-axis. *Geographical and IP/24 footprint* are reported in the bottom part of the plot: 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.

Big fishes. It is fairly easy to spot major players of the Internet ecosystem in Fig. 4.4. 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.,

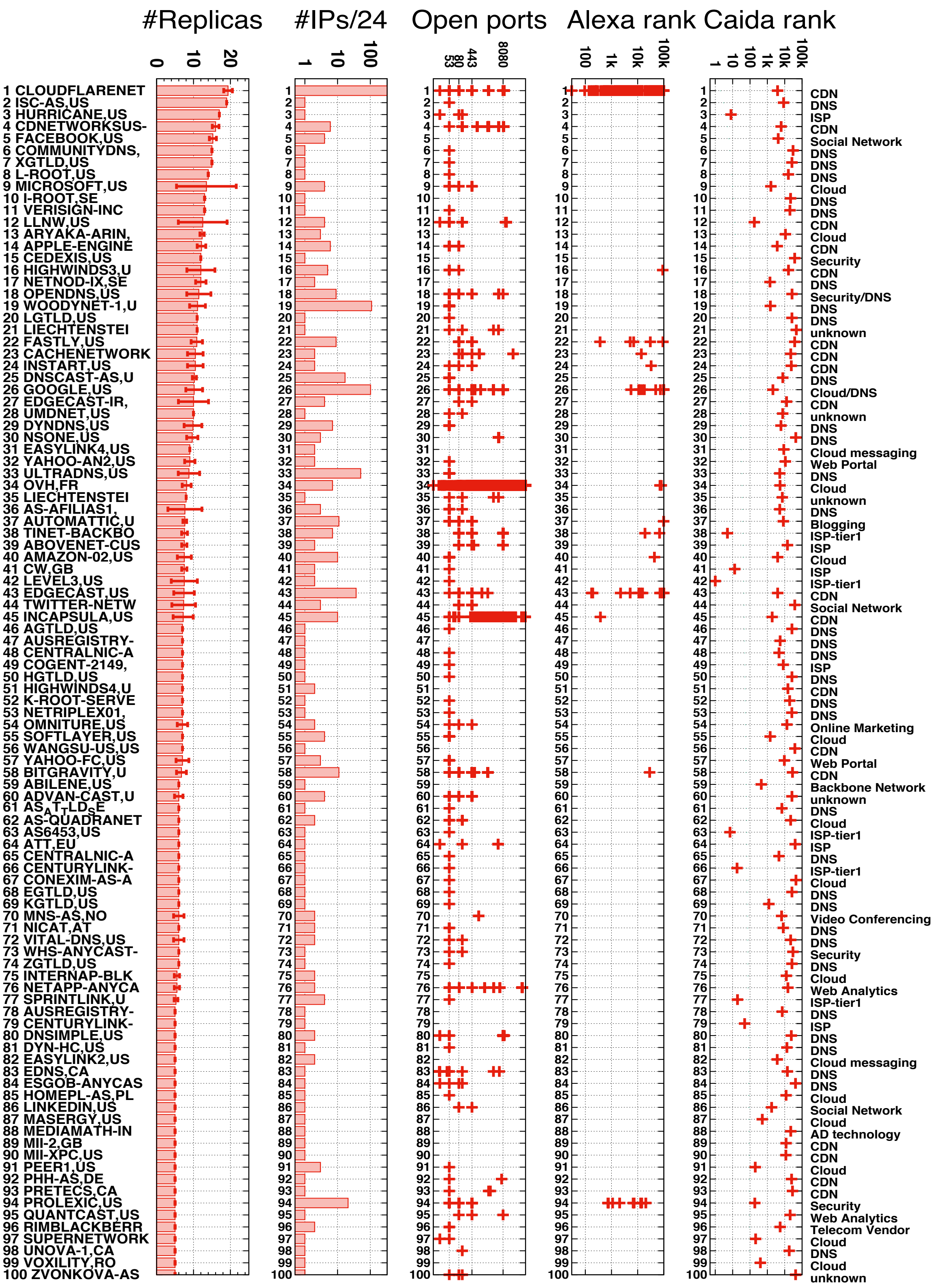


Figure 4.4: Bird's eye view of Top-100 anycast ASes (ranked according to geographical footprint)

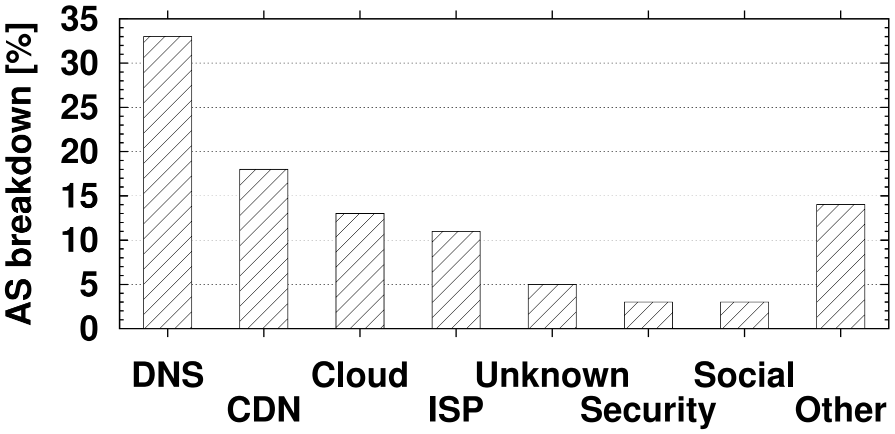


Figure 4.5: Breakdown of AS classes.

Twitter, Facebook, LinkedIn), and security companies that provide mitigation services against DDoS attacks (e.g., OpenDNS, Prolexic). 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. 4.5, crisply showing that DNS now represents about one third of IP anycast activities. Plots in Fig. 4.4 clearly illustrate the large diversity of anycast usage, under all metrics. Indeed, no correlation appears 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* (first plot from the bottom in Fig. 4.4) championed by the CDN CloudFlare in our measurement. Overall, we observe that 25 ASes 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 easily exceed $O(10^3)$, 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. 4.6 further reports the cumulative number of replicas per IP/24, depicting both individual results about each census in isolation, as well as results coming from a combination of censuses. 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. 4.6, 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.

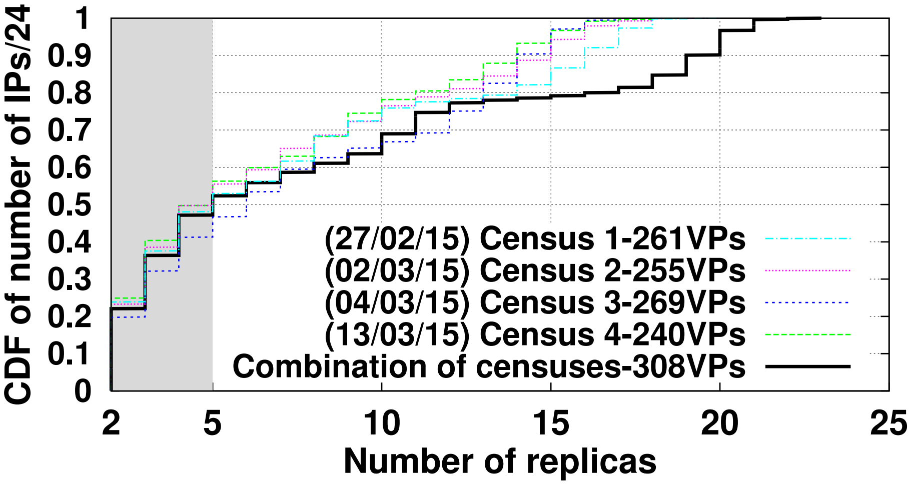


Figure 4.6: CDF of geographically distinct replicas per IP/24 (individual censuses and overall)

In what follows, we limitedly consider results from the combination, and remark that results are quite consistent across censuses (notice that curves overlap in Fig. 4.6). 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.

IP/24 footprint. In terms of the the *number of anycast IPs/24 per AS* (second plot from the bottom in Fig. 4.4), 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. 4.7 shows that about half have exactly one IP/24 (e.g., LinkedIn and

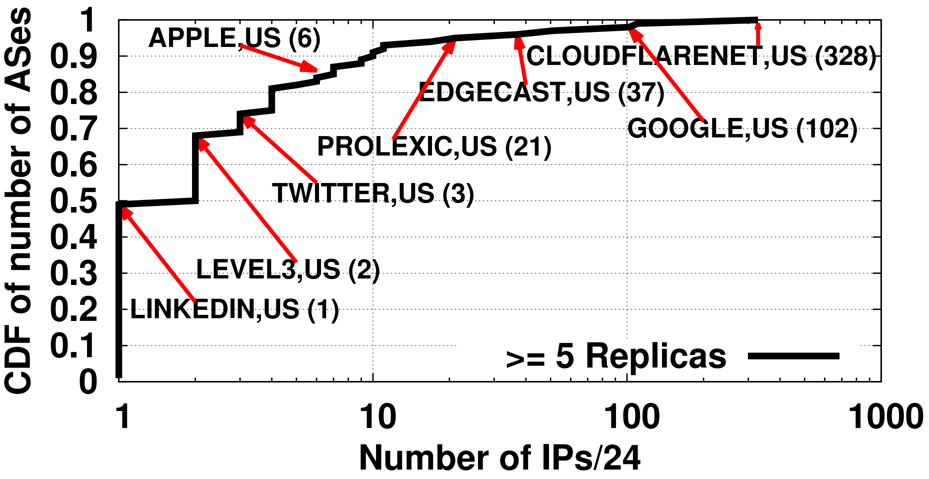


Figure 4.7: Number of IPs/24 per AS

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. 4.5, 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

nmap portscan statistics

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

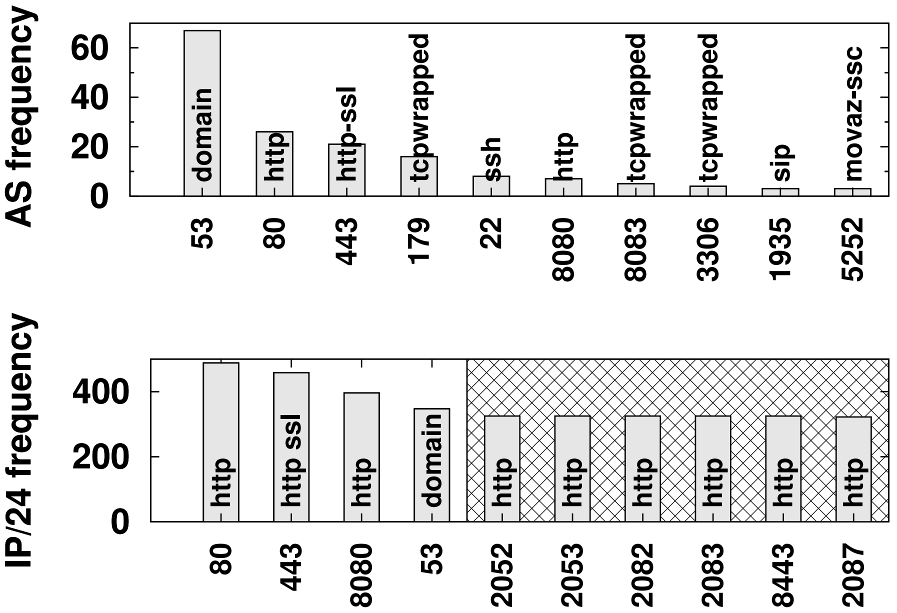


Figure 4.8: Overall nmap portscan statistics and Top-10 open TCP ports (per AS and per /24).

listed by Alexa are hosted by Google anycast IPs. Finally, 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.

2.3 Anycast Services

Portscan campaign. In reason of the long-standing myth relegating anycast usage to stateless, and especially DNS, services, we believe it to be important to provide a 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. 4.8.

We test all anycast IP/24 of the top-100 ASes and pick the single IPs/32 representative per IP/24, we scan, at low rates, all 2^{16} TCP ports. Our results are conservative in that different IP/32 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. 4.8 the frequency count of the top ten open TCP ports by number of IPs/24 (bottom) and number of ASes (top). 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.

Stateful services. Fig. 4.9 presents the complementary CDF of the number of open TCP ports per AS. We make the following observations: (i) roughly half of the ASes (53%) have at least one open TCP port, (ii) about 12% 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

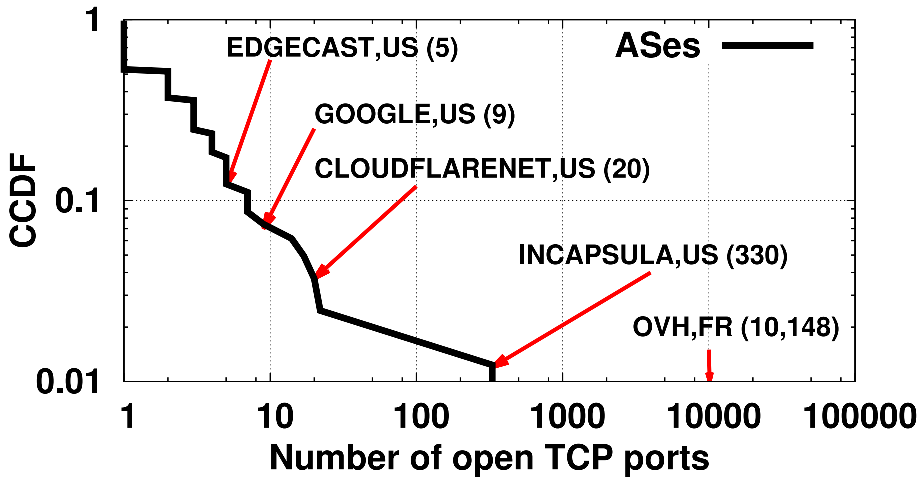


Figure 4.9: Complementary CDF of the number of open TCP ports per AS.

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 [116]. 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. 4.10 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 most adopted protocol to handle DNS requests over anycast.

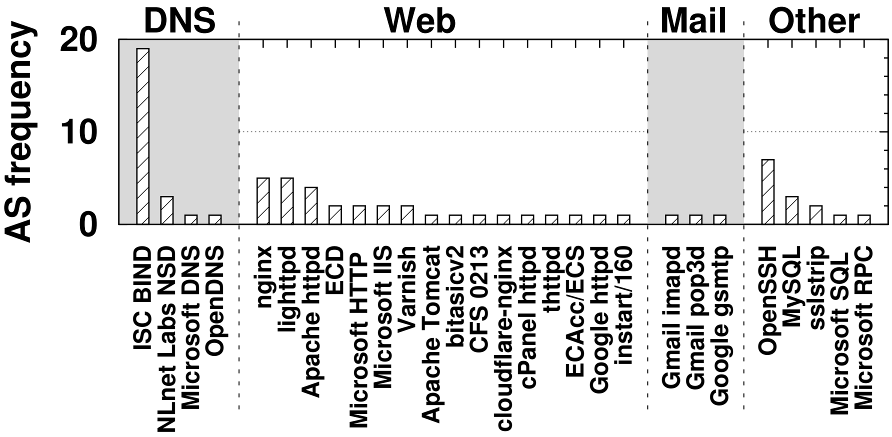


Figure 4.10: Breakdown of software running on anycast replicas.

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 andcPanel 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 [29] 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).

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 allowed us to gather the census results described so far. 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 [32] is to ignore or block prefixes shorter that /24, which implies a granularity of at least /24 for anycasted services. Additionally, while in principle we could use one IP/32 per each announced BGP prefix, [30] observes that prefixes longer than /24 have low visibility: as such, we limit the granularity to IP/24 level (i.e., one IP/32 per /24) targeting less than *0.4%*of the whole address space. It follows that any alive /32 belonging to the /24 is equivalent in telling whether the whole /24 is anycast (or unicast): spot verifications for all IP/32 on some IP/24 belonging to EdgeCast confirm this assumption, validating a /24 granularity.

Target liveness. We rely on the USC/LANDER hitlist periodically published by [6] to get a *responsive* IP address in every /24-prefix. The hitlist consists in generally one representative IP address for $O(10^7)$, 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 (with 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.

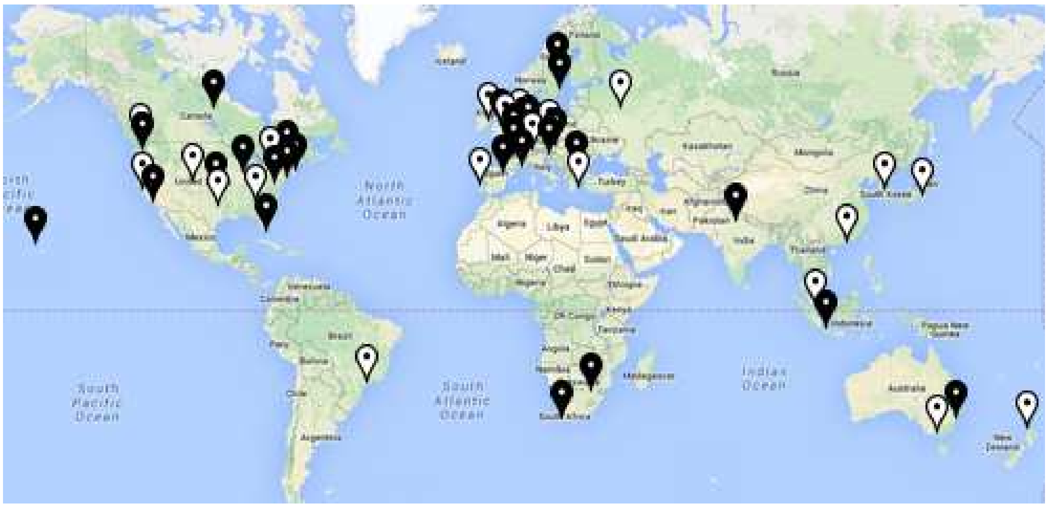


Figure 4.11: Microsoft deployment as seen from PlanetLab (21 replicas) vs RIPE Atlas (54 replicas). Notice that PlanetLab results (white markers) are a subset of RIPE Atlas(white *and* black markers)

Coverage. Given our census aim, we verify how well this hitlist covers all routed /24 prefixes, finding a 99.99% coverage of the RIPE and RouteViews BGP collectors. We additionally cross-check our observed target responsiveness with the expected recall. Specifically, recent ICMP scans [131] observe $4.9 \cdot 10^6$ used /24 subnets: our campaigns capture $4.4 \cdot 10^6$ responsive subnets, with thus a 90% recall with respect to [131].

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 [14] 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. As introduced in Sec. 2, several measurement platforms exist that have different characteristics in terms of vantage points cardinality, AS diversity, geographic coverage and limits, in terms of probing traffic or rate [40]. In this work we make use of two platforms, namely PlanetLab and RIPE Atlas, that we select due to their complementarity. In the first part, we rely only on the TopHat Dedicated Measurement Infrastructure (TDMI) [124] that uses PlanetLab. While RIPE is more interesting for geographical diversity due to its scale, it has a limited control on the rate (cf. Sec. 3.3) 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). Additionally, the larger number of vantage points would mechanically increase about 20-fold the amount of probes per census with respect to PlanetLab (in case all VPs are used). Conversely, TDMI is limited by PlanetLab node availability (generally around 300 vantage points, but it is decreasing over time, refer to Sec. 4), but offers full flexibility for deploying custom software on PlanetLab and run it at high speed. We use it to perform exhaustive censuses to a large set of targets, that we expect to be mostly unicast.

We depict for illustration purposes an application of our technique from measurements collected from PlanetLab vs RIPE Atlas in Fig. 4.11: as PlanetLab 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 Atlas measurements. It follows that to improve the results, we need to *combine* both platforms: thus, we use RIPE Atlas to refine the the geolocation of anycast /24 detected via PlanetLab(Sec. 4).

3.3 Measurement software

Fastping. An efficient measurement is needed to maximize the probing capacity of our VPs. While at first sight Zmap [67] 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 PL virtualization layer. We therefore resort to Fastping [73], 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 [41] (type 3 code 13) and the remaining in reason of communications administratively prohibited at network or host levels (respectively 1.3%, code 10 [48] and 0.2%, code 9 [48]).

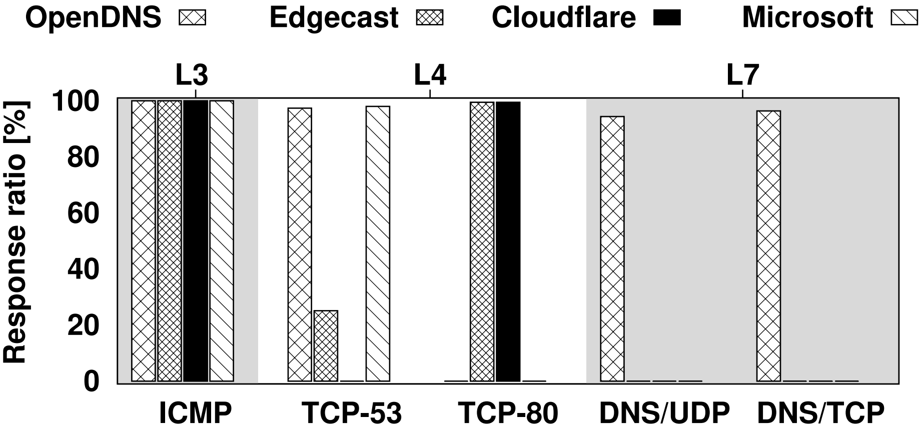


Figure 4.12: Response rates seen by heterogeneous protocols across different targets.

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 [114], 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 [131]. 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. 4.12 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 dis-

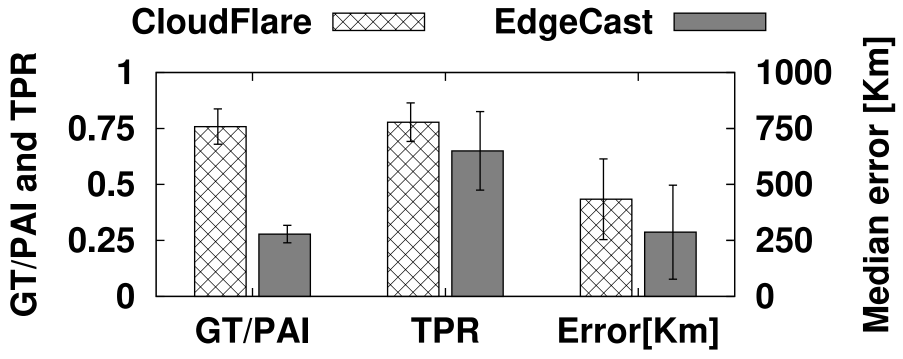


Figure 4.13: Validation with CloudFlare and EdgeCast ASes. Bars represent standard deviation among IP/24 of the AS.

criminative 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 [56], 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 measurement with `curl\` from PL: 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. 4.13: 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 Atlas.

Consistency. Additionally, in the case of openDNS, we verify consistency across multiple RTT latency measurement techniques used early in Fig. 4.12. In this case we rely on public information that maps 24 locations [3]. For all protocols, applying [56] 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 [56] toward city population (Philadelphia is 33 times more populated than Ashburn), but as observed in [51] 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 took 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 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[2], that we

[2] While it would be 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.

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

Table 4.1: Textual (0) vs binary (1-4) censuses

verified empirically not triggering the above problems. Consequently, probing $6.6 \cdot 10^6$ targets at 10^3 takes less than two hours: as shown in Fig.4.14 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).

Output size and analysis duration. A second issue we initially overlooked concerns the output format. Fastping indeed logged, in textual format, a wealth of information amounting to 270M per node and 80GB overall per census (cf. Tab.4.1). 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 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. Textual format was already a culprit: due to disk fragmentation, reading large text files could also lead to very inefficient processing (we indeed have gathered a complete Census-0 in this format, but stopped its processing after 3 days of CPU time). Even with binary files, only $O(10^3)$ subnets are found to be anycast, all the $O(10^6)$ subnets for which we have valid echo reply samples need to be processed. For a single target, the running time of [56] is $O(10^{-1})$ sec, 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*. Additionally, 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

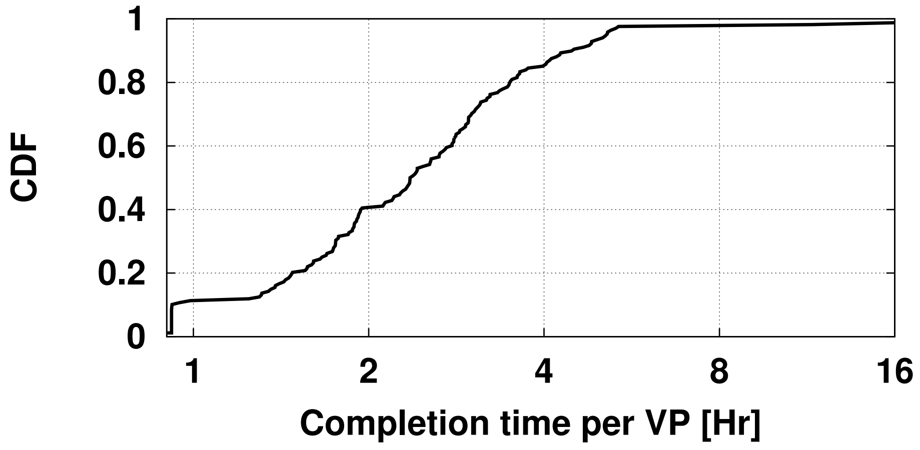


Figure 4.14: CDF of census completion time

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.

4 Temporal evolution

In the previous sections, we present a very complete and detailed view of the *spatial* characteristics of anycast deployments, e.g. their geographical distribution and the services offered over anycast, however this study represents a snapshot at a fixed point in time (four censuses in a month). Thus, we decide to complement the picture with a (brief) longitudinal study: for more than one year, we perform regular IP monthly IP anycast censuses. Thus, we provide a broad picture including all deployments (Sec.4.2), as well as a more detailed view by cherry-picking some representative ones (Sec.4.3). Without loss of generality, we refer to the last year worth of censuses collected between May 2016 and May 2017 and make all our datasets (raw measurement from PlanetLab and RIPE Atlas), results (monthly geolocated anycast replicas for all IP/24) and code available to the community.

4.1 Measurement campaign

Targets. For the selection of the targets, we rely on the USC/LANDER hitlist [6], providing a list of (likely alive) target IP/32 host per each /24 prefix. Every two months the list is updated, and so our target selection. We only consider hitlist IPs that have been successfully contacted (i.e., denoted by a positive score [6]), which leaves us about 6.3 millions potential targets (out of 14.7 millions).

Vantage points (VP). We perform VP selection as follows: in PlanetLab, where the total number of VPs is small (and decreasing), we simply select all the available ones; in RIPE Atlas, where the number of VPs is large and due to credits limit, we carefully select 500 VPs, making sure that each VP is far from the others by at least 200 km (roughly 2ms). Similar results can be obtained by clustering VP together geographically and performing a stratified selection in each cluster [77].

Notice also that the selection of a limited number of VPs is necessary to limit the measurement stress on the infrastructure, so that an anycast IP/24 census generate roughly the same amount of probes than a full IP/32 census. Clearly, increasing the cardinality and diversity of the VP set, as well as performing multiple measurements to reduce the lantency noise (i.e., using the minimum over multiple samples) would otherwise yield more complete and accurate results. We point out that these supplementary measurements could be performed *after* the anycast detection step, significantly limiting the subset of IP/24 requiring additional probing.

Fig. 4.15 shows the evolution of the number of available vantage points during our campaign. In the PlanetLab case, the number of nodes available drastically decrease from 300 in March 2015 (month of the spatial study) to roughly 50 in May 2017.

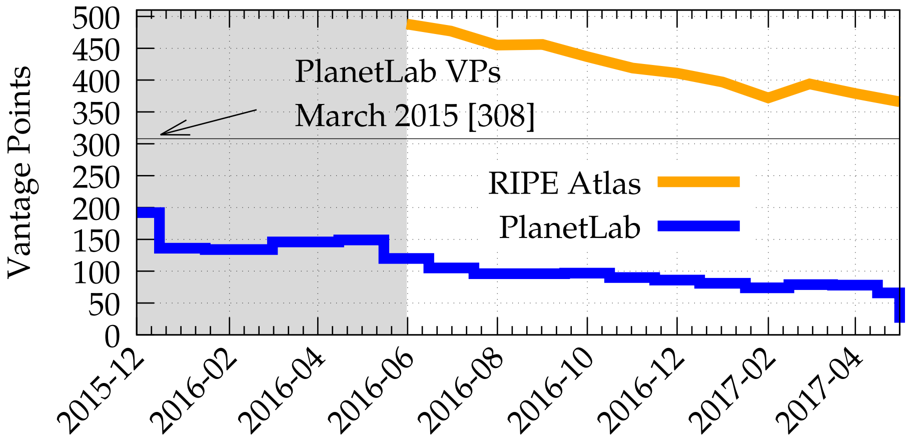


Figure 4.15: Measurement campaign: evolution of number of PlanetLab and RIPE Atlas VPs

In the case of RIPE Atlas, the decrease is due to the fact that we launched a long-standing periodic measurement in June 2016 with an initial set of VPs, some of which later become unavailable. Interestingly, we will see that anycast results appears to be consistent despite this decrease. As early stated, despite a handful of carefully selected vantage points [77] allow to *correctly detect* anycast deployments, it is clear that the shrinking size of the available PlanetLab VP it is not adequate to *thoroughly enumerate* all the locations of an anycast deployment – for which RIPE Atlas measurements become necessary as we shall see.

Censuses. Anycast censuses require the same target to be probed from multiple vantage points: to limit the intrusiveness of our scans, and since we expect that changes in the anycast deployments happen at a low pace, we decide to run scans at a *monthly* frequency. We decided to re-engineer our system and started to run monthly censuses from PlanetLab in December 2015. We kept tuning and improving system performance and reliability until June 2016, date at which we additionally started the measurements

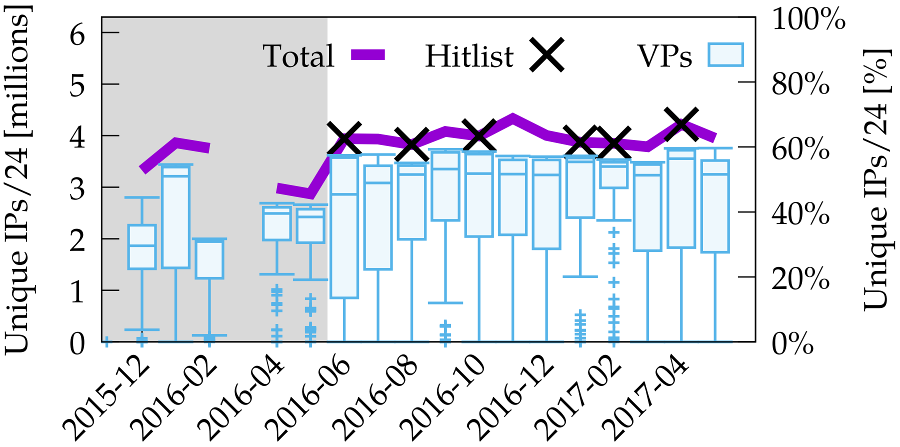


Figure 4.16: Measurement campaign: box plot of the number of responsive unique IPs/24 across all PlanetLab VPs.

from RIPE Atlas. We opted to strip down as much as possible the information collected per VP, narrowing down to about 30MB per VP on average, so that the (compressed) raw PlanetLab measurement data hosted at [12] amount to about 60GB per year worth of censuses. The RIPE Atlas measurement are publicly accessible via RIPE Atlas (measurement identifiers are at [12]).

Fig. 4.16 shows the number of unique IPs that responded to at least one of our PlanetLab VPs in each census, and the right y-axis reports this number as the fraction of replies from the contacted targets. The shadowed part indicates the months where we were still updating the system. Notably, we slowed down the probe rate per VP to about 1,000 targets per second to comply with recommendations in [81], noticing a decrease in the packet loss rate as a beneficial side effect. We can see that the total number of unique IPs is always greater than the number observed by a single VP, and that it fluctuates between 2,9 millions (May 2015) and 4,3 millions (Nov 2016) coherently with [61, 131]. This number has increased since June 2016 when we started to regularly update the hitlist [6] (denoted with crosses). However, notice that even with fresh hitlists not all targets are responsive, which correlates with the availability score of the targets: particularly, the average score for responsive targets (89) is higher than the score of non-responsive ones (40). The figure also reports the distribution of responsive IPs per vantage points (box plots): the recall varies widely per census, per VP, and over time, with some VPs able to collect only few hundred ICMP replies. Luckily, albeit the number of PlanetLab VPs decreases, the median number of contacted targets consistently exceeds 3 millions.

4.2 Results: broad view

Longitudinal view. First, we assess the extent of variability of anycast deployments. We start by considering an IP/24 granularity, and depict in Fig. 4.17 the evolution of the number of IP/24 anycast deployments, i.e., the number of deployments that have been found to be anycast by running iGreedy [57] over PlanetLab measurements. We recall that iGreedy requires to solve a Maximum Independent Set (MIS) optimization problem for each of the over 4 million responsive targets every month: the code available on GitHub [13] is able to complete the analysis of a census in few hours, which returns the set of geolocated replicas $\mathcal{G}_t$ for each responsive IP/24 target t. While full details of the geolocation for each target and over all months are available online as a Google-map interface [12], in this paper we limitedly consider the footprint $G_t = |\mathcal{G}_t|$ of the deployment, i.e., the number of distinct instances irrespectively of their location.

The figure shows that in our censuses, the number of anycast deployments has slightly (+10%) increased in the last year, peaking in April 2017 at 4729 IP/24 belonging to 1591 routed BGP prefixes and 413 ASes. In the last six months, the number of anycast deployments has never dropped below 4500 while in June 2016, when we started the censuses regularly, we found only 4297 IP/24, 1507 BGP prefixes and 379 ASes. Compared to spatial study of March 2015 presented previously, this represents a 2,5-fold increase in detected anycast instances over a period of 2 years. This may be due

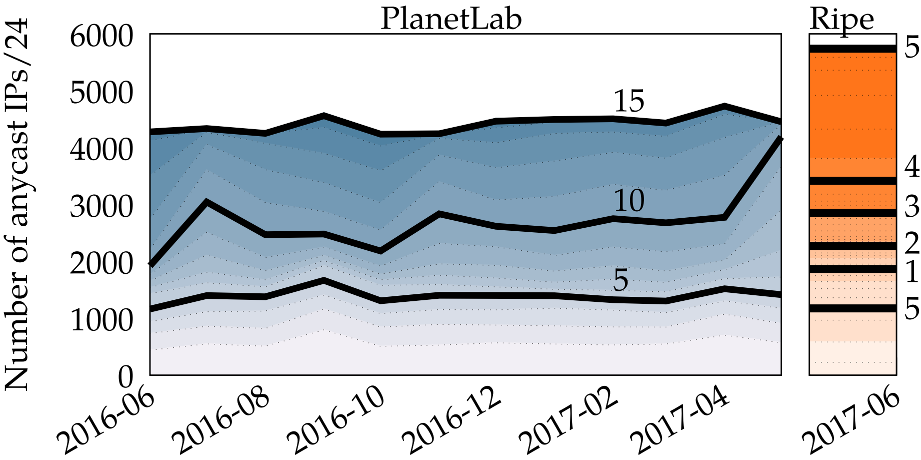


Figure 4.17: Broad longitudinal view of anycast evolution: Number of IP/24 anycast deployments (y-axis) and breakdown of their geographical footprint (heatmap and contour lines) in PlanetLab (left, over the last year) vs RIPE Atlas (right, last month).

to several reasons: part of it is rooted in increased anycast adoption over time, whereas another part is rooted in system improvements to reduce packet losses at PlanetLab monitors (during the gray-shaded beta-testing period), which increases the recall. This also means the previous results are fairly conservatively assessing the extent of the anycast Internet.

Fig. 4.17 additionally encodes, as a heatmap, the estimated geographical footprint G_t, where deployments are ranked from bottom to top in ascending size (equivalently, darker colors). A few contour lines indicate the cumulative number of deployments having no more than 5, 10 or 15 replicas. Interestingly, Fig. 4.17 shows that, despite a shrinking number of PlanetLab VPs, the number of anycast IP/24 remains steady over time. Particularly, the number of deployments having few replicas (e.g., 5 or less) remains flat over time, hinting to the fact that the geographical coverage of PlanetLab is still enough to correctly detect most anycast deployments.

Yet, as previously observed, the shrinking number of PlanetLab VPs surely affects the completeness of the replica enumeration. We thus complement PlanetLab censuses with a refinement campaign from RIPE Atlas, which is also reported in Fig. 4.17: during June 2017, we target all IP/32 that have been found to be anycast in PlanetLab during the previous year. Out of the overall 5841 IP/24s, approximately 300 were not reachable in June 2017 and 5105 IP/24s are confirmed to be still anycast. Particularly, we used 500 RIPE Atlas VPs, i.e., about one order of magnitude more than PlanetLab, which ensures a good geographic coverage (although, admittedly, the results could be refined further by increasing the VP set and the number of latency samples per VP). Thus, while PlanetLab may provide a rather conservative lower bound of the actual footprint for a target t, we expect $G_t^{RIPE} > G_t^{PL}$. Fig. 4.17 confirms these expectations: in several cases, the number of anycast instances discovered in RIPE Atlas doubles with respect to PlanetLab, and the maximum number exceeds 49 replicas (18 in PlanetLab). Overall, according to RIPE Atlas, half of the deployments are in more than 5 different locations, but only few of them have more than 35 locations (including DNS root servers, Verisign, Microsoft, WoodyNet Packet Clearing House (PCH) and Cloudflare). Consider also that as a consequence of the drop in the number of PlanetLab VPs in the the last months, the largest footprint measurable from PlanetLab drops as well (notice the sharp increase for deployments with at most 10 replicas). This confirms that PlanetLab remains useful for anycast detection, but also that RIPE Atlas becomes necessary for enumeration and geolocation, reinforcing the need for a more systematic coupling of complementary measurement infrastructures such as PlanetLab and RIPE Atlas.

Aggregation level. Clearly, while we operate censuses at IP/24 level, it is then possible to aggregate the information at BGP or AS level. Denoting with $\mathcal{S}_x$ the set of IP/24 included in a BGP-announced prefix (or an AS) x, we can define the *spatial IP footprint* as $S_x = |\mathcal{S}_x|$. By extension, we can define the BGP-level (AS-level) geographic footprint G_B (G_A) by considering only the largest IP/24 in the prefix $G_B = \max_{t \in \mathcal{S}_B} G_t$ ($G_A =$

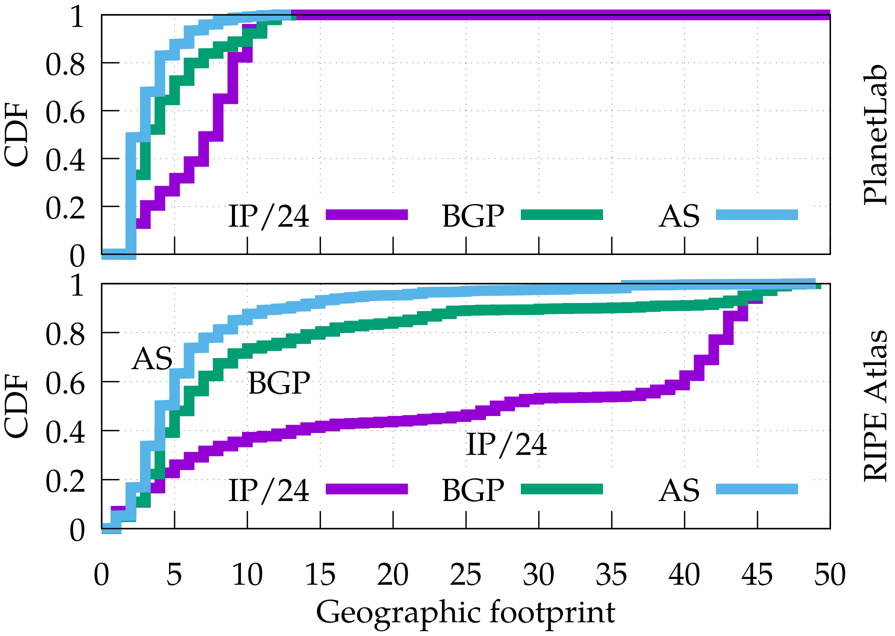


Figure 4.18: Distribution of the geographic footprint of anycast deployments at IP/24 (G_t), BGP-announced prefix (G_B) and AS level (G_A). Results from PlanetLab (top, all months) vs RIPE Atlas (bottom, last month).

$\max_{t \in \mathcal{S}_A} G_t$). To perform this aggregation step, for each month in the census dataset, we retrieve the AS and prefix information using all the RIPE-RIS and RouteViews collectors with BGPStream [111], and cross-validate the information using the TeamCymru IP to ASn service [7].

The different viewpoints are illustrated in Fig.4.18 that reports for PlanetLab (top, all months) vs RIPE Atlas (bottom, last month) the cumulative distribution function of the geographic footprint at IP/24, BGP-announced prefix and AS levels. The geographic footprint per-IP/24 vs per-BGP/AS varies widely, which is due to the fact that the spatial distribution is highly skewed, so that ASes making use of a large number of IP/24 are over-represented. Particularly, while more than 50% of the ASes (75% of BGP announced prefixes) make use of a single anycast IP/24, about the 10% of the ASes

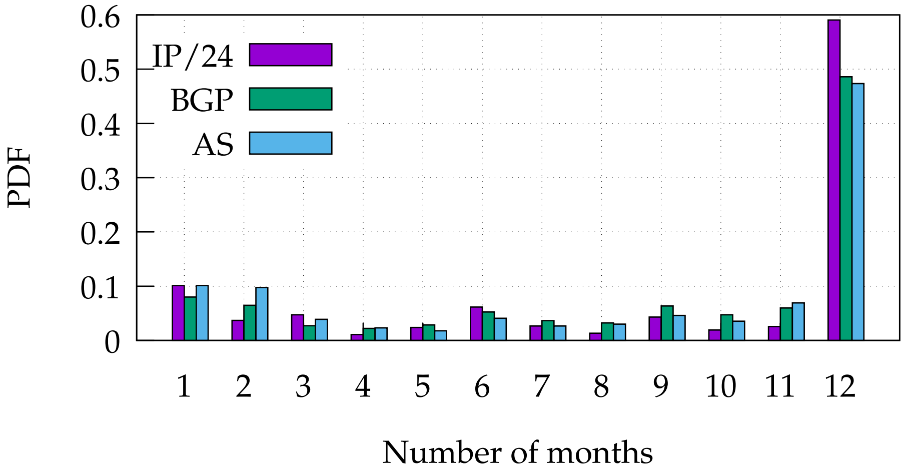


Figure 4.19: Anycast deployments stability over time: number of censuses where the IP/24, BGP prefix or AS is present over the one year observation period.

(BGP prefixes) hosts more than 10 anycast IP/24, topping to 384 (for 104.16.0.0/12) and 3016 (for AS13335). Since all three level of aggregation have relevance to give an unbiased picture of Internet anycast, we make available monthly snapshots with IP/24, BGP and AS aggregations as tabular data [12], which is also browsable online.

Finally, as a rough measure of persistence of individual anycast deployments, Fig.4.19 depicts a breakdown of the number of months that these catchments are present in our censuses at IP/24, BGP or AS levels. Notice that over 45% of anycast ASes (60% of anycast IP/24) consistently appear in our measurements for the whole year and 70% ASes (78% IP/24) appear at least 6 months. Only less than 10% deployments are seen only once.

4.3 Results: focused view

Top-10 deployments. We now provide a more detailed view of a few selected ASes out of the 566 in our censuses. Particularly, Tab. 4.2 reports detail concerning the top-10 deployments (company name and type, AS number and the number of BGP prefixes announced by that AS), the spatial footprint (i.e, the number S_A of IP/24 per AS and its temporal variability) and the geographical footprint (i.e, the number G_A of distinct replicas and its temporal variability). To compactly represent the size of a deployment, we report the maximum number S_A^+ of observed anycast IP/24 over the last year for that AS, as well as the $G_A^+ = \max G_A^{RIPE}(t)$ maximum number of locations observed from RIPE Atlas (recall that the number of locations is lower bound of the actual number of anycast replicas due to additive noise in the propagation latency measurements). The selection in Tab. 4.2 reports the top-5 in terms of S_A^+ spatial footprint (top) and the top-5 for G_A^+ geographic footprint (bottom).

Considering the spatial footprint IP/24, Cloudflare (AS13335) has a leading role: it is present in all the censuses with over 3 thousands IP/24 belonging to about 200 announced prefixes (mainly /20 but also less specific prefixes, as a /12 or a /17), and we did not observe significant variation over time. Furthermore, as confirmed from RIPE Atlas, the deployment has an heterogeneous geographical footprint, with some /24 having only 10–15 instances, while in the majority of the cases the /24 appear at over 40 distinct locations (119 according to [19]). Notice that, this would had been unnoticed if we had performed censuses at a different granularity (i.e., one IP/32 per BGP prefix as opposite as to one IP/32 per IP/24 in that prefix). Few other companies have over 100 anycast IP/24 prefixes in our censuses. For instance, Google (AS15169) exhibits a 3-fold increase in the number of IPs/24 in the last year (from 130 IPs/24 announced mainly by /16 in June 2016, to 330 IP/24 announced also by 190 new /13 prefixes in March 2017), the majority of which have instances at more than 30 locations. Opposite behaviors are also possible: for instance, the Fastly CDN (AS54113), shrunk its spatial footprint, the majority of which belong to an IP/16 and regularly appear in all our censuses. Interestingly, as early depicted in Fig. 4.17, the overall aggregate of all anycast deployments (i.e., the number of anycast /24 in the Internet and their geographical breakdown) is stable despite the variability of the individual deployments.

Deployment footprint:				Spatial		Geographical	
Company	AS	Type	BGP	S_A^+	CV_S	G_A^+	CV_G
Cloudflare	13335	CDN	206	3016	0.04	49	0.07
Google	15169	SP	16	524	0.38	30	0.08
Afilias	12041	TLD	218	218	0.15	6	0.10
Fastly	54113	CDN	34	175	0.09	20	0.07
Incapsula	19551	DDoS	146	146	0.23	15	0.17
Cloudflare	13335	CDN	206	3016	0.04	49	0.07
L root	20144	DNS	1	1	0	47	0.13
F root	3557	DNS	2	2	0	40	0.19
Woodynet	42	TLD	132	133	0.02	39	0.12
Verisign	26415	Reg.	2	2	0	36	0.20

Table 4.2: Focused view: Footprint variability of top-5 spatial (top) and top-5 geographical (bottom) deployments.

We recall that the common way to deploy anycast is to announce an IP prefix from multiple points using the same AS [87]. Another way is to announce the IP prefix using multiple ASes, usually referred as MOAS prefixes. In our dataset, we identified hundred IPs/24 as MOASes, that are commonly announced by few siblings, i.e., different ASes belonging to the same organization that announce the same prefix. However we spot cases where the number of ASes is greater than 10: for instance, we find that Verisign announces MOASes with 17 different ASes in the range of AS36617-AS36632; similarly, the Registry and DNS company AusRegistry, announces MOASes with 13 different ASes.

To compactly represent the spatial footprint evolution over time, we use the coefficient of variation, computed as the ratio of the standard deviation over the average number of anycast IP/24 per month $CV_S = \text{std}(S_A(t))/\mathbb{E}[S_A(t)]$, From Tab. 4.2, we can see that for deployments that have large spatial footprint (top), the variability CV_S can be important (e.g., Google or Incapsula), hinting to deployments that have grown (or shrunk) significantly. Conversely, among deployments with large geographical footprint, several have a very small spatial footprint ($S_A^+ \geq 2$) and exhibit no variation $CV_S = 0$.

Finally, as simple indicator of geographical footprint variability we compute $CV_G = \text{std}(G_A^{PL}(t))/\mathbb{E}[G_A^{PL}(t)]$ from PlanetLab measurements. Notably, we expect part of the variability to be due to measurement imprecision: e.g., shrinking number of VPs, packet losses and increased delay, can lead to underestimate the number of distinct locations. Yet, as it can be seen from Tab. 4.2, we find that the geographical variability is lower than the spatial one: this is reasonable since, while spatial variability hints to configuration changes in software, the geographical one possibly hints to physical deployments of new hardware.

Temporal variability. We now inspect temporal variability at a finer grain. We start by depicting in Fig.4.20 the temporal evolution of the spatial footprint, normalized over the maximum observed for that deployment (i.e., $S_A(t)/max_t S_A^+(t)$) for catchments in the top-5 (Afilias, Google) as well as for other key Internet players (Microsoft, Akamai, Netflix, Windstream). Evolutions represent a sample of what can be found in our censuses: for instances the two ASes reported in the picture owned by Akamai (AS21342) and Neflix (AS40027) start being announced as anycast during our observation period and either systematically (Akamai) or abruptly (Neflix) increase the amount of responsive anycast IP/24 over time. Google (AS15169) and Microsoft (AS8068) both have a sizeable presence at the beginning of the observation period, with roughly 50% of the IP/24 already in use, and roughly double the amount of IP/24 used at the end

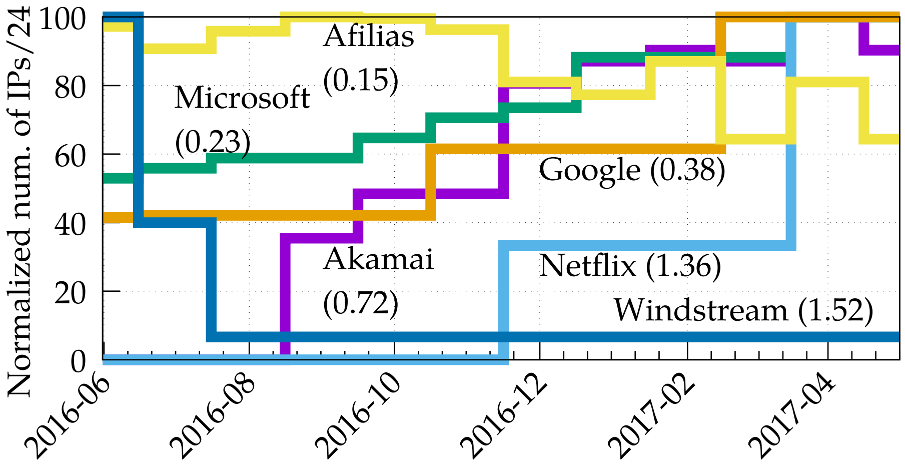


Figure 4.20: Spatial footprint evolution: Number of IPs/24 for selected anycast AS deployments (PlanetLab).

of the period in a smooth (Microsoft) or abrupt (Google) fashion. Finally, close to the beginning of our observation period, Windstream drastically reduces its anycast spatial footprint, keeping just a single anycasted IP/24. While these observations have anecdotal value, and cannot explain the reason behind changes in the deployment, they however confirm that anycast deployments have a rather lively temporal evolution, the extent of which is captured by the coefficient of variation. It is worth recalling that individual deployment exhibit wide variations, however the aggregate remains quite stable over time (recall Fig.4.17).

It is intuitive that the number (and location) of vantage points upper-bounds (and constrains) the number of anycast instances that can be found. Given the slow but steady decrease of the PlanetLab VPs, we unfortunately do not deem PlanetLab measurement reliable in assessing, at a fine grain, the geographic growth (which can be underestimated) or reduction (which can be due to VP decrease) of anycast deployments. We thus decided to regularly monitor anycast prefixes using 500 Ripe Atlas VPs. We picked targets from two key CDN players, namely Cloudflare (8 different

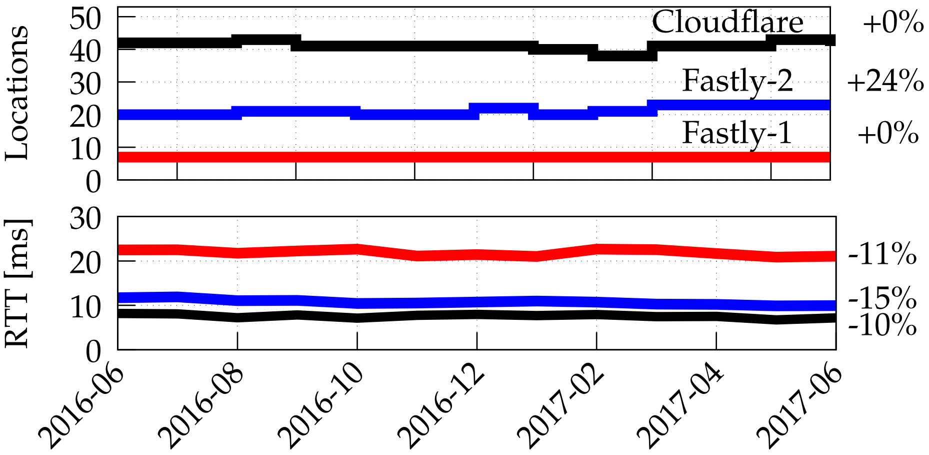


Figure 4.21: Geographical evolution of selected anycast deployments: Number of locations (top) and delay toward the replicas (bottom) measured from RIPE Atlas.

IP/20) and Fastly (5 different IP/24). As per Tab. 4.2, Cloudflare is the top-1 player over all anycast, and given its sheer footprint, we expect it to grow at a slower rate with respect to other deployments: especially, the number of locations appears to already significantly exceed the one[3] suggested in [62]. As such, we use Cloudflare as a litmus paper for our measurement. Our selection of Fastly is then motivated by the fact that despite it appears in the top-5 and so we expect it to steadily appear in our measurement, it has 1/10 the spatial footprint and 1/2 the geographic footprint of Cloudflare: so it not only has room to grow, but also possibly has the money necessary for the investment.

Fig. 4.21 comprises three lines: for Cloudflare, we report the average over all IP/20, whereas for Fastly we cherry pick 2 out of the 5 IP/24 we monitored, that are representative of the typical (Fastly-1) and smaller-than-typical Fastly deployment

[3] In particular [62] states that “After carefully studying four very different anycast deployments, we claim that 12 anycast sites would be enough for good overall latency. Two sites per continent, in well chosen and well connected locations, can provide good latency to most users.”. The Cloudflare catchment exceeds this suggestion by approximatively 5× in our measurement, and by nearly an order of magnitude according to to the Cloudflare blog [17, 19] that reports 119 nodes at times of writing.

(Fastly-2). In the case of Cloudflare the figure shows that, as expected, the number of instances is stable (i.e., the growth rate is slow with respect to our observation window), so that fluctuations are only measurement artifacts. Whereas the Fastly-1 IP/24 remains stable IP/24 with only 7 different locations, in the case of the Fastly-2 IP/24, we observe a growth from 19 locations in June 2016 to 24 in June 2017. For reference purposes, 33 locations are mentioned in [20]: this corresponds to a 73% recall, in line with expectations for the iGreedy methodology [57]. Latency measurements are shown in the bottom part of the figure. We can observe a 10% of latency reduction also for stable deployments (Fastly-1 and Cloudflare) and it thus to be imputed to other causes (e.g., increased peering connectivity). At the same time, at least for Fastly, it appears that increasing the number of instances reduces the average latency toward our RIPE Atlas probes by an additional 5%, and *halves* the 95th percentile (from 137ms in June 2016 to 68ms in June 2017), unlike expectations [62]. A significant take-away from Fig. 4.21 is that the advantage to increase the catchment size appears to have diminishing returns: in other words, the (delay) gain from Fastly-1 to Fastly-2 is significant, whereas the (delay) gain to expand further to reach the size of Cloudflare deployment is modest. Clearly, the fact that L and F DNS root servers and Cloudflare [17] deployments significantly exceed 100 distinct locations, implies that one size may not fit all for anycast deployments – and that further research is needed to provide a more accurate answer so as to what should be a reasonable size for anycast catchments.

5 Summary

Internet anycast is an important building block of the current Internet: the study of deployments and their evolution is useful to enrich our understanding of Internet operations.

We present a spatial and longitudinal study of IPv4 anycast deployments through numerous censuses, gathered with an original and robust technique implemented with an efficient and scalable system design. Alongside sharing the knowledge gathered in this study, we especially believe that by making our datasets and tools available [12] to the scientific community, we can contribute to enrich the Internet map along the anycast dimension.

In the spatial study, 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, we find that the long-standing myth of anycast being used only for stateless services over UDP, and DNS in particular, no longer holds. Especially, our portscan campaign of anycasted subnets reveals over 450 well-known services from over 10^4 unique open ports. Additionally, we uncover 30 software implementations, with a relative breakdown that differs from software ranking in the unicast IP world.

In the longitudinal study, we learn that anycast detection (important for censorship studies [113]) is reliable in spite of varying (and especially diminishing) vantage points. We additionally see that anycast *spatial footprint* (i.e., the number of anycast /24 per AS) evolves significantly for individual deployments, though it remains steady in the aggregate. PlanetLab censuses can reliably measure this variability.

However, while we gather that anycast *geographical footprint* evolves, we also acknowledge that to accurately track the state of anycast Internet at replica level a large set of vantage points are needed. In this case, due to decreasing PlanetLab VPs, a more tight coupling with RIPE Atlas would be needed (e.g., monthly detection from PlanetLab, followed from a refinement of geolocation for detected anycast deployments).

Finally, by closely monitoring a few deployments with RIPE Atlas, we gather that even anycast deployments that already have a large geographical footprint, apparently benefit (in that their service latency decreases, though with diminishing returns) from further growing the deployment beyond sound rules of thumb [62], which requires more systematic investigation. In particular, IP anycast is an appealing way to implement, at relatively low cost, an effective replication scheme for a variety of services [78], as the paths leading to anycasted replicas are (with few exceptions) significantly stable over time [129]. Given anycast importance, a broad and systematic analysis of the current catchments is hopefully helpful to update and distill deployment guidelines along the lines of [17, 62, 106].

Chapter 5

IPV4 Anycast Traffic Characterization from Passive Traces

In the previous chapters, we provide a broad characterization of anycast deployments. However, the active studies presented are not able to provide some information such as how popular anycast services are, how much traffic they attract, and which applications they serve that can be only gathered via passive measurement.

In this Chapter, we perform a passive characterization study and focus on Internet key players that have started adopting this technology to serve web content via Anycast-enabled CDNs (A-CDN). To the best of our knowledge, in the literature, there are studies that focus on a specific A-CDN deployment, but little is known about the users and the services that A-CDNs are serving in the Internet at large. Hence, using the IP anycast list found with our census in March 2015, we leverage a passive study and provide a first characterization of modern usage of A-CDNs.

In the remainder of this chapter, we start investigate the properties of IP anycast traffic (Sec. 1), then we focus on the stability of routes towards the anycast servers (Sec. 3.3) and finally, we conclude with a discussion of open issues (Sec. 4).

We summarize our main findings as follows:

- We confirm IP anycast to be not anymore relegated to the support of connectionless services: in a month we observe over 16,000 active anycast servers contacted via TCP, mapping to more than 92,000 hostnames.

- While hard to gauge via passive measurement, and despite the limited scope that our single vantage point offers, we in general observe stable paths, with only a handful changes during one month.

- Both large players like Edgecast or Cloudflare, and smaller but specialized A-CDNs are present: content served include heterogeneous services such as Twitter Vine, Wordpress blogs, TLS certificate validation and BitTorrent trackers. Footprint of A-CDN is also very different, with some being pervasive enough to have servers at few ms from customers, while others have fewer replica nodes that turn out to be more than 100ms far away.

- In our datasets, A-CDNs are fairly popular: 50% of users encounter at least an A-CDN server during normal web activity. Thus, penetration of A-CDN is already very relevant.

- Most of TCP connections last few tens of seconds and carry a relatively small amount of bytes; surprisingly however, we see video and audio streaming services being supported by A-CDNs, whose TCP flow last for several hours. The latter could be affected by sudden routing changes that could break TCP connections. However, given the infrequent occurrence of such events, it is hard to measure (and suffer from) it in practice.

1 Methodology Overview

1.1 Anycast subnet lists

In a nutshell, our workflow first extract the subset of flows that are directed to anycast servers, and then characterize the traffic they exchange with actual internet users by leveraging passive measurements. To identify anycast servers, we rely on the *exhaustive* list of /24 subnet prefixes that result to host at least one anycast server according to the IP censuses we presented in the previous Chap. 4 in March 2015.[1] As described in Chap. 4, we compile an *exhaustive* list of 1696 anycast subnets, by simultaneously pinging an IP/32 from all valid IP/24 networks from PlanetLab probes. The scan runs on all IP address range. Next, we ran the anycast detection technique developed in in Chap. 3 to identify IP/32 anycast addresses (i.e., located in more than one geographical location). We flag as anycast network any network having at least two locations in our censuses. From this list, we extract a more *conservative* set of 897 subnets, having at least five distinct locations. Since the conservative set is biased toward larger and likely more popular deployments, we expect the conservative set to yield an incomplete but comprehensive picture of IP anycast. In the following, we use this list to inform the passive monitor about the subnets of interest.

1.2 Passive monitor

We instrumented a passive probe at one PoP of an operational network in an European country-wide ISP.[2] The probe runs Tstat [74], a passive monitoring tool that observes packets flowing on the links connecting the PoP to the ISP backbone network. The probe uses professional Endace cards to guarantee all packets are efficiently exposed in

[1] Recall that BGP announced prefixes have a minimum granularity of a /24 subnet. Thus, we do not expect an anycast subnet smaller than a /24.

[2] Results from other vantage points in other PoPs are practically identical. For easy of presentation, we focus on one PoP.

user-space for processing. No sampling is introduced, and the probe is able to process all packets [126]. Tstat rebuilds in real time each TCP and UDP flow in real time, tracks it, and, when the flow is torn down or after an idle timer, it logs more than 100 statistics in a simple text file. For instance, Tstat logs the client and server IP addresses[3], the application (L7) protocol type, the amount of bytes and packets sent and received, the TCP Round Trip Time (RTT), etc.

Tstat implements DN-Hunter [45], a plugin that annotates each TCP flow with the server Fully Qualified Domain Name (FQDN) the client resolved via previous DNS queries. For instance, assume a client would like to access to *www.acme.com*. It first resolves the hostname into IP address(es) via DNS, getting 123.1.2.3. DN-Hunter caches this information. Then, when later the same client opens a TCP connections to 123.1.2.3, DN-Hunter returns *www.acme.com* from its cache and associate it to the flow. Our vantage points observe all traffic generated by clients, including DNS traffic directed to local resolvers. Client DNS cache is rebuild in Tstat, resulting in more than 95% accuracy [45]. This is particularly useful for unveiling *services* accessed from simple TCP logs. This is useful since it unveils the service being offered by the server having IP address 123.1.2.3, even in presence of encrypted (e.g., HTTPS) or proprietary protocols[4].

For this study we leverage a dataset collected during the whole month of March 2015. It consists of 2 billions of TCP flows being monitored, for a total of 270 TB of network traffic. 1.4 billion connections are due to web (HTTP or HTTPS) generating

[3] We take care of obfuscating any privacy sensitive information in the logs. Customer IP addresses are anonymised using irreversible hashing functions, and only aggregate information is considered. The deployment and the information collected for this have been approved by the ISP security and ethic boards.

[4] Collisions may be present, e.g., when the same client contacts *mail.acme.com* which is hosted by the same server 123.1.2.3. Since in this work we are interested on which services a given server hosts, collisions are not critical, e.g., we can discover that 123.1.2.3 serves both *www.acme.com* and *mail.acme.com*.

209 TB of data. More important, we observe more than 20,000 ISP customers active over the month, which we identify via the static anonymized client IP address[5]. All traffic generated by any device that accesses the internet via the home gateway is thus labeled by the same client IP address. This includes PCs, smartphone, Tablets, connected TV, etc. that are connected via WiFi or Ethernet LAN to the home gateway.

Among the many measurements provided by Tstat for each TCP flow, we only focus on: (i) the server IP address; and (ii) the anonymized client IP address; (iii) The minimum Round-Trip-Time (RTT) between the Tstat probe and the server; (iv) The amount of downloaded bytes; (v) The application layer protocol (e.g., HTTP, HTTPS, etc.); (vi) The FQDN of the server the client is contacting as returned by DN-Hunter. These metrics are straightforward to monitor, and details can be found in [45, 74].

1.3 Temporal properties

We first provide an overall characterization of the anycast traffic. Not surprisingly, we observe that all anycast UDP traffic is labeled as DNS protocol – which we avoid investigating given the literature on anycast DNS. More interestingly, we observe a sizeable amount of anycast traffic carried over TCP: overall, almost 59 million TCP connections are managed by anycast servers. Those correspond to approximately 3% of all web connections and 4% of the aggregate HTTP and HTTPS volume, for a total of 6 TB of data in the entire month. Definitively a not-negligible amount of traffic, especially when considering the relatively small number of /24 anycast subnets.

The large majority of traffic is directed to TCP port 80 or 443, that the DPI classifier labels as HTTP and SSL/TLS, respectively. This suggests that hosted services are indeed offered by A-CDNs and served over HTTP/HTTPS. A minority of the traffic (less than 1% of all anycast traffic) is instead related to some protocols for multimedia

[5] The ISP adopts a static addresses allocation policy, so that each customer home gateway is uniquely assigned the same static IP address.

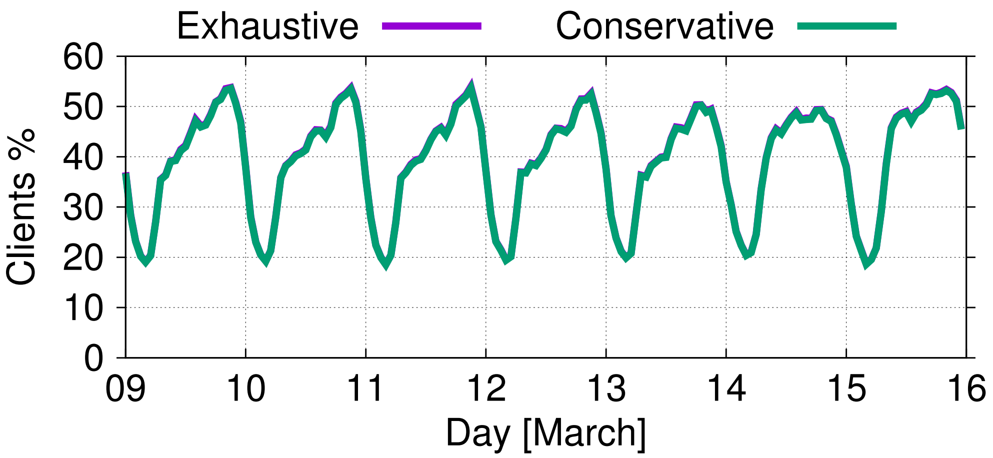


Figure 5.1: Percentage of clients that contact at least one A-CDN server in each 1h time slot, both curves overlap.

streaming, email protocols, Peer-to-Peer traffic, or DNS over TCP. We will provide further details when digging into some selected examples.

2 Results: broad view

Results confirm the footprint of anycast traffic, and A-CDN in particular. To corroborate this, Fig. 5.1 shows evolution during one-week of the percentage of active customers that have encountered at least one anycast server during their normal web browsing activities (the ratio is computed at hourly intervals, normalizing over the number of client active in that hour). Besides exhibiting the day/night pattern due to the different nature of services running on the network with fewer services served over anycast at night, the figure shows that at peak time the probability to contact at least one anycast server is higher than 50%. Notice that Fig. 5.1 reports the probabilities according to both the exhaustive and the conservative lists: these curves cannot be distinguished as they perfectly overlap, which hints to the fact that the conservative list is, as expected, comprehensive enough for our purposes. This clearly may change on vantage points

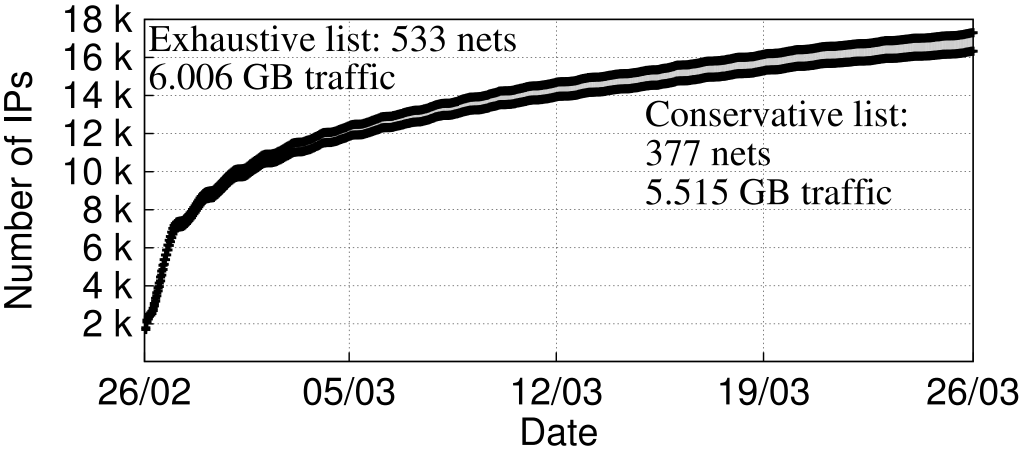


Figure 5.2: Cumulative number of distinct servers encountered over the month.

located in different countries. However, for the purpose of this work, we prefer to take a conservative approach.

Fig. 5.2 reports the cumulative number of unique IP anycast addresses observed over time, again for both the conservative and exhaustive lists. In total, over 16,000 distinct IP addresses are observed during the whole month for the conservative list. The picture is additionally annotated with the traffic volume exchanged with such servers: notice that despite the exhaustive list is twice as big as the conservative one, the number of servers contacted and bytes exchanged are fairly similar. This happens since the major A-CDN players are present in the conservative list, to which we thus limitedly focus on the following. Notice also that the number of distinct servers encountered over the month quickly grows during the first days, during which most popular services and servers are contacted.

2.1 Service diversity

We now provide an overall picture of A-CDN diversity with a dual violin plots (Fig. 5.3) and parallel coordinate (Fig. 5.4). Violin plots compactly represent the marginals for some metrics of interest, whereas parallel coordinate plots allow us to grasp the

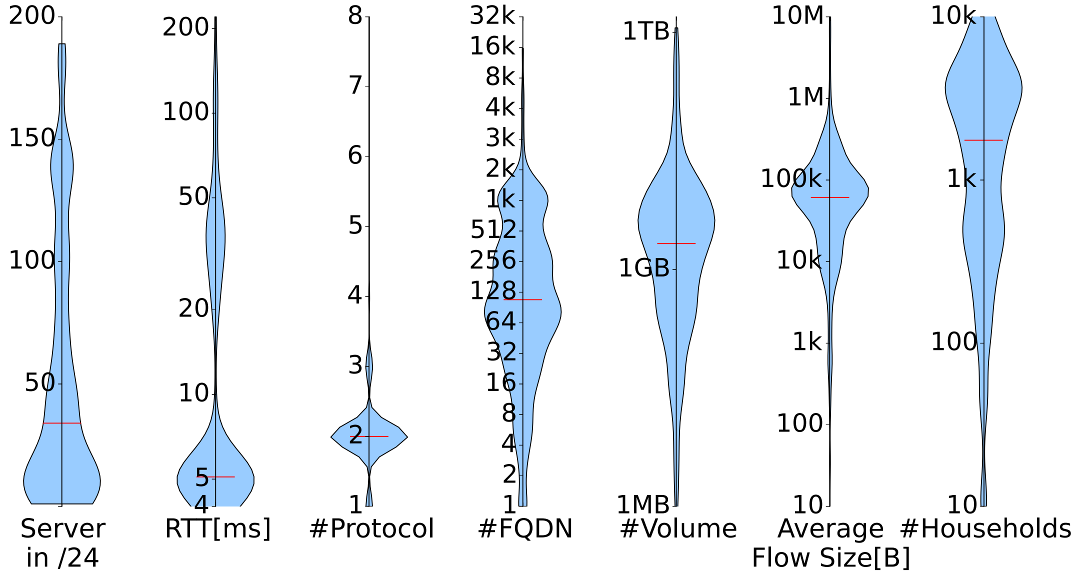


Figure 5.3: Anycast at a glance: violin plot

correlation between these metrics for some specific deployments. On both plots, we select the following axes:

- the number of active servers in the /24 subnet;
- the average minimum RTT for any server in that /24; (
- the number of distinct protocols;
- the number of distinct FQDNs/services;
- the total amount of bytes served during the whole period;
- the average flow size in bytes;
- the number of households that contacted one server in the /24;

In more details, violin plots of Fig. 5.3 are an intuitive representation of the Probability Density Function (PDF): the larger their waist is, the higher is the probability of

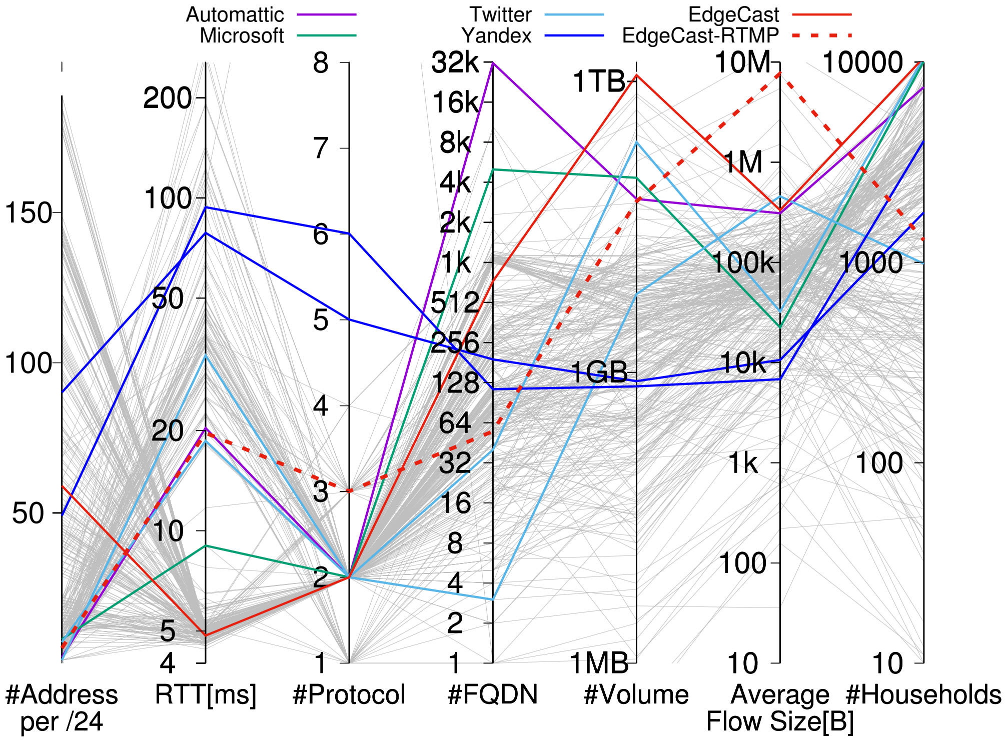


Figure 5.4: Anycast at a glance: parallel coordinate plot

observing that value; red bars are a further visual reference, corresponding to the median of the distribution. Overall, the plot shows that most of /24 host few servers, which are in general quite close to the PoP (RTT<10 ms), use 2 or at most 3 protocols (HTTP or HTTPS mostly). Diversity starts to appear in the number of served FQDNs – with some /24 being used for a handful services, while others serve several thousands. Served volume varies widely. Similarly, while only half of flows exceed 50 kB, and most are below 1 MB, flow size peaks up to several hundreds MB (see Sec.1 for more details). Considering popularity, some /24 are used by several thousands end-users, others by less than 10.

Parallel coordinates instead allow the observation of a specific deployment: each line represent a /24 subnet, and the "path" among the vertical axes highlights the

Table 5.1: Dataset summary

/24 subnet	Owner	IP/32	Vol. [GB]	Flows [k]	Users	FQDN	Protocols	Content/Service
93.184.220.0	EdgeCast-1	105	357	8,018	10,626	3,611	HTTP/s	generic
199.96.57.0	**Twitter-generic**	7	219	7,318	10,508	40	HTTP/s	twitter, vine
68.232.34.0	**EdgeCast-2**	59	1,071	3,484	10,490	736	HTTP/s	microsoft, spotify
68.232.35.0	EdgeCast-3	104	480	5,059	10,354	904	HTTP/s	twitter, gravatar, tumblr
94.31.29.0	NetDNA	73	80	1,292	10,218	609	HTTP/s	generic
93.184.221.0	EdgeCast-4	49	708	2,031	10,155	1,467	HTTP/s	generic
204.79.197.0	**Microsoft**	8	93	4,508	10,044	5,088	HTTP/s	bing, live, microsoft
205.185.216.0	Highwinds-1	2	180	1,411	9,705	267	HTTP/s	generic
108.162.232.0	CloudFlare-1	13	53	550	9,274	14	HTTP	ocsp certificates
178.255.83.0	Comodo	5	3	601	8,837	65	HTTP	ocsp certificates
192.0.72.0	**Automattic**	2	57	199	7,477	32,037	HTTP/s	wordpress
108.162.206.0	CloudFlare-2	122	14	246	4,695	465	HTTP/s, Torrent	P2P trackers, generic
213.180.193.0	**Yandex-2**	49	0.7	105	4,031	114	HTTP/s SMTP	yandex
213.180.204.0	**Yandex-1**	90	0.7	76	1,771	191	HTTP/s, SMTP	yandex
93.184.222.0	**EdgeCast-RTMP**	5	53	8	1,289	55	RTMP, HTTP	soundcloud, video
199.96.60.0	**Twitter-vine**	1	6	14	983	3	HTTP/s	vine
Total	Exhaustive	17,298	6,006	58,885	10,830	120,151	-	-
Total	Conservative	16,329	5,515	54,045	10,828	117,768	-	-

characteristic of that A-CDN over different dimensions. We report most[6] /24 with light gray color, and additionally we highlight some of them using different colors. First, observe that the wide dispersion of the light gray paths testifies great diversity across A-CDN deployments.

Next, observe per-deployment dispersion of the few selected /24. For the sake of illustration, consider the Automattic curve, which is the A-CDN that serves websites hosted by Wordpress: it can be seen that the two active servers found in the /24 are located at about 20 ms from the PoP. They use both HTTP and HTTPS, for a total of more than 32,000 FQDNs. Total volume accounts for 20 GB during the month, transferred over flows that are 200 kB long on average. At last 7400 users (37%) accesses some

[6] To reduce visual cluttering, we report in light gray color the subset of /24 that served at least 1000 flows and 10 distinct households during a month.

content hosted by Automattic. We have highlighted some examples such as: Twitter A-CDN (which serve few domains); Microsoft A-CDN (bing.com and live.com services, for a total of more than 5000 FQDNs); one /24 of EdgeCast as an example of a generic A-CDN; a specialized EdgeCast platform serving audio and video streams over Real Time Media Protocol (RTMP), see the red dashed line. Finally, we selected two /24 belonging to Yandex, the most popular search engine and social platform in Russia, that are however not among the most popular in the geographic region of our vantage point. It clearly appears that these examples, which we more deeply investigate in the next section, are representative of quite diverse anycast deployments. This makes it difficult to highlight common trends, e.g., we observe popular A-CDN whose RTT is quite large, and unpopular A-CDN whose RTT is instead much smaller.

3 Results: focused view

3.1 Candidate selection

Tab. 5.1 offers details for some selected deployments. Rows highlighted in bold font refers to the same subnets previously highlighted in Fig. 5.4.

For each /24 subnet, Tab. 5.1 lists the Owner (i.e., the organization managing it as returned by Whois), the number of distinct server addresses that have been contacted at least once, the total volume of bytes served, the number of flows, of users, and of distinct FQDNs. The last two columns report the most prominent protocols and services the A-CDN offers.

The table, which also serves as a summary of our anycast dataset, comprises the top-10 most popular /24 A-CDN prefixes and subnets (top part). To avoid an excessive bias toward only popular services (where HTTP and HTTPS are predominant), we additionally report some manually selected A-CDNs (bottom part).

As it can be observed, the portfolio of services supported by A-CDNs includes email via SMTP, video/audio streaming via RTMP, certificate validation via Online Certificate Status Protocol (OCSP), and even BitTorrent Trackers.

The table precisely quantifies the very heterogeneous scenario early depicted by the Fig. 5.3. EdgeCast is the major player in our dataset, managing 4 of the top-10 subnets: each of these serves between 350 GB/month and 1 TB/month to more than 10,000 (50%) households in our PoP. This is not surprising, given EdgeCast claims to serve over 4% of the global Internet traffic[7].

Popular A-CDNs includes Microsoft, which directly manages its own A-CDN. It serves Bing, Live, MSN, and other Microsoft.com services. Since it handles quite a small amount of data and flows, we checked if there are other servers not belonging to the Microsoft A-CDN that handle those popular Microsoft service. We found that all of *bing.com* pages and web searches are actually served by the Microsoft A-CDN, while static content such as pictures or map tiles are instead by the Akamai CDN. Thus, Microsoft is using an hybrid solution based on a traditional CDN and its own A-CDN at the same time.

Finally, popular A-CDN services include Highwinds and Comodo. Highwinds offers video streaming for advertisement companies, and images for popular adult content websites (notice the relative longer lived content). Instead, Comodo focuses its business on serving certificate validations using OCSP, Online Certificate Status Protocol, services (with lot of customers who fetch little information).

Overall, major A-CDN players serve thousands of FQDNs including very popular web services like Wordpress, Twitter, Gravatar, Tumblr, Tripadvisor, Spotify, etc. This explains why about 1 out of 2 end-users likely contacts at least one A-CDN server during their navigation. Most FQDNs are uniquely resolved to one IP address – but the same IP address serves multiple FQDNs. Interestingly, this behaviour is shared among most of the studied A-CDNs, meaning that they purely rely on anycast routing for load-

[7] `http://www.edgecast.com`

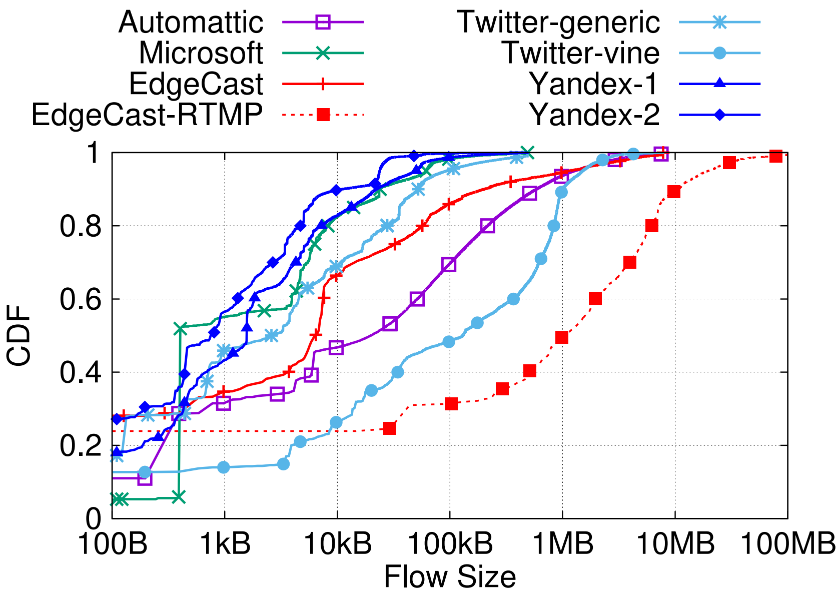


Figure 5.5: Metrics characterization: Flow size CDF

balancing. An exception is CloudFlare's A-CDN which uses also DNS load-balancing. Cloudflare uses up to 8 IP addresses in the same /24 to serve the same FQDN.

3.2 Per-deployment view

For the selected deployments, we details the cumulative distribution function (CDF) of some interesting metrics. The CDF is computed considering all the flows being served by the same /24 subnet.

As for the metrics of interest, we report the CDF of flow size (Fig. 5.5), duration (Fig. 5.6), and Round Trip Time (Fig. 5.7). Fig. 5.5 shows how the size of the content hosted by A-CDNs varies across deployments (the different supports only partly overlap), and also between flows of the same deployment (the support is large and, with few exceptions, there is no typical object size). In general served objects are shorter than 1 MB, with the notable exception of audio and video streams served by EdgeCast specialized deployment that support RTMP streaming. In this case, flows are

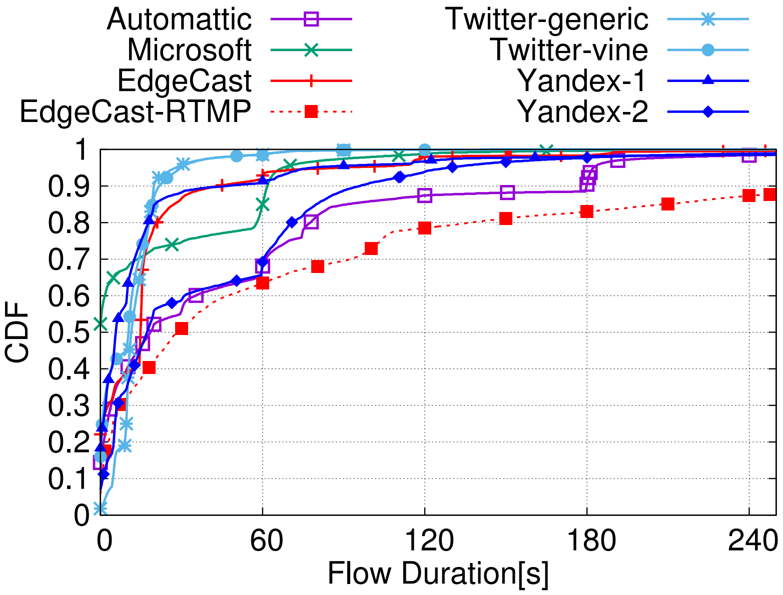


Figure 5.6: Metrics characterization: Flow duration CDF

larger than 100 MB.

The small amount of data is reflected on the TCP flow duration CDF. Indeed, Fig.5.6 shows that flow lifetime is in general shorter than 180 s, with specific values that reflects typical HTTP server timeouts (multiple of 60 s). Once again, the only exception is the EdgeCast-RTMP deployment, for which over 10% of the TCP flows exceed 5 minutes (visible in the picture), and ranges up to hours (not visible in the picture). Finally, minimum Round Trip Time CDF in Fig.5.7 reveals that the popular A-CDNs have a good footprint (at least in Europe). The only exception is Yandex that have anycast replicas in the eastern Europe and in Russia (which is not surprising due to the language specific content it serves).

Notice how sharp the CDF are for EdgeCast or Microsoft deployment. This suggests that the path to their servers is very short, but also very stable. RTT of other A-CDNs is instead very similar, e.g., EdgeCast-RTMP, Twitter-generic, and Automattic. This suggests that their servers are located in the same place, and reached by the same path. Interestingly, the minimum RTT shows a deviation which suggests the presence of

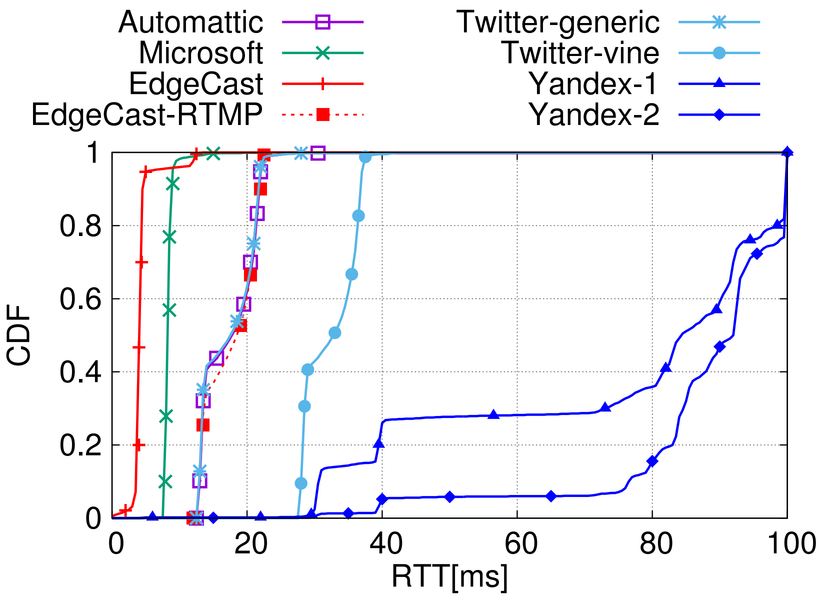


Figure 5.7: Metrics characterization: Round-trip time CDF

some extra delay for 60% of samples, possibly accounting for some queuing delay due a possibly congested link on that path (which belongs to the Twitter-vine path as well).

3.3 RTT variation over time

In this section we look for evidences may suggest possible path properties changes. We based our analysis studying the TCP minimum RTT. Intuitively, a *sudden* change in the minimum RTT could highlight a possible sudden change in the routing or network infrastructure. We argue that a dramatic change in the TCP minimum RTT, is likely due to a routing change. Conversely, a *smooth* shift could suggest the presence of queuing delay due to possible congestion on some path links. Passive measurements could only suggest to investigate more deeply of eventual changes, e.g., triggering active measurements to provide a more reliable ground truth to distinguish between hardware improvement and routing changes. For instance, checking HTTP headers could be used to reliably reveal the anycast replicas.

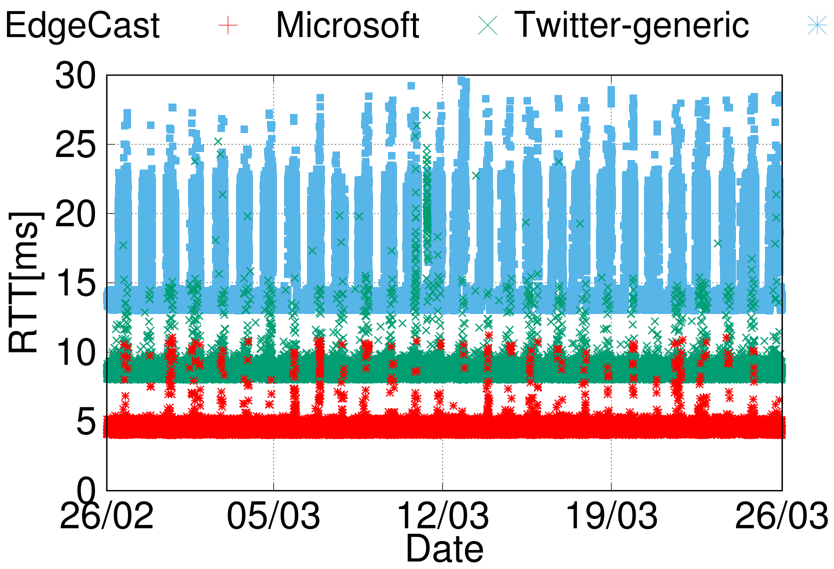
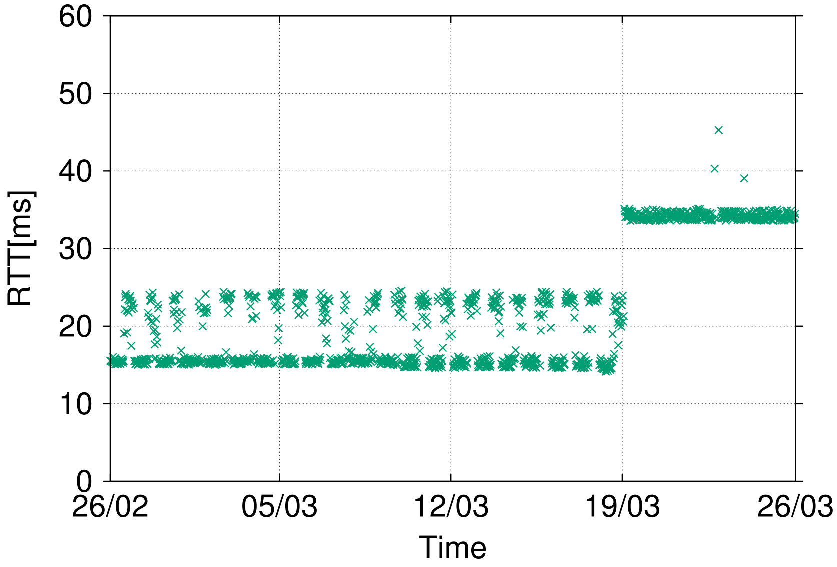


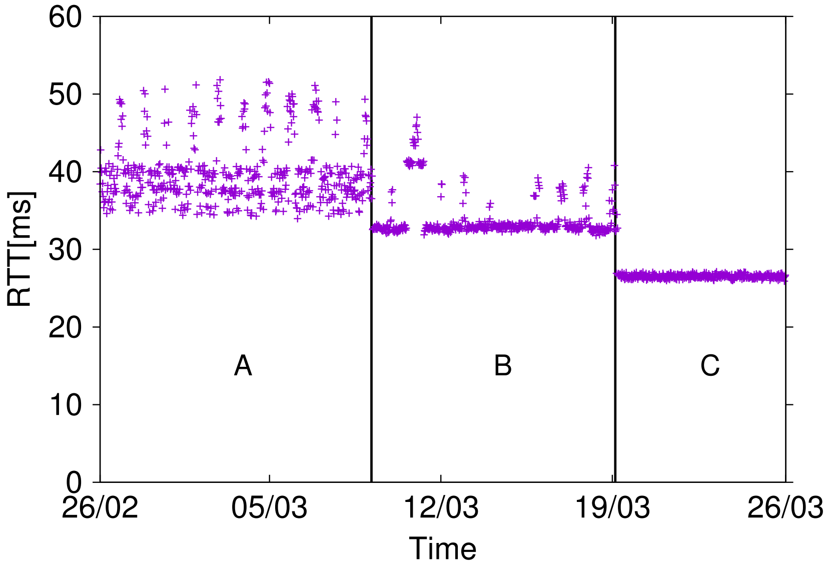
Figure 5.8: Stable Situation. Single curve plots can be found at [55]

Fig. 5.8 shows the minimum RTT values for each TCP flows over time. Three most used A-CDNs during the entire month of March 2015 are considered. The figure suggests that no sudden changes in the path are visible. Yet, observe the periodic and smooth increase of RTT during the peak time. This could be explained as a congestion events, that are reflected in the smooth changes in the minimum RTT CDF as shown in Fig. 5.7. For these three A-CDN, data suggest stable but possibly congested paths to the anycast addresses.

We investigated the RTT evolution over time of other /24 subnets. We found few cases that we believe could highlights possible sudden changes in the routing plane, possibly affecting server affinity. Fig. 5.9(a) and Fig. 5.9(b) report two examples. Plot on the top shows an example of sudden changes affecting another /24 subnet. In this case, the minimum RTT suddenly increases by almost 15ms. Notice also that the path to the longer location is not affected by periodical increases during peak time, suggesting no congestion is present in this second path. Look now at the plot on the bottom. It suggests changes on March 9th, and 19th (with possibly a short change on the 11th).



(a) Event 1: sudden change



(b) Event 2: multiple changes

Figure 5.9: RTT changes

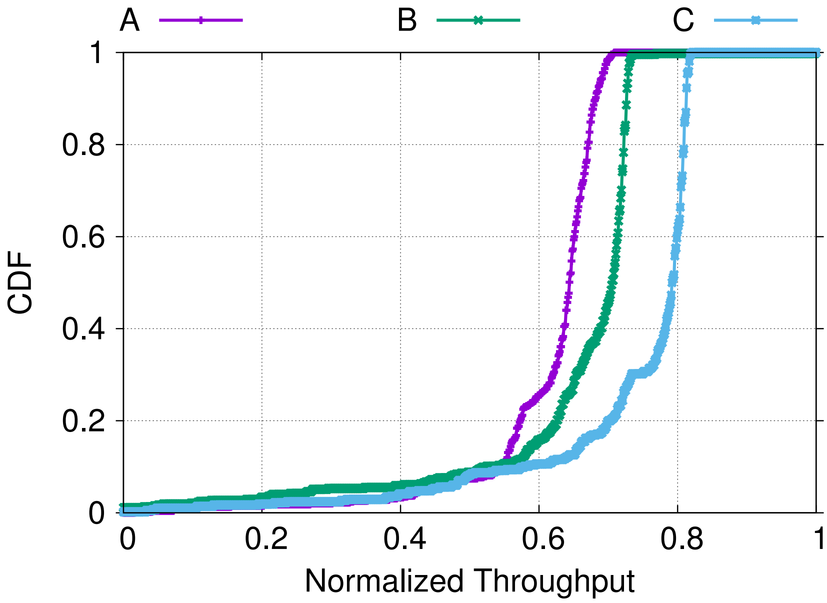


Figure 5.10: Event 2: Throughput Implications

Indeed, minimum RTT properties differ quite significantly and with a sharp change. To study the implication of this from a client perspective, we report the CDF of the throughput for the three distinct periods in Fig. 5.10. We normalize the throughput between 0 and 1 for ISP privacy motivation. The three distribution show that when the RTT decreases, i.e., the server serving the flow is closed to the users, the throughput improves. Thus, A-CDN changes have an impact on performance as well.

We also tried to investigate if changes have implications on TCP connections. In particular, one would expect that an on-going TCP connections to be abruptly terminated if the routing change implies a server change as well. We tried to investigate this by correlating number of TCP flows abruptly terminated by a server RST message with possible routing change events. We are not able to observe any clear evidence. Indeed, on the one hand, TCP flows are very short – cfr. Fig. 5.6 – and, on the other hand, changes are sudden and very few. Thus only an handful TCP connections could possibly be involved during a change event. This support the intuition that anycast is indeed well suited for connection oriented services.

In summary, while we observe sharp changes in the anycast path to reach the A-CDN caches, those events are few and occasional, with each different routing configuration that lasts for days. Clearly, deeper investigation is needed to better understand eventual routing changes over the time. In this direction we are trying to exploit other metrics as the Time To Live (TTL) and the Time To First Byte (TTFB) to highlight routing changes. Combining this methodology with active measurements, it could provide a better understanding of routing stability that may affect A-CDN deployments.

4 Summary

We presented in this Chapter a first characterisation of Anycast-enabled CDN. Starting from a census of anycast subnets, we analysed passive measurements collected from an actual network to observe the usage and the stability of the service offered by A-CDNs. Our finding unveil that A-CDNs are a reality, with several players adopting anycast for load balancing, and with users that access service they offer on a daily basis. Interestingly, passive measurements reveal anycast to be very stable, with stable paths and cache affinity properties. In summary, anycast is increasingly used, A-CDNs are prosperous and technically viable.

Chapter 6

Temporal evolution of IPV4 Anycast

Chapter 7

Conclusion

1 Summary of our contribution

The Internet works thanks to an ecosystem of companies, organizations, and communities. It is an ever-evolving environment used for new things each day– no longer what it once was in the 90's.

These days, users access a wide variety of content (e.g., software, video, and pictures) through the Internet and expect a certain quality of experience. To meet these demands, the organizations involved use forefront technologies to implement different solutions. One example is the duplication of content at various points of the globe for later retrieval. This mechanism helps reduce the traffic of the network and keeps the users satisfied.

This thesis raises awareness and sheds light on another one of these mechanisms: anycast. In most of the anycast deployments, contents are replicated over multiple servers. This holds many advantages e.g. improving reliability and solving scalability issues. Some companies use anycast as a first layer of defense against Distributed-denial-of-service (DDoS) attacks.

Clearly, IP-anycast is not a cure-all: first, it requires financial investment (depending on the number of anycast servers). Second, the contents need to be consistent. Finally, in case the company wants offer stateful services over anycast, it has to do a significant amount of engineering to avoid flapping between different anycast servers.

The first contribution (Chap. 3) of this thesis is the design of a methodology. It is able to discover, enumerate and geolocate IP-anycast instances. We release its implementation as open-source software, iGreedy.

The second contribution (Chap. 4) is the design and implementation of a system, which uncovers the IP-anycast players through various Internet censuses. Their evolution is presented with a longitudinal study. Monthly updates are published through a list of anycast IPs, including their footprint and geolocation. Additionally, the results are integrated in the RIPE Atlas project, OpenIPmap [31].

Finally, a passive study (Chap. 5) was conducted to observe both the usage and the stability of anycast CDNs, the results of which demonstrate both the popularity and reliability of anycast services.

2 Future work

Future research directions can be summarised as follows.

Troubleshooting. iGreedy could be useful in general, adding e.g., a relevant feature for troubleshooting [132], including e.g., ensuring reachability of specific anycast replicas, or detecting unexpected affinity between a specific replica and (a faraway) vantage point. Additionally, some inference techniques can be applied only on the unicast context [46, 68], where authors generally have to resort to some heuristic to discard suspiciously anycasted instances: in this context, iGreedy could either automatically validate the assumption, or raise a flag forbidding to use such unicast-only techniques in case of positive detection.

BGP hijack detection. As IP-level anycast is realized through announcement of the same BGP prefix from multiple points, the iGreedy technique could be used to assist BGP hijacking detection – as anycast and hijacks are indeed "syntactically" equivalent with respect to a router speaking the BGP "lingo". Despite a large literature on the topic [35, 88], to the best of our knowledge most work exploits features related to AS-origins, AS-paths and prefix announced whereas latency information such as the one we propose here has so far been ignored. Technique such as iGreedy could be run reactively, detecting geo-inconsistencies for knowingly unicast prefixes and gaining in timeliness with respect to executing a full traceroute.

Horizontal comparison with IP unicast. Albeit very challenging, more efforts should be dedicated to compare Unicast vs Anycast CDNs for modern web services. To the very least, a statistical characterization and comparison of the pervasiveness of the deployments (e.g., in term of RTT) and its impact on objective measures (e.g., time to the first byte, average throughput, etc.) could be attempted.

Vertical investigation of CDN strategies. From our initial investigation, we noticed radically different strategies, with e.g., hybrid DNS resolution of few anycast IP addresses, use of many DNS names mapping to few anycast IPs, use of few names mapping to more than one anycast IPs, etc. Gathering a more thorough understanding of load balancing in these new settings is a stimulant intellectual exercise which is not uncommon in our community.

Further active/passive measurement integration. As anycast replicas are subject to BGP convergence, a long-standing myth is that it would forbid use of anycast for connection-oriented services relying on TCP. Given our results, this myth seems no longer holding. Yet, while we did not notice in our time frame significant changes in terms of IP-level path length, more valuable information would be needed from heterogeneous sources, and by combining active and passive measurements.

Appendix A

List of publications and awards

1 Publications

We report here the list of published papers.

Journal

[J1] Cicalese, Danilo, Joumblatt, Diana, Rossi, Dario, Buob, Marc-Olivier, Auge, Jordan and Friedman, Timur, ”Latency-Based Anycast Geolocalization: Algorithms, Software and Datasets,” in *IEEE Journal on Selected Areas of Communications, Special issue on Measuring and Troubleshooting the Internet, IEEE JSAC*, 2016. **Presented in Chap. 3**

Magazine

[M1] Cicalese, Danilo, Rossi, Dario, ”A longitudinal study of IP Anycast”, in *ACM Computer Communication Review, ACM CCR*, 2018. **Presented in Chap. 4**

Conference

[C1] Cicalese, Danilo, Joumblatt, Diana, Rossi, Dario, Buob, Marc-Olivier, Auge, Jordan and Friedman, Timur, ”A fistful of pings: Accurate and lightweight anycast enumeration and geolocation.” in *Proceedings of IEEE Conference on Computer Communications, IEEE INFOCOM*, 2015. **Presented in Chap. 3**

[C2] Cicalese, Danilo, Joumblatt, Diana, Rossi, Dario, Buob, Marc-Olivier, Auge, Jordan and Friedman, Timur, ”Characterizing IPv4 Anycast Adoption and Deployment.” in *Proceedings of ACM Conference on Emerging Networking Experiments and Technologies, ACM CoNEXT*, 2015. **Presented in Chap. 4**

[C3] Salutari, Flavia and Cicalese, Danilo and Rossi, Dario J, ”A closer look at IP-ID behavior in the Wild”, in *Proceedings of International Conference on Passive and Active Network Measurement, PAM*, 2018.

Workshop

[W1] Danilo Giordano, Danilo Cicalese, Alessandro Finamore, Marco Mellia, Maurizio Munafò, Diana Zeaiter Joumblatt, Dario Rossi, ”A first characterization of anycast traffic from passive traces.,” in *IFIP Network Traffic Measurement and Analysis Conference, TMA*, 2016. **Presented in Chap. 5**

Demo

[D1] Cicalese, Danilo, Joumblatt, Diana, Rossi, Dario, Buob, Marc-Olivier, Auge, Jordan and Friedman, Timur, ”A lightweight anycast enumeration and geolocation.” in *IEEE INFOCOM, Demo Session*, 2015. **Presented in Chap. 3**

[D2] Baron, Loic, Scognamiglio, Ciro, Rahman, Mohammed Yasin. Klacza. Radomir, Cicalese, Danilo, Kurose, Nina, Friedman, Timur, Fdida Serge, ”Onelab: Major computer networking testbeds open to the ieee infocom community.” in *IEEE INFOCOM, Demo Session*, 2015.

Other Dissemination

- Cicalese, Danilo and Rossi, Dario, ”Where are the anycasters?,” in *NANOG 66*, 2016.

- Cicalese, Danilo and Rossi, Dario, ”Data plane BGP Hijack detection via latency measurement,” in *Workshop on Active Internet Measurements, AIMS*, 2016.

- Cicalese, Danilo, Joumblatt, Diana, Rossi, Dario, Buob, Marc-Olivier, Auge, Jordan and Friedman, Timur, ”Where are the anycasters?,” in *Ripe 71*, 2015.

- Cicalese, Danilo, Joumblatt, Diana, Rossi, Dario, Buob, Marc-Olivier, Auge, Jordan and Friedman, Timur, ”Where are the anycasters? Reloaded,” in *Ripe 71 - MAT working group* , 2015.

- Cicalese, Danilo, Joumblatt, Diana, Rossi, Dario, Buob, Marc-Olivier, Auge, Jordan and Friedman, Timur, ”Anycast census and geolocation.,” in *Workshop on Active Internet Measurements, AIMS*, 2015.

- Cicalese, Danilo, Joumblatt, Diana, Rossi, Dario, Buob, Marc-Olivier, Auge, Jordan and Friedman, Timur, ”Anycast enumeration and geolocation,” in *RESCOM*, 2015.

- Cicalese, Danilo and Friedman, Timur, ”Onelab, future internet testbeds.,” in *RESCOM*, 2015.

2 Awards

- Our paper [54] was among the winners of the *Applied Networking Research Prize - Internet Research Task Force*, 2016.
- BGP Hackaton winning team, *CAIDA BGP Hackathon*, San Diego, USA, 2016.
- Best poster at *IFIP Network Traffic Measurement and Analysis Conference, TMA*, Louvain La Neuve, Belgium, 2016.
- 2nd prize of *IEEE Communications Society (ComSoc) France Chapter Price*, 2015.

Appendix B

iGreedy usage

1 iGreedy Usage

For completeness, we provide an overview of the capabilities of the version of iGreedy released on at [22].

This section is loosely based on the README file released in the tool [22]. Briefly, the tool allows to (i) analyze existing measurement or (ii) generate and analyze new measurement (iii) visualize the measurement on a GoogleMap. The package also contains (iv) datasets corredated with ground-truth to assess the accuracy of the tool. Notice that the released version does not allow to generate new ground-truth, as this step often involves some level of manual verification (e.g., recall the SGW example in the Ground Truth section), and is thus difficult to fully automate.

1.1 Installation and configuration

iGreedy should run out of the box: iGreedy is written in Python (2.7 flavor) and there is no Python depenedency which we are aware of: all the code you need is in the `code/` folder of the tarball.

While running iGreedy on the provided datasets does not require any special configuration, however to launch new measurement from RIPE Atlas you need to (i) have a RIPE Atlas account (ii) have enough credits (iii) configure your authentication. Measurement are launched by `code/RIPEAtlas.py` which is going to read your RIPE Atlas key from the `datasets/auth` file.

Finally, we assume that the chrome (or chromium) Web-browser is installed (which in our experience renders the Google maps better), at it is automatically launched whenever users request to display results in a map.

1.2 Usage

iGreedy faciliates interaction by providing some command line parameters to either process existing measurement from an input file (`-i input`) or to launch new measurement toward a target alive IP address (`-m target`). This choice is the only mandatory parameter `(-i input|-m target)`.

In case Ground Truth (GT, preferable) or Public Available Information (PAI, in lack of GT) is available, then it can be specified (`-g groundtruth`). The format of GT or PAI files is very simple: GT files correlate IATA code seen by each vantage point (each line has two fields following a `hostname iata` format); PAI files only contain a list of `iata` airport codes, one per line, associated with the target (e.g., on a public webpage).

The output files (CSV and JSON) can be customized (`-o outputfile`), possibly triggering in-browser graphical rendering of the geographical map (`-b`). The code allows to tune the disk threshold (by default set to ∞) and the latency vs population weight α (by default set to $\alpha = 1$). Changing the default settings is discouraged from a performance and accuracy perspective, but it can be nevertheless useful for verification or understanding of the underlaying algorithm.

1.3 Examples: process historic measurement

We now provide a list of examples to process historic measurement released in the code. For instance, to run iGreedy on the F root server of the provided dataset:

```
./igreedy -i datasets/measurement/f-ripe
```

To run iGreedy over the F root server dataset, showing results on a map (opening your browser):

```
./igreedy -i datasets/measurement/f-ripe -b
```

To run iGreedy over the F root server dataset, showing results and ground truth on a map (opening your browser):

```
./igreedy -i datasets/measurement/f-ripe \
  -g datasets/ground-truth/f-ripe -b
```

To run iGreedy over the EdgeCast dataset measured (from RIPE Atlas), using publicly available information (from Webpages):

```
./igreedy -i datasets/measurement/edgecast-ripe \
 -g datasets/public-available-information/edgecast
```

To run iGreedy over the CloudFlare dataset measured (from PlanetLab), using ground truth information (from PlanetLab):

```
./igreedy -i datasets/measurement/cloudflare-planetlab \
 -g datasets/ground-truth/cloudflare-planetlab
```

1.4 Examples: run and process new measurements

The released version allows to run iGreedy from RIPE Atlas. The version we use internally allows to run measurement from RIPE Atlas or PlanetLab. As getting a RIPE Atlas key is much easier than obtaining, configuring and maintaing a PlanetLab slice (on which we surely wouldn't want to provide support or assistance), we do not intend to let the version of the code supporting multiple MIs public anytime soon (but feel free to contact us via email if you are an experienced PlanetLab user and want to profit of iGreedy results gathered on this platform as well).

Configuring RIPE Atlas key. Simply put your RIPE Atlas key in this file: `datasets/auth`. Voilà, iGreedy is now ready to run new RIPE Atlas measurement with your account.

Configuring RIPE Atlas vantage points. As the sensitivity in Sec.5.3 illustrated, iGreedy results are highly dependent on the vantage point selection: a bad vantage point selection (e.g., few vantage points, or bad geographical coverage) is expected to yield bad enumeration/geolocation results. Vantage points selection is made by modifying the content of the `datasets/ripe-vps` file. Depending on the aim (eg. detection, geolocation, census) each experiment may need a careful assessment of the tradeoff coverage-number of RIPE Atlas credits.

To simplify bootstrap, we provide two examples configurations that are rather conservative in the number of probes but useful for anecdotal use of detection (`datasets/ripe- vps.rand10`) and enumeration/geolocation (`ripe-vps.suggested200`). It is worth stressing that the set of RIPE Atlas vantage points used by default (`datasets/ripe-vps`) is conservative (10 random probes `ripe-vps.rand10`) and useful at most for detection (and to avoid burning all your credits with a single "for" loop. The set of `ripe-vps.suggested200` is again very conservative and useful for familiarizing with the tool before launching a measurement campaign. We count on releasing slightly larger but more accurate defaults on the website.

Hello, Anycast world. Finally, you are ready to run new measurement with iGreedy. For instance, to run iGreedy on the F root server 192.5.5.241,

```
 ./igreedy -m 192.5.5.241 -b
```

note that the set of new measurements is saved in `datasets/measurement` for further post-processing. Have fun!

Bibliography

[1] Edgecast. http://www.edgecast.com.

[2] Internet Systems Consortium. 21 Oct 2002 Root Server Denial of Service Attack - Report. https://web.archive.org/web/20110302164416/http://www.isc.org/f-root-denial-of-service-21-oct-2002.

[3] OpenDNS. Data Center Locations. https://www.opendns.com/data-center-locations.

[4] Shadow server. Open Resolver Scanning Project. https://dnsscan.shadowserver.org/.

[5] The sound of dial-up internet. https://www.youtube.com/watch?v=gsNaR6FRuO0.

[6] Internet addresses hitlist dataset. Provided by the USC/LANDER project http://www.isi.edu/ant/lander, 2006.

[7] Team Cymru. IP to ASN mapping. http://www.team-cymru.org/Services/ip-to-asn.html, 2008.

[8] Carna Botnet. Internet Census 2012: Port scanning/0 using insecure embedded devices. http://census2012.sourceforge.net/paper.html, 2012.

[9] Internet Society. Who Makes the Internet Work: The Internet Ecosystem. https://www.internetsociety.org/internet/who-makes-it-work, 2014.

[10] Edgecast continued growth. http://www.edgecast.com/company/news/edgecast-continued-growth, 2016.

[11] Edgecast. Network status. https://status.edgecast.com, 2016.

[12] http://www.telecom-paristech.fr/~drossi/anycast, 2018.

[13] https://github.com/TeamRossi/anycast-census, 2018.

[14] CAIDA. Archipelago (Ark) Measurement Infrastructure. http://www.caida.org/projects/ark, 2018.

[15] Caida. Measurement Infrastructure Comparison. http://www.caida.org/research/performance/measinfra/evaltable.xml, 2018.

[16] Cloudflare. http://www.cloudflare.com, 2018.

[17] Cloudflare. Amsterdam to Zhuzhou: Cloudflare network expands to 100 cities. https://blog.cloudflare.com/amsterdam-to-zhuzhou-cloudflare-global-network/, 2018.

[18] Cloudflare. Cloudflare system status. http://www.cloudflarestatus.com, 2018.

[19] Cloudflare. The Cloudflare Global Anycast Network. http://www.cloudflare.com/network, 2018.

[20] Fastly. A new architecture for the modern internet. https://www.fastly.com/network-map/, 2018.

[21] ICANN Managed Root-Server Locations. http://lrootmap.dns.icann.org/, 2018.

[22] iGreedy. Open-source software for anycast detection, enumeration and geolocation. http://goo.gl/7ESrCR, 2018.

[23] MASSCAN: Mass IP port scanner. https://github.com/robertdavidgraham/masscan, 2018.

[24] Maxmind. IP geolocation and online fraud prevention. https://www.maxmind.com, 2018.

[25] Planetlab. https://www.planet-lab.org, 2018.

[26] RIPE atlas. https://atlas.ripe.net, 2018.

[27] RIPE NCC. OpenIPmap, a Collaborative Approach to Mapping Internet Infrastructure. https://openipmap.ripe.net/, 2018.

[28] Verizon. health check: Network status. https://status.verizondigitalmedia.com/, 2018.

[29] W3Techs - World Wide Web Technology Surveys. Usage of web servers for websites. http://w3techs.com/technologies/overview/web_server/all, 2018.

[30] E. Aben. Has the routability of longer-than-/24 prefixes changed? https://labs.ripe.net/Members/emileaben/has-the-routability-of-longer-than-24-prefixes-changed, 2015.

[31] E. Aben. Measuring Countries and IXPs with RIPE Atlas. https://labs.ripe.net/Members/emileaben/measuring-ixps-with-ripe-atlas, 2015.

[32] J. Abley and K. Lindqvist. Operation of Anycast Services. IETF RFC 4786, 2006.

[33] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman. Zippier ZMap: Internet-Wide Scanning at 10 Gbps. In *USENIX Workshop on Offensive Technologies*, 2014.

[34] Akamai. `http://www.akamai.com`.

[35] B. Al-Musawi, P. Branch, and G. Armitage. BGP anomaly detection techniques: A survey. *IEEE Communications Surveys & Tutorials*, 2017.

[36] Z. Al-Qudah, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van der Merwe. Anycast-aware transport for content delivery networks. In *ACM International conference on World Wide Web*, 2009.

[37] H. A. Alzoubi, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van Der Merwe. A practical architecture for an anycast CDN. *ACM Transactions on the Web*, 2011.

[38] `http://www.root-servers.org`.

[39] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding traceroute anomalies with Paris traceroute. In *ACM Internet Measurement Conference*, 2006.

[40] V. Bajpai and J. Schonwalder. A Survey on Internet Performance Measurement Platforms and Related Standardization Efforts. *IEEE Communications Surveys & Tutorials*, 2015.

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

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

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

[44] B. Barber, M. Larson, and M. Kosters. Traffic Source Analysis of the J Root Anycast instances. 39th Nanog, 2006.

[45] I. N. Bermudez, M. Mellia, M. M. Munafo, R. Keralapura, and A. Nucci. DNS to the rescue: discerning content and services in a tangled web. ACM Internet Measurement Conference, 2012.

[46] R. Beverly and A. Berger. Server Siblings: Identifying Shared IPv4/IPv6 Infrastructure Via Active Fingerprinting. In *Passive and Active Measurement Workshop*. 2015.

[47] P. Boothe and R. Bush. DNS anycast stability: Some early results. 19th APNIC, 2005.

[48] R. Braden. *RFC 1122, Requirements for Internet Hosts – Communication Layers*, 1989.

[49] J. Brutlag. Speed matters for Google web search, 2009.

[50] CacheFly. http://www.cachefly.com/about.html.

[51] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan. Mapping the expansion of google's serving infrastructure. In *ACM Internet Measurement Conference*, 2013.

[52] M. Calder, A. Flavel, E. Katz-Bassett, R. Mahajan, and J. Padhye. Analyzing the Performance of an Anycast CDN. In *ACM Internet Measurement Conference*, 2015.

[53] C. Chirichella and D. Rossi. To the Moon and back: are Internet bufferbloat delays really that large. In *IFIP Network Traffic Measurement and Analysis Conference*, 2013.

[54] D. Cicalese, J. Auge, D. Joumblatt, T. Friedman, and D. Rossi. Characterizing IPv4 Anycast Adoption and Deployment. In *Proc. ACM Conference on Emerging Networking Experiments and Technologies*, 2015.

[55] D. Cicalese, D. Giordano, A. Finamore, M. Mellia, M. Munafò, D. Rossi, and D. Joumblatt. A first look at anycast cdn traffic. *arXiv preprint arXiv:1505.00946*, 2015.

[56] 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 *IEEE INFOCOM*, 2015.

[57] D. Cicalese, D. Joumblatt, D. Rossi, M.-O. Buob, J. Auge, and T. Friedman. Latency-based anycast geolocalization: Algorithms, software and datasets. *IEEE Journal on Selected Areas of Communications*, 2016.

[58] Cloudflare. Fast, global content delivery network. https://www.cloudflare.com/features-cdn.

[59] L. Colitti. Measuring anycast server performance: The case of K-root. 37th Nanog, 2006.

[60] D. Dagon, N. Provos, C. P. Lee, and W. Lee. Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority. In *Network and Distributed System Security Symposium*, 2008.

[61] A. Dainotti, K. Benson, A. King, B. Huffaker, E. Glatz, X. Dimitropoulos, P. Richter, A. Finamore, and A. C. Snoeren. Lost in space: Improving inference of ipv4 address space utilization. *IEEE Journal on Selected Areas of Communications*, 2016.

[62] R. de Oliveira Schmidt, J. Heidemann, and J. H. Kuipers. Anycast latency: How many sites are enough? In *Passive and Active Measurement Workshop*, 2017.

[63] http://www.enst.fr/~drossi/anycast, 2016.

[64] P. Dixon. Shopzilla site redesign: We get what we measure. In *Velocity Conference Talk*, 2009.

[65] R. Durairajan, S. Ghosh, X. Tang, P. Barford, and B. Eriksson. Internet Atlas: A Geographic Database of the Internet. In *ACM HotPlanet*, 2013.

[66] Z. Durumeric, M. Bailey, and J. A. Halderman. An internet-wide view of internet-wide scanning. In *USENIX Security Symposium*, 2014.

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

[68] R. Ensafi, J. Knockel, G. Alexander, and J. R. Crandall. Detecting intentional packet drops on the Internet via TCP/IP side channels. In *Passive and Active Measurement Workshop*. 2014.

[69] B. Eriksson, P. Barford, J. Sommers, and R. Nowak. A learning-based approach for IP geolocation. In *Passive and Active Measurement Workshop*, 2010.

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

[71] T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric intersection graphs. In *SIAM Journal on Computing*, 2005.

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

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

[74] A. Finamore, M. Mellia, M. Meo, M. Munafo, and D. Rossi. Experiences of Internet Traffic Monitoring with Tstat. *IEEE Network*, 2011.

[75] 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 *USENIX Symposium on Networked Systems Design and Implementation*, 2015.

[76] M. J. Freedman, K. Lakshminarayanan, and D. Mazières. OASIS: Anycast for Any Service. In *USENIX Symposium on Networked Systems Design and Implementation*, 2006.

[77] G. Gallois-Montbrun and V. Nguyen. Defining a minimum vantage point selection to detect bgp hijack with igreedy.
http://perso.telecom-paristech.fr/ drossi/teaching/INF570/projects/2015-paper-3.pdf, 2015.

[78] D. Giordano, D. Cicalese, A. Finamore, M. Mellia, M. Munafo, D. Joumblatt, and D. Rossi. A first characterization of anycast traffic from passive traces. In *IFIP Network Traffic Measurement and Analysis Conference*, 2016.

[79] V. Giotsas, G. Smaragdakis, B. Huffaker, and M. Luckie. Mapping peering interconnections to a facility. In *Proc. ACM Conference on Emerging Networking Experiments and Technologies*, 2015.

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

[81] H. Guo and J. Heidemann. Detecting ICMP Rate Limiting in the Internet. In *Passive and Active Measurement Workshop*, 2018.

[82] J. Hamilton. The cost of latency, 2009.

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

[84] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman. Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices. In *USENIX Security Symposium*, 2012.

[85] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. RFC 4291, 2006.

[86] R. Holz, L. Braun, N. Kammenhuber, and G. Carle. The SSL landscape: a thorough analysis of the X.509 PKI using active and passive measurements. In *ACM Internet Measurement Conference*, 2011.

[87] X. Hu and Z. M. Mao. Accurate real-time identification of IP prefix hijacking. In *IEEE Symposium on Security and Privacy*, 2007.

[88] G. Huston, M. Rossi, and G. Armitage. Securing BGP – A literature survey. *IEEE Communications Surveys & Tutorials*, 2011.

[89] ICANN. Factsheet - Root server attack on 6 February 2007. `https://www.icann.org/en/system/files/files/factsheet-dns-attack-08mar07-en.pdf`.

[90] K. Jansen. Approximation algorithms for geometric intersection graphs. In *Graph-Theoretic Concepts in Computer Science*, 2007.

[91] D. Johnson and S. Deering. Reserved IPv6 Subnet Anycast Addresses. RFC 2526, 1999.

[92] D. Karrenberg. Anycast and BGP Stability: A Closer Look at DNSMON Data. 34th Nanog, 2005.

[93] D. Katabi and J. Wroclawski. A Framework for Scalable Global IP-anycast (GIA). In *ACM SIGCOMM*, 2000.

[94] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: illuminating the edge network. In *ACM Internet Measurement Conference*, 2010.

[95] T. Krenc, O. Hohlfeld, and A. Feldmann. An internet census taken by an illegal botnet: A qualitative assessment of published measurements. 2014.

[96] S. Laki, P. Mátray, P. Hága, T. Sebok, I. Csabai, and G. Vattay. Spotter: A model based active geolocation service. In *IEEE INFOCOM*, 2011.

[97] D. Leonard and D. Loguinov. Demystifying Internet-wide service discovery. *IEEE/ACM Transactions on Networking*, 2013.

[98] M. Levine, B. Lyon, and T. Underwood. Operational experience with TCP and Anycast. 37th Nanog, 2006.

[99] J. Liddle. Amazon found every 100ms of latency cost them 1% in sales. `http://blog.gigaspaces.com/amazon-found-every-100ms-of-latency-cost-them-1-in-sales`.

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

[101] S. Lohr. For impatient web users, an eye blink is just too long to wait. *New York Times*, 2012.

[102] G. F. Lyon. *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure, 2009.

[103] M. Linsner, P. Eardley, T. Burbridge, F. Sorensen. Large-scale broadband measurement use cases. *IETF RFC7536*, 2015.

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

[105] B. M. Maggs and R. K. Sitaraman. Algorithmic nuggets in content delivery. *ACM SIGCOMM Computer Communication Review*, 2015.

[106] D. McPherson, E. Osterweil, D. Oran, and D. Thaler. Architectural considerations of IP anycast. *IETF RFC 7094*, 2014.

[107] Microsoft. The bing network audience. `https://advertise.bingads.microsoft.com/en-us/insights/planning-tools/bing-network-audience`, 2018.

[108] A. Mirian, Z. Ma, D. Adrian, M. Tischer, T. Chuenchujit, T. Yardley, R. Berthier, J. Mason, Z. Durumeric, J. A. Halderman, et al. An Internet-Wide View of ICS Devices. In *IEEE Privacy, Security and Trust Conference*, 2016.

[109] C. Moallemi and M. Saglam. The cost of latency. *SSRN eLibrary*, 2010.

[110] G. Moura, R. d. O. Schmidt, J. Heidemann, W. B. de Vries, M. Muller, L. Wei, and C. Hesselman. Anycast vs. DDoS: evaluating the November 2015 root DNS event. In *ACM Internet Measurement Conference*, 2016.

[111] C. Orsini, A. King, D. Giordano, V. Giotsas, and A. Dainotti. BGPStream: a software framework for live and historical BGP data analysis. In *ACM Internet Measurement Conference*, 2016.

[112] C. Partridge, T. Mendez, and W. Milliken. Host anycasting service. IETF RFC 1546, 1993.

[113] P. Pearce, R. Ensafi, F. Li, N. Feamster, and V. Paxson. Augur: Internet-wide detection of connectivity disruptions. In *IEEE Symposium on Security and Privacy*, 2017.

[114] C. Pelsser, L. Cittadini, S. Vissicchio, and R. Bush. From Paris to Tokyo: On the suitability of ping to measure latency. In *ACM Internet Measurement Conference*, 2013.

[115] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye. IP geolocation databases: Unreliable? *ACM SIGCOMM Computer Communication Review*, 2011.

[116] D. Rossi, G. Pujol, X. Wang, and F. Mathieu. Peeking through the BitTorrent seedbox hosting ecosystem. In *IFIP Network Traffic Measurement and Analysis Conference*, 2014.

[117] C. Rossow. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *Network and Distributed System Security Symposium*, 2014.

[118] S. Sarat, V. Pappas, and A. Terzis. On the use of anycast in DNS. In *International Conference of Computer Communications and Networks*, 2006.

[119] E. Schurman and J. Brutlag. Performance related changes and their user impact. In *Velocity, the Web Performance and Operations Conference*, 2009.

[120] Y. Shavitt and N. Zilberman. A geolocation databases study. *IEEE Journal on Selected Areas in Communications*, 2011.

[121] A. Singla, B. Chandrasekaran, P. B. Godfrey, and B. Maggs. The internet at the speed of light. In *Proc. SIGCOMM Workshop on Hot Topics in Networking*, 2014.

[122] S. Stoikov and R. Waeber. Reducing transaction costs with low-latency trading algorithms. *Quantitative Finance*, 2016.

[123] F. Streibelt, J. Böttger, N. Chatzis, G. Smaragdakis, and A. Feldmann. Exploring EDNS-client-subnet Adopters in Your Free Time. In *ACM Internet Measurement Conference*, 2013.

[124] Top hat. `http://top-hat.info`.

[125] B. Trammell, P. Casas, D. Rossi, A. Bar, Z. Ben-Houidi, I. Leontiadis, T. Szemethy, and M. Mellia. mPlane: an Intelligent Measurement Plane for the Internet. *IEEE Communications Magazine*, 2014.

[126] M. Trevisan, F. Alessandro, M. Mellia, M. Munafò, and D. Rossi. DPD-KStat: 40Gbps Statistical Traffic Analysis with Off-the-Shelf Hardware. *Technical report*, 2016.

[127] Twitter. The infrastructure behind twitter: Scale. `https://blog.twitter.com/engineering/en_us/topics/infrastructure/2017/the-infrastructure-behind-twitter-scale.html`, 2018.

[128] S. Valenti, D. Rossi, A. Dainotti, A. Pescape, A. Finamore, and M. Mellia. Reviewing traffic classification. In *Data Traffic Monitoring and Analysis: From measurement, classification and anomaly detection to Quality of Experience*. 2013.

[129] L. Wei and J. Heidemann. Does anycast hang up on you? In *IFIP Network Traffic Measurement and Analysis Conference*, 2017.

[130] S. Woolf and D. Conrad. Requirements for a Mechanism Identifying a Name Server Instance. IETF RFC 4892, 2007.

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

[132] H. Zeng, P. Kazemian, G. Varghese, and N. McKeown. A survey on network troubleshooting. Technical report, 2012.

TITRE

Danilo CICALESE

RESUME : Les motivations des recherches réalisées dans cette thèse viennent de la curiosité de découvrir IP anycast. Cette technique est couramment utilisée pour partager la quantité dŠinformation d'une variété de services globaux. L'adoption de cette technique a augmenté au cours de ces dernières années. Une fois reléguée aux serveurs racine et domaine de premier niveau du système de noms de domaine (DNS), anycast est maintenant communément utilisé par les réseaux de distribution de contenu (CDN) et d'autres acteurs clés d'Internet. Dans cette thèse, nous visons à sensibiliser la communauté de l'utilisation d'IP anycast en composant une image complète afin de montrer quelles entreprises l'utilisent et comment elles le font.

Tout d'abord, nous nous concentrons sur l'identification d'une technique, qui ne depends pas d'un protocole specific, utilisé pour la découverte et la géolocalisation des répliques anycast. D'autres techniques utilisées pour identifier et énumérer des répliques anycast existent déjà. Toutefois, ils exploitent les spécificités du protocole DNS, ce qui limite leur applicabilité à ce service. Nous fournissons également à la communauté des logiciels open-source avec des jeu de donnée. Ceux-ci servent à reproduire nos résultats expérimentaux et à faciliter le développement de nouvelles techniques.

Cette méthodologie nous a permis de mettre en oeuvre la prochaine étape de notre recherche : dévoiler toutes les entreprises qui utilisent actuellement anycast pour leurs services. Nous effectuons plusieurs IPv4 census d'anycast, en utilisant des mesures de latence à partir d'une plateforme distribuée. Nous collectons et les analysons. Ces census révèlent finalement que de nombreuses grandes entreprises d'Internet utilisent anycast. De plus identifier les sociétés et avoir des informations à jour sur les IP anycast et leur géolocalisation peuvent servir à d'autres finalités. Nous avons donc décidé de mettre en place un système capable d'effectuer des census IPv4 mensuels et d'analyser les résultats d'un an. Nos résultats, données et codes sont également partagés avec la communauté.

Enfin, pour compléter l'étude, nous analysons les utilisateurs et les services que les CDN anycast diffusent sur Internet. Nous effectuons une caractérisation passive en mettant l'accent sur les services qu'ils offrent et leur pénétration etc. Nos résultats révèlent que normalment plus de 50% des internautes accèdent au contenu servi par les CDN anycast. Une large gamme de services TCP (Transmission Control Protocol) est offerte sur anycast. Celle-ci peut inclure des services d'audio, de streaming vidéo ou des HTTP & HTTPS, ces derniers étant les plus populaires.

MOTS-CLEFS : Geolocation, Anycast, Surveillance du réseau, Mesure de réseau, IPv4