# MAnycast Reloaded: a Tool for an Open, Fast, Responsible and Efficient Daily Anycast Census

Remi Hendriks University of Twente remi.hendriks@utwente.nl Matthew Luckie CAIDA/UC San Diego mjl@caida.org Mattijs Jonker University of Twente m.jonker@utwente.nl

Raffaele Sommese University of Twente r.sommese@utwente.nl

# Roland van Rijswijk-Deij University of Twente

r.m.vanrijswijk@utwente.nl

## **ABSTRACT**

IP anycast is a widely adopted technique in which an address is replicated at multiple locations, to, e.g., reduce latency and enhance resilience. Due to anycast's crucial role on the modern Internet, earlier research introduced tools to perform anycast censuses. The first, iGreedy, uses latency measurements from geographically dispersed locations to map anycast deployments. The second, MAnycast<sup>2</sup>, uses anycast to perform a census of other anycast networks. MAnycast<sup>2</sup>'s advantage is speed, performing an Internet-wide census in 3 hours, but it suffers from problems with accuracy and precision. Inversely, iGreedy is highly accurate but much slower. On top of that, iGreedy has a much higher probing cost.

In this paper we address the shortcomings of both systems and present MAnycast Reloaded (MAnycastR). Taking MAnycast<sup>2</sup> as a basis, we completely redesign its measurement pipeline, and add support for distributed probing, additional protocols (UDP, TCP and IPv6) and latency measurements similar to iGreedy. We validate MAnycastR on an anycast testbed with 32 globally distributed nodes, compare against an external anycast production deployment and extensive latency measurements with RIPE Atlas, and crosscheck over 60% of detected anycast prefixes against operator ground truth. This shows that MAnycastR achieves high accuracy and precision. We make continual daily MAnycastR censuses available to the community and release the source code of the tool under a permissive open source license.

# 1 INTRODUCTION

Anycast is a technique where an IP address is made available in multiple, autonomous locations such that packets are routed to one of several available locations [24]. This technique is widely used to satisfy client requests from multiple locations to ensure low-latency responses. Anycast also allows for load-balancing by distributing traffic over multiple sites. Finally, it can enhance resilience by ensuring the availability of a service in multiple locations that can absorb traffic in case one location goes down. Prominent examples of

services that use anycast include the Domain Name System (DNS) [29] and Content Delivery Networks (CDNs) [8, 23].

Given its widespread use, determining how widely and for what services anycast is used is vital to our understanding of the modern-day Internet. Several techniques have been developed to perform Internet-wide IP anycast censuses. The two most prominent ones are iGreedy [9] and MAnycast<sup>2</sup> [30]. The first, iGreedy, uses latency measurements from geographically dispersed locations to infer anycast use. iGreedy detects anycasted prefixes by looking for "speed-of-light violations" (i.e., latencies from two locations to a target that are physically impossible). A full census of the IPv4 address space using iGreedy takes days [9]. The second technique, MAnycast<sup>2</sup>, involves probing from multiple Vantage Points (VPs) using an anycast source address. A target will send its responses to the VP that is closest in terms of BGP routing. A unicast target will therefore send responses from one location to a single VP, whereas an anycast target's responses are sent from multiple locations to multiple VPs. The advantages of MAnycast<sup>2</sup> are its speed, performing a full census of the routable IPv4 space in under 3 hours, and that it needs a far more modest probing budget to discover anycasted prefixes. Unfortunately, MAnycast<sup>2</sup> also suffers from problems with accuracy and precision. It is particularly prone to misclassifying targets as anycast for cases where responses to probes end up at a low number of VPs [30].

In this work, we tackle these shortcomings and present MAnycast Reloaded (MAnycastR). Taking MAnycast<sup>2</sup> as a starting point, we design and build a new measurement pipeline from the ground up. We improve accuracy by adding support for synchronized probing. We also incorporate the latency-based measurement algorithm from iGreedy. MAnycastR's streamlined implementation outperforms MAnycast<sup>2</sup> in terms of speed, allowing for a census and accompanying latency-based validation to be performed daily. In other words: MAnycastR is faster than MAnycast<sup>2</sup> (and thus much faster than iGreedy), while being just as accurate as iGreedy.

The contributions of this paper are that we:

1

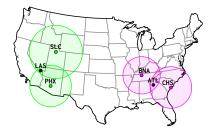


Figure 1: Using latency measurements from VPs (green, purple) for GCD-based anycast site (black) detection.

- Design and implement MAnycastR from the ground up;
- Add support for IPv6, UDP- and TCP-based probing;
- Validate MAnycastR's performance against ground truth from operators, measurements on an anycast production deployment, commercial datasets, and extensive latencybased measurements through RIPE Atlas;
- Release daily anycast censuses for the full IPv4 address space and state-of-the-art IPv6 hitlists<sup>1</sup>;
- Provide a user-friendly dashboard to query the census<sup>2</sup>;
- Release the full source code for MAnycastR under a permissive open source license (upon publication).

The remainder of this paper is organized as follows. First, we provide background on iGreedy and MAnycast<sup>2</sup> and discuss other related work on anycast measurement methodologies in §2. In §3 we discuss the requirements for MAnycastR. Next, in §4 we introduce the design and implementation of MAnycastR. §5 discusses our extensive validation. Finally, we conclude in §6.

## 2 BACKGROUND AND RELATED WORK

This section provides background information on iGreedy and its latency-based anycast detection method and on the MAnycast<sup>2</sup> anycast-based approach. We end with a discussion of other related work on anycast detection approaches.

## 2.1 iGreedy and the Great Circle Distance

The first comprehensive anycast census technique is iGreedy, introduced by Cicalese et al. in 2015 [9]. iGreedy's anycast detection approach is based on using the *Great-Circle Distance* (GCD). iGreedy leverages assumptions about the speed of light to determine the maximum geographical distance an Internet packet travels based on its Round-Trip Time (RTT).

By capturing latency data from multiple VPs to a probed target, iGreedy determines the area in which the target must reside using GCD. Figure 1 illustrates this principle. A probe is sent from each VP (green and purple dots). The RTT for



Figure 2: Using an anycast setup (light blue) to detect anycast. Probes to a unicast site (orange) result in responses to a single VP (blue), probes to anycast sites (green) result in responses to multiple VPs (red).

these probes is then used to draw a circle indicating the distance a packet can travel. The point where the circles intersect is then used to approximate the location of the target. Wherever there are non-overlapping circles (green and purple circles in the example), iGreedy infers the target to be anycast as it must be at multiple locations. Enumeration is achieved by determining the minimum set of areas required to satisfy a valid location for all VPs, thereby not violating any speed-of-light constraints. Next, geolocation is achieved by picking the most likely city (based on population counts) where the anycast site resides in each area.

The GCD approach relies on assumptions about the speed at which packets travel. iGreedy's default is the speed of light in fibre (~200,000 km/s). This estimate discards, e.g., delays due to buffering. Therefore, the GCD approach likely underestimates both the number of anycasted IP prefixes and the number of sites per prefix. In particular, it may fail to detect anycast deployed in close proximity (regional anycast).

# 2.2 MAnycast<sup>2</sup>

MAnycast<sup>2</sup> takes a different approach by using anycast to detect anycast. The basic principle behind MAnycast<sup>2</sup> is based on an intuition by De Vries et al. [13] who, in their paper, test "Verfploeter", an approach to map anycast catchments, with a catchment defined as all prefixes that send their traffic to a given point of presence of an anycast service. When De Vries et al. tested catchment stability, they noticed that some prefixes showed up in the catchment of many locations. They intuited that these prefixes were themselves using anycast, and confirmed some of this with ground truth about DNS root servers.

MAnycast<sup>2</sup> leverages this to map anycast. Figure 2 illustrates the techique, showing how probing a unicast address from an anycast setup leads to responses being received by a single VP, whereas probing an anycast address leads to responses at multiple VPs. As such, if a single VP receives all responses for a given target, MAnycast<sup>2</sup> infers it to be

<sup>&</sup>lt;sup>1</sup>https://github.com/anycast-census/anycast-census

<sup>&</sup>lt;sup>2</sup>https://manycast.net

unicast, and when multiple VPs receive responses, the target is inferred to be anycast. However, there are corner cases where this intuition fails, resulting in the following possibilities: **True positive (TP)** - an anycast target has sites respond to different VPs; **True negative (TN)** - a unicast target consistently responds to the same VP; **False positive (FP)** - a unicast target responds to multiple VPs; **False negative (FN)** - an anycast target has all sites respond to the same VP.

The errors in this approach can occur for the following reasons: A **FP** can occur when the unicast target is 'in between' two VPs, meaning there are equal-cost (in terms of BGP hops) paths to the VPs where, e.g., Equal-Cost Multi-Path (ECMP) routers send responses to both sites. Furthermore, route flips may cause a target to respond to a different VP when a flip occurs between sequential MAnycast<sup>2</sup> probes. A **FN** can occur when a probed target's anycast sites are in the catchment of a single VP, meaning all sites of the target will respond to the same VP. To minimize FNs the measuring anycast infrastructure must consist of many geographically dispersed VPs, lowering the odds of a probed anycast deployment being entirely in the catchment of a single VP.

Whilst MAnycast<sup>2</sup> is able to detect and enumerate anycast, iGreedy detects anycast with fewer FPs, and is able to geolocate anycast sites at a city-level granularity. At the same time, iGreedy suffers FNs when sites are geographically close. To find a middle ground between probing cost and reducing FPs, the MAnycast<sup>2</sup> paper proposes a combined approach, in which an anycast-based measurement creates a target set of potential anycast prefixes, which is then refined using an iGreedy measurement using a much larger number of VPs. In this paper, we build on this combined approach and publicly share daily census results of both methodologies.

#### 2.3 Other related work

Anycast is used extensively for the DNS, particularly for the DNS root servers. For troubleshooting purposes, it is useful to identify which instance a client reaches when it sends a query. For this reason, RFC 4892 specifies a mechanism for identifying anycast sites relying on CHAOS-class DNS records, where each site discloses its identity in a TXT record [33]. Through a CHAOS query, a client can then determine which site it reaches. CHAOS queries have been used to study anycast deployments. In 2011 Fan et al. proposed "ACE" (Anycast Characterization and Evaluation) that uses CHAOS records in combination with traceroute to detect and enumerate anycast [17]. Two years later Fan et al. improved the scope of this approach by proposing Internet-class (IN) DNS records that can be requested using existing recursive DNS infrastructure [16]. They find that 10 k VPs are required for a recall of 80% when enumerating the number of anycasted

replicas. Fan et al.'s approach is only usable for the DNS, whereas iGreedy and MAnycast<sup>2</sup> are protocol-agnostic.

Apart from iGreedy and MAnycast<sup>2</sup>, Bian et al. introduced one other approach for service-agnostic anycast detection. They passively collected BGP announcements to detect anycast by looking for geographically diverse upstreams [6]. However, their methodology suffered from false positives due to remote peering, since remote peering allows a unicast prefix to have upstreams in geographically distant locations.

In addition to developing iGreedy, Cicalese et al. also performed a finite longitudinal census of IP anycast on a monthly basis with iGreedy (covering December 2015 to May 2017) [10]. In this work, we significantly expand on this by providing a fine-grained daily longitudinal dataset that will enable detailed study of the IP anycast landscape.

# 3 SYSTEM REQUIREMENTS

We set the following goals in this work. We want to build a system that: i) combines anycast- and latency-based GCD measurements; ii) is accurate and precise; and iii) performs a responsible measurement in terms of probing budget. Furthermore, we want to release a continual daily census for the community and release the toolchain as open source for others to build on.

To achieve these goals we set the following requirements:

- R1 **Accuracy** MAnycastR must minimise false positives and false negatives. Additionally, MAnycastR must convey confidence in results through independently listing the classification (unicast, anycast, or unresponsive) for the anycast-based and GCD approach.
- R2 **Precision** The differences between subsequent censuses due to measurement errors must be minimised. In other words: MAnycastR must be precise over time.
- R3 **Responsibility** MAnycastR must perform measurements with low impact on the platform it runs on and the Internet as a whole. MAnycastR must support low probing rates without impacting accuracy and precision.
- R4 **Increased coverage** iGreedy and MAnycast<sup>2</sup> only support ICMPv4 probing. We want our system to also support IPv6, transport layer protocols (TCP, UDP) and service-aware probing.
- R5 **Robustness** A daily census requires a robust measurement system. External issues must be handled properly and be contained, e.g., a VP outage must not impact ongoing measurements besides that particular site.
- R6 **Timeliness** A full IPv4 census at /24-prefix granularity, and a hitlist-based IPv6 census must complete fast enough to be able to perform a census for multiple protocols in a single day.

- R7 **Ease of deployment** It must be easy to deploy MAnycastR on different anycast infrastructures that may exist for diverse hosting providers.
- R8 **Security** To protect measurement integrity, we must secure communication between components.
- R9 **Replicability** Independent parties must be able to run MAnycastR and make changes to the code.
- R10 **Efficiency** MAnycastR must support high probing rates and deployment on hosts with limited resources.
- R11 **Scalability** MAnycastR must function on anycast deployments of all sizes.

#### 4 DESIGN AND MEASUREMENT SETUP

In this section, we detail the design of the MAnycastR system and show how it meets the requirements from §3. We then explain how we deploy MAnycastR for validation and in production to create daily anycast censuses.

# 4.1 System design

In this work we reimplement, from scratch, the MAnycast<sup>2</sup> approach pioneered by Sommese et al. (see §2.2). Furthermore, we integrate a latency-based approach inspired by iGreedy, and include an improved version of the iGreedy iterative enumeration and geodetection process that significantly reduces processing time, from hours to minutes.

4.1.1 Components. The tool consists of three components: CLI – Takes a measurement as command-line input, and instructs the Orchestrator to perform that measurement. Orchestrator – Central controller of MAnycastR.

**Worker** – Deployed at the anycast sites and receives instructions from the Orchestrator to probe the Internet.

Our design minimizes the burden on Workers by offloading computation to the Orchestrator whenever possible, allowing us to operate the census at a low cost. The separate CLI component allows measurements to be started from multiple environments (*e.g.*, locally).

4.1.2 Measurement process. When starting a measurement, the CLI takes a measurement definition as input and forwards this to the Orchestrator. The Orchestrator then instructs all Workers that a measurement is starting, including a definition of the measurement such that Workers know what probes to send and listen for. The Workers then start listening. Next, the Orchestrator streams a list of IPs – at the CLI-defined probing rate – to each Worker. Workers send out probes as they receive hitlist targets from the Orchestrator. Each hitlist target receives a single probe from each Worker, the Orchestrator orchestrates it such that these probes are scheduled within a configurable offset after each other. When performing an ICMP-based measurement at an offset of 1 second, this means that from the target's perspective

they receive a set of probes from the same source address one second after each other, mimicking a regular ping sequence. We encode information regarding the sending Worker ID and the transmit time in fields that are echoed in responses from targets. For ICMP this is achieved using the ICMP payload, for DNS we encode information in the domain name of the request, and for TCP we use the acknowledgement number.

If a probed target is responsive, it sends back a response for each received probe, all these responses are routed to the nearest Worker, except for the case of the probed target being anycast as explained in §2. The Workers capture responses to probes and ensure they belong to the ongoing measurement by checking the information encoded in the probe that is echoed by the target. It then creates a result consisting of information encoded in the original probe and information such as the receive time and the receiving Worker ID. Results are streamed back to the Orchestrator, which receives streams from all Workers and in turn streams it to the CLI. At the CLI, results are stored as a single file aggregating received responses from all Workers.

4.1.3 Features. Below, we list the features of the new system and link them to the requirements from §3.

Unicast probing – Workers can perform a latency measurement using their unicast addresses (IPv4 and IPv6).

• Latency-based measurements allow for accurate GCD-based anycast detection (R1).

**Synchronized probing** – A key limitation of MAnycast<sup>2</sup> is that it performs measurements sequentially from individual VPs, leading to a significant time interval between outgoing probes to the same target. To address this limitation we use synchronized probing, where the Orchestrator instructs Workers to probe in parallel, with configurable offsets.

- Route flips in-between probes are less likely to occur when probing in parallel, since each target receives probes in a short timeframe, reducing the number of FPs (R1, R2).
- Fewer FPs translate to fewer targets to validate with a follow-up GCD-based measurement (R3).
- The measurement is performed at all Workers in parallel, rather than for each Worker individually (R5, R11).
- Probe offsets prevent bursts of probes to targets (R3).
- Rate-limiting is less likely when spacing out probes (R1).

**IPv6 support** - The new system supports IPv6 probing.

• Coverage is expanded with IPv6 censuses (R4).

Additional protocols – MAnycastR supports ICMP, TCP, and UDP/DNS probing. TCP probing uses SYN/ACK packets to high port numbers, for which we receive RST packets. Since anycast is widely used for DNS infrastructures, we implemented DNS probing by sending out A record queries. We also support TXT record queries for the CHAOS class.

4

Platform	Anycast/unicast	# of VPs	Limitations
MAnycastR production	Both	32	Few VPs for GCD
Archipelago [11]	Unicast only	Up to 180 (IPv4)	Suitable for GCD only
(Ark)		Up to 100 (IPv6)	Fewer nodes IPv6/TCP compatible

Table 1: Measurement platforms used in this work.

- Coverage is increased by supporting service-aware UDP/DNS and TCP probing (R4).
- TCP probing is done responsibly using SYN/ACK packets, which does not create state at the target host (R3).

**Failure awareness** – When a Worker disconnects during a measurement, the Orchestrator completes the measurement with the remaining Workers. Disconnecting the CLI can be used to cancel incorrect measurements.

- Outage at an anycast site will be handled by continuing measurements without this Worker (R5).
- Misconfigured measurements can be aborted, such that no unnecessary probes are sent out (R3).

**Buffering and aggregation** – At the start of a measurement, the CLI sends the hitlist to the Orchestrator, which buffers it and streams it to the Workers at the CLI-defined probing rate. Furthermore, results captured at Workers are forwarded to the Orchestrator for aggregation.

- Results are immediately submitted by Workers, reducing the impact of Worker outages on results (R5).
- Workers do not store the hitlist nor results (R10).

**Containerization** – MAnycastR supports Docker containerization, with an image size of ~130 MB.

• Docker allows for easy, OS-independent deployment. Furthermore, a small image size results in easy distribution and low storage requirements (R7, R9, R10).

**Secure inter-component communication** – MAnycastR can be deployed using an Orchestrator certificate and private key, requiring public keys at the Workers.

• Components are authenticated and communication between components is encrypted (R8).

All source code is made available under a permissive license [27] with documentation to satisfy R9. The final measurement system allows for a large range of measurements, e.g., anycast-based, GCD, and, e.g., anycast catchment measurements [14]. Furthermore, it makes a daily anycast census feasible both in terms of probing time and probing cost.

# 4.2 Measurement pipeline

In this section we describe the measurement platforms we use and our pipeline as implemented on these platforms.

4.2.1 Measurement platforms. Our pipeline makes use of two measurement platforms: MAnycastR production and CAIDA's Archipelago (Ark) measurement platform [11].

MAnycastR production is our anycast deployment that is deployed on Vultr's cloud and makes use of all its 32 sites,

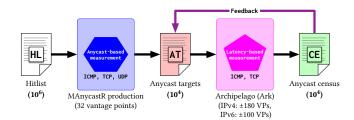


Figure 3: Overview of the measurement pipeline.

located in 19 countries on 6 continents [32]. We announce a /24 IPv4 prefix and a /48 IPv6 prefix from all sites. For latency-based GCD measurements we need a large number of geographically distributed VPs, for which we use the Ark platform. Ark has 180 VPs that allow for measurements using the scamper tool [25]. We chose Ark due to its reliability, well-documented VP locations, and good geographical coverage to allow for accurate longitudinal GCD measurements. Table 1 specifies the properties of both platforms.

4.2.2 Pipeline. Figure 3 shows the measurement pipeline for our daily anycast census. The MAnycastR system is deployed on the anycast deployment. Each anycast site runs a Worker instance, the Orchestrator instance is deployed on-premise at our institution. The CLI component used to trigger the daily census is also deployed on-premise. This setup performs anycast-based measurements for each supported protocol (ICMP, TCP, DNS) for both IPv4 and IPv6. We perform GCD measurements using ICMP and TCP, we do not use DNS due to the possible jitter introduced by DNS request processing by the target.

Given the low probing cost and feasibility, we perform the anycast-based measurements using the anycast deployment toward full hitlists (IPv4 and IPv6). This measurement leads to a set of candidate anycast prefixes, which we call anycast targets (AT). We extend these ATs to include anycast prefixes found with GCD in prior measurements (purple arrow) allowing our census to also cover rare cases where the anycast-based methodology fails to detect anycast (*i.e.*, FNs of MAnycast<sup>2</sup>). The initial set of prefixes, fed back into the measurement, is found using large-scale GCD measurements and operator ground truth, which we discuss later.

Next, using the *Ark* platform we perform GCD measurements towards the aforementioned candidate 'anycast targets'. This step allows us to enumerate and geolocate anycast sites using iGreedy's analysis approach. By performing the GCD measurement only on 'anycast targets' we significantly reduce the probing cost by two orders of magnitude, from millions to tens-of-thousands. This reduction is necessary as measuring the entire hitlist with GCD is impossible for a daily census at responsible probing rates that do not overburden the Ark platform nor the Internet.

We find that *Ark* provides sufficient geographical coverage for GCD. Whilst we acknowledge that more VPs lead to higher enumeration of anycast sites, we believe the burden and probing cost of performing GCD using even larger scale measurement platforms outweighs the value. Furthermore, fully enumerating large anycast deployments is infeasible as backed by previous work [16]. We later confirm this by sharing results of GCD measurements using 481 geographically distributed RIPE Atlas probes (§5.2).

4.2.3 Probing strategy. Our ping targets come from ISI's IPv4 hitlist [15], which ranks ping-responsive addresses per /24 prefix. For DNSv4 we merge the ISI hitlist with a set of authoritative name server IPs obtained from the OpenINTEL project [31], and we prefer name server IPs as representative addresses for a specific /24 over the ISI hitlist. We adopt this approach to maximize chances of probing an active DNS server. For IPv6 we use the IPv6 hitlist from TU Munich (TUM) [18] combined with AAAA DNS records obtained from the OpenINTEL project. Our IPv4 hitlist totals 5.9 M targets of which ~4.0 M responsive (depending on churn), and for IPv6 we have 2.5 M targets of which ~1.7 M responsive. We regularly update the hitlists to improve coverage of the census. In particular, we observe the IPv6 hitlist sources to show large increases in coverage of IPv6 address space over time.

We probe at a /24-IPv4-prefix and /48-IPv6-prefix granularity as these are likely to be the smallest prefix size to be propagated by BGP. Whilst this assumption works well in most of the cases, Schomp et al. showed that there may be rare cases of more specific partitioning [28]. For example, *hypergiants* announcing a /24 prefix at their PoPs may assign part of a prefix to a replicated service, whilst utilizing all other addresses in that prefix as unicast for individual servers within their private backbone. While we acknowledge this limitation, we continue to probe at a /24 prefix granularity to keep our scanning rate low for responsible scanning. We address and quantify this limitation in §5.6.

4.2.4 Daily output. Our daily census is finally constructed as follows. For each prefix where either the anycast-based or GCD methodology detects anycast, we publish the results to our public Git repository<sup>1</sup>. The census includes information found using both methodologies for all protocols. Furthermore, we add the estimated number of anycast sites found by each measurement, and for the latency-based measurements we include expected geolocations generated using iGreedy's city population-based geo-detection algorithm. This data can be used by the community to evaluate anycast deployments with different levels of confidence.

#### 4.3 Ethics

Our ethics review board does not require a specific ethics approval for measurements conducted according to community best practices. In this work, we follow these practices by ensuring that the address space of our measurement infrastructure has correct pointers to abuse contacts and references to a page explaining the goal of the measurements, with clear instruction for opting out. The analysis conducted in this paper was also helpful in establishing a significantly lower and manageable probing rate. This allows us to perform a responsible daily census that minimises performance impact on the targeted infrastructure.

## 5 VALIDATION

In this section we validate the performance of MAnycastR. We focus on: (§5.1) accuracy and precision, (§5.2) accuracy of site enumeration, (§5.3) the benefits of increased protocol coverage, (§5.4) replicability in other deployments, (§5.5) accuracy at low probing rates, (§5.6) prefix partitioning, (§5.7) comparison against other censuses and (§5.8) ground truth.

# 5.1 Accuracy and precision

5.1.1 Large-scale GCD measurement. As discussed in §2, GCD is highly accurate in detecting anycast. However, the probing cost for GCD measurements on the entire hitlists is too high for frequent measurements. Because of the accuracy benefits of such a measurement, we conducted GCD measurements on the entirety of both hitlists in February and December of 2024. We analyze the latter measurement run in this section. In this case we used 227 Ark nodes for ICMPv4 and 118 for ICMPv6. The number of Ark nodes is higher than our daily census, as it made use of the Ark development environment, which has more VPs. To limit impact, we performed the run at a low probing rate over a period of 22 days for IPv4 and 9 days for IPv6. For clarity, we refer to this validation measurement as  $GCD_{Ark}^3$ . This measurement provides a lower bound of the anycast deployments on the Internet, since using the GCD approach tends to underestimate the number of anycast deployments (§2.1). The results of this validation run provide insight into the number of FNs of the anycast-based measurement and will be used as a baseline for further validation. Table 2 shows the results of this measurement, compared to the candidate anycast targets found using the anycast-based approach of MAnycastR (red list marked "AT" in Figure 3).

The GCD<sub>Ark</sub> ICMPv4 measurement finds 13,692 anycast prefixes, of which 13,168 are also found by the anycast-based measurement in the MAnycastR pipeline. Ergo, the pipeline misses 524 prefixes. Of these 524 prefixes, 365 are covered by

<sup>&</sup>lt;sup>3</sup>Note: the MAnycast<sup>2</sup> paper also discusses a GCD measurement, limited to a 2% sample of prefixes to save on probing cost (§4.4 of [30]).

Protocol	Anycast-based	$GCD_{Ark}$	Anycast-based $\cap$ GCD <sub>Ark</sub>	FNs (FNR%)	$\neg GCD_{Ark}$
ICMPv4	25,396	13,692	13,168	524 (3.8%)	12,228
ICMPv6	6,315	6,221	6,006	215 (3.5%)	94

Table 2: Comparing any cast prefixes found using the any cast-based measurement to  $GCD_{Ark}$ .

our public census as they were added to the AT list after the GCD $_{Ark}$  scan in February. As for accuracy, the anycast-based approach finds 25,396 anycast targets, of which 12,228 were classified as unicast by GCD $_{Ark}$ . While these are mostly FPs of the anycast-based approach, ground truth shows these also contain TPs that we discuss later. For IPv6 we find 6,221 anycasted /48s using GCD $_{Ark}$  and 6,315 using MAnycastR and an intersection of 6,006 /48s.

This analysis shows there is value in doing large-scale latency measurements, as this covers prefixes missed by the anycast-based approach. However, the drawbacks in terms of probing cost, timing, and burdening of infrastructure outweighs the gain in visibility for daily measurements. To balance the benefits, we repeat large-scale GCD measurements bi-annually and feed the additional prefixes found into our AT list so they are covered in subsequent daily census results.

5.1.2 RIPE Atlas GCD measurement campaign. To assess whether the daily pipeline provides sufficient geographical coverage for GCD measurements, we ran an extensive measurement campaign using 481 RIPE Atlas nodes. We selected nodes with stable uptime, filtered out nodes with false user-reported locations (by verifying with both MaxMind and IP2location databases [21, 26]), and ensured that no two nodes are within 100 km of each other. We conducted the measurement towards the ICMPv4 ATs of 16 September, 2024, which consisted of 23,821 prefixes.

Comparing the results, we detect 12,186 prefixes as any-cast using *Ark* in the MAnycastR GCD measurement and 12,202 prefixes using RIPE Atlas, with an intersection of 11,953 prefixes. RIPE Atlas missed 233 prefixes found only in the MAnycastR census, mostly due to probe measurement failures. With RIPE Atlas we find 248 prefixes not found using GCD in the census. Of these, 189 belong to Imperva, a CDN that offers DDoS protection. We suspect these to be cases of temporary anycast, as we discuss later. For the remaining 59 prefixes we see cases of a single RIPE Atlas node reporting low RTT, which we suspect to be ASes blocking content and regional anycast deployments where MAnycastR lacks VPs.

Overall these results show that RIPE Atlas provides little additional coverage. Whilst we find the higher geographical coverage to increase enumeration capabilities, as we will discuss later, it incurs a significant increase in probing cost. Furthermore, even with increased spending quotas the measurement lasted three days. For these reasons we find RIPE Atlas to be unsuitable for the daily MAnycastR pipeline. We provide more detail in Appendix A

# of sites	Candidate	GCD	¬GCD	Overlap
receiving	anycast	confirmed	confirmed	(in %)
2	12,099	709	11,390	5.86%
3	602	364	238	60.47%
4	418	333	85	79.67%
5	439	378	61	86.10%
5-10	1,147	1,018	129	88.75%
10-15	848	729	119	85.97%
15-20	4,775	4,766	9	99.81%
20-25	2,822	2,818	4	99.86%
25-32	2,078	2,078	0	100.00%
Total	25,228	13,193	12,035	52.30%

Table 3: Comparing any cast-based ICMPv4 results per number of sites receiving responses with  $GCD_{Ark}$ .

5.1.3 Investigating Disagreement. As mentioned, the anycast-based approach determines anycast by counting the number of VPs receiving responses for a given target. For a VP count of one, the target is marked as unicast. For a count above one, the target is marked as anycast. However, as shown in the original MAnycast<sup>2</sup> paper, the FP case (unicast responding to multiple sites) most commonly occurs when two sites receive responses. A comparison of the anycast-based and GCD results, based on the number of sites receiving responses, is given in Table 3. We find that the vast majority of disagreement occurs when 2 VPs receive responses, followed by fewer cases for 3, 4, and 5 VPs receiving responses.

When more than 5 VPs receive responses they are mostly confirmed with GCD, those not confirmed with GCD all originate from Imperva. We suspect this is related to their ondemand DDoS mitigation service, where prefixes are anycast for short periods of time.

Investigating the total prefixes not confirmed by GCD we observe the majority (8-9k for any given day) originate from Microsoft's AS 8075. While we lack information to confirm this, we speculate, based on a 2019 study [3], that this is due to Microsoft's internal routing policies, where prefixes are announced globally to ingress traffic into their network as soon as possible, with internal routing to a unicast destination. Knowing of globally announced prefixes (*i.e.*, global BGP) that route to a single location (*i.e.*, unicast) is valuable as it can help with understanding routing properties for such networks. Using traceroute we confirm probes ingressing at distinct nearby PoPs. Future work is to include global BGP in our census and expand discovery of this practice.

5.1.4 Influence of load balancers. The MAnycast<sup>2</sup> paper ([30], §5.1) indicated that load balancers might be a cause of FPs when splitting traffic over links. Load balancers typically calculate a hash using packet headers, to determine the outgoing link. To test whether our approach – that varies the ICMP payload and checksum – triggers load balancer decisions, we performed the anycast-based measurement using

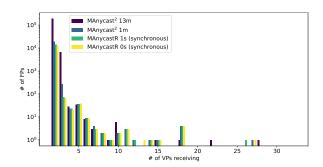


Figure 4: Comparison of MAnycast<sup>2</sup> with a 13- and 1-minute interval between probes, and MAnycastR with a 1- and 0-second interval between probes.

static probes (*i.e.*, all Workers send exactly the same ICMP Echo Request, without payload and checksum variation). The results match our regular measurement, indicating – as suggested by the literature [1] – that load balancer hashes are calculated only on flow headers. Thus, load balancers do not affect our results as we keep these headers static. This rules out load balancers as a cause of FPs, contradicting the hypothesis in the MAnycast<sup>2</sup> paper.

5.1.5 Benefits of synchronous probing. One shortcoming of MAnycast<sup>2</sup> is that it probes an entire hitlist from each VP in sequence, leading to a considerable interval between probes to the same target address. Instead, in MAnycastR we implemented synchronous probing where the Orchestrator ensures Workers send out probes to the same target one second after each other.

To compare our approach to the one used in the MAnycast<sup>2</sup> paper (where there was a 13-minute gap between probes to the same target), we performed a measurement using a 13minute interval between probes. This immediately demonstrates a shortcoming of the MAnycast<sup>2</sup> approach: as our deployment has 32 sites, a single run takes around 7 hours. We therefore also ran a second measurement using a 1-minute interval between probes, to obtain a shorter runtime. We compare this to MAnycastR measurements with inter-probe offsets of 1 second and 0 seconds. Figure 4 shows the FP rate for these experiments. It plots the number of FPs on a logarithmic y-axis against the number of VPs receiving on the x-axis. Overall, the number of FPs increases as the interprobe interval increases. Probing with a 0-second interval yields 13,312 FPs, followed by 14,506 FPs for a 1-second interval, 19,830 FPs for 1 minute, and 198,079 FPs for 13 minutes.

An increased probing interval increases the time between first and last probes to a target, leaving more time for route flips to occur. We can therefore likely attribute the rise in FPs for larger intervals to routing instability, where route flips

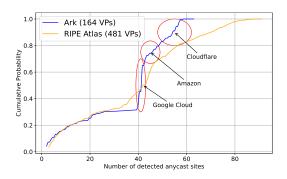


Figure 5: Number of sites detected per /24-prefix.

cause targets to respond to a different VP, causing a misclassification as anycast as stated in the MAnycast<sup>2</sup> paper [30].

While a reducing the inter-probe interval from 1 to 0 seconds leads to a reduction of 1,194 FPs, we argue the benefits, in terms of responsible probing, when spacing probes (like a regular ping) outweigh the slight decrease in FPs.

5.1.6 Longitudinal precision. To assess the precision of our daily census we evaluate results for the first 56 days, from March 21 to May 15, 2024. We focus on prefixes detected using ICMPv4 for the anycast-based and GCD approaches. On average, we observe 27.5 k prefixes with the anycastbased approach and 12.1 k prefixes with GCD daily, during this period. Taking the union of all prefixes observed with the anycast-based approach during the 56-day timeframe, we find 78,687 /24-prefixes. Of these prefixes, 15,791 /24s are consistently observed on all 56 days. Conversely 62,896 prefixes only showed up on some days. Of these we find only a few rare cases that are anycast according to GCD as well, but had downtime and/or switched to unicast during the time period. For most we observe they are prefixes not detected as anycast by GCD, and suspect many are FPs. For the GCDconfirmed prefixes we observe 12,605 prefixes detected as anycast for at least one day, 11,359 are observed every day. The 1,246 prefixes not observed every day, include regional anycast deployments that are difficult to detect with GCD, cases of suspected BGP prefix hijacking (causing FPs), and anycast deployments that had downtime.

Overall, we observe that our anycast-based set has a high variability, whereas the GCD set is much more stable, confirming that our combined approach – using anycast and GCD in MAnycastR– is important to achieve high precision.

## 5.2 Anycast site enumeration

One feature from iGreedy that we include in MAnycastR is estimating the number of replicas for an anycasted prefix. Figure 5 shows a cumulative distribution plot of the number of anycast sites detected using latency data obtained from Ark (blue) and RIPE Atlas (orange). The circles show the

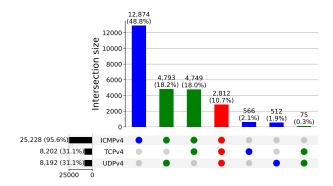


Figure 6: MAnycastR detection of anycast-based method for ICMPv4, TCPv4, and UDPv4.

enumeration counts attributed almost entirely to prefixes from three *hypergiants* (Google, Amazon and Cloudflare).

These counts are a lower bound of the actual number of anycast sites. For example, Google has presence in 103 cities [20] where we find  $\sim$ 41 sites and Cloudflare has PoPs in 300+ cities [12] where we detect  $\sim$ 54 sites. For small deployments (< 30 VPs) enumeration is near the actual number of sites which we validate using ground truth in §5.8.

Comparing RIPE Atlas with Ark, we find both have similar results for small anycast deployments (bottom left). For large deployments (top right), RIPE Atlas is able to achieve a higher enumeration of 80 compared to 60 with Ark. However, it has large variability in the number of sites detected due to inconsistency in the number of RIPE Atlas nodes participating in measurements. We provide a more elaborate comparison between these platforms in Appendix A.

Future work is to improve enumeration and geolocation data in our daily census, using, *e.g.*, traceroute [16]. However, a complete enumeration for large anycast infrastructures is likely infeasible, as also shown by Fan et al. that found 10 k vantage points achieve an 80% enumeration recall [17].

## 5.3 Increasing coverage

5.3.1 Protocol support. To extend coverage compared to iGreedy and MAnycast<sup>2</sup> we added TCP and UDP probing. Figure 6 shows a breakdown of which prefixes can be detected using our anycast-based approach with different protocols. First, the bottom left shows the totals for each protocol, *i.e.*, we detect 25,228 anycast prefixes with ICMP, 8,202 with TCP, and 8,192 with UDP. Next, below the histogram we indicate the intersection shown by each bar. We color this by three categories: **blue** for results of a single protocol, **green** for groupings of two protocols, and **red** for the intersection of all three protocols. The bars show the totals and percentage that each intersection gives, *e.g.*, the first bar shows that 12,874 prefixes (48.8% of the total) were discovered to be anycast using **only** ICMP (*i.e.*, TCP and UDP did not detect these

Protocol   Anycast targets		Anycast targets	Intersection
	our deployment	ccTLD deployment	
ICMPv4	25,324	16,208	13,912
ICMPv6	6,996	6,501	6,255

Table 4: ICMPv4 anycast-based ATs found using two distinct anycast deployments.

prefixes as anycast). Next, we have the prefixes detected as anycast by both ICMP and UDP, totaling to 4,793, etc.

These results show that ICMP probing uncovers the most anycasted prefixes, as indicated by the ICMP total (bottom-left) and the low totals for the intersections that exclude ICMP (last three bars in the histogram). Furthermore, we see an increase in coverage using TCP and UDP, with 566 prefixes that are only detected as anycast using TCP (third-to-last bar), and 512 with only UDP (second-to-last bar). Of these prefixes detectable only with UDP, we find 97 prefixes received at more than 3 VPs where the anycast-based approach has high accuracy as shown in Table 3. *E.g.*, DNS G-root (operated by the US Department of Defense), LACNIC, Oracle, eBay, several registrars, and various TLDs deploy DNS anycast that is only detectable with anycast-based UDP probing.

Prior work (see §2.3) also attempted to detect and enumerate anycast using CHAOS queries. We therefore added support for CHAOS queries. However, we find CHAOS records are often used for co-located servers at a single location (suggested by values such as 'auth1' and 'auth2'). As such, we find multiple CHAOS records to be a weak indicator of anycast. We provide a more detailed comparison on CHAOS records in Appendix C.

5.3.2 IPv6 support. To fully cover the IP anycast land-scape, we also added support for IPv6. Overall, we find fewer prefixes with the anycast-based approach for IPv6, 6,864 compared to 27,573 for IPv4. This is unsurprising as our IPv6 hitlist is much smaller. Like IPv4, most anycast candidates are detected using ICMP (6,659). Furthermore, we find a large number of anycasted /48s responsive to TCP (4,476). The higher TCP responsiveness, compared to IPv4, was expected due to the different hitlist origins. Unlike the ISI hitlist, which is based on ping-scanning, TUM's and OpenINTEL's IPv6 hitlists more likely reflect active services. For the specific distribution for IPv6 we refer to Figure 7 in the appendix.

# 5.4 Replicability

To validate if our methodology is deployment-agnostic, we deploy MAnycastR on an independent, ccTLD registry operated anycast production infrastructure, providing VPs at 12 distinct locations. Using this deployment we perform an anycast-based measurement for ICMPv4 and ICMPv6. The results are shown in Table 4. We observe that the external anycast platform finds fewer candidate anycast prefixes. Especially for IPv4, they find 16,208 IPv4 anycasted /24-prefixes compared to 25,324 prefixes found with our own deployment.

Deployment		ATs	¬GCD	(¬GCD %)	Probing cost
EU-NA	2 VPs	12,492	2,164	(15.8%)	12 M
1-per-continent	6 VPs	14,221	1,311	(9.6%)	35 M
2-per-continent	11 VPs	27,379	633	(4.6%)	65 M
ccTLD	12 VPs	16,208	632	(4.6%)	71 M
MAnycastR production	32 VPs	25,324	263	(1.9%)	188 M
GCD <sub>Ark</sub> (full hitlist)	227 VPs	13,692	0	(0.0%)	1,335 M

Table 5: The number of anycast targets (ATs), missed GCD-confirmed prefixes ( $\neg$ GCD) and probing cost for various anycast deployments and GCD<sub>Ark</sub>.

In total, our census finds 11,322 IPv4 anycast targets not found by the ccTLD deployment. In both cases, the vast majority (> 98%) are targets that are detected by only 2 VPs. Since the FP rate is high when 2 VPs receive responses, this suggests that the non-intersecting targets are largely FPs.

Of 13,912 IPv4 ATs seen on both platforms, 12,816 are confirmed using  $GCD_{Ark}$ . Additionally, ATs seen exclusively on our own or on the ccTLD platform include 481 and 112 GCD-confirmed prefixes, respectively. These latter sets of prefixes are small and difficult to detect anycast prefixes only seen by either platform due to topological differences. When taking the union of ATs, we find the anycast-based approach detects 13,409 out of 13,692 GCD $_{Ark}$  prefixes (98.0%).

For IPv6 we find 6,996 ATs with our own and 6,501 with the ccTLD platform, with an intersection of 6,255. Most nonintersecting ATs are seen at 2 VPs and not GCD-confirmed.

## 5.5 Reducing probing costs and impact

5.5.1 Reducing deployment. Measuring the entire hitlist requires 188 M probes for our daily ICMPv4 anycast-based measurement compared to GCD<sub>Ark</sub> requiring 1.3 B probes. For this reason, we perform GCD only towards ATs found with the anycast-based approach. The previous section showed that a smaller anycast deployment finds considerably fewer ATs, thereby reducing the number of targets for the follow-up GCD measurement. However, this comes at a cost of FNs (GCD-confirmed anycast prefixes missed by the anycast-based approach). This raises the question if we can reduce the anycast deployment used in the daily pipeline to reduce the number of ATs with a minimal increase in FNs.

To answer this question we performed additional anycast-based measurements. First, we pick two sites per continent, maximizing geographical distance (e.g., sites on the US East and West Coast), totaling 11 VPs (we have one site in Africa). Then, we limit ourselves to the sites that receive the most responses on each continent (6 VPs). Finally, we use only two VPs; one in North America and one in Europe. Table 5 shows the result of these measurements, including the results of our full deployment and the ccTLD deployment.

Interestingly, we observe the highest number of ATs when probing with two sites in each continent. This shows that reducing the number of VPs does not necessarily reduce the number of ATs. Despite the increase in ATs, this deployment has a lower recall due to an increase of 370 FNs compared to our regular anycast-based measurement. When probing with one site per continent, we find 14,221 ATs. However, this comes at a cost of 1,311 missed anycast prefixes. Finally, when probing with only two VPs we find the fewest ATs (12,492) at the cost of the most FNs (2,164). This also shows that even with only two VPs, 84% of GCD-confirmed prefixes are detected, which is unsurprising considering most anycast deployments have a global presence with at least one site in North America and Europe. The FNs of each measurement largely consist of small anycast deployments that are confined to small geographical areas (*i.e.*, regional anycast).

We argue it is impossible to determine an optimal number of sites, as the value of each VP highly depends on how it connects to the wider Internet. We also argue that, despite a slightly higher probing cost, our current deployment has merit as it uncovers most difficult to detect regional anycast.

5.5.2 Reducing probing rate. A key strength of MAny-castR is a low probing cost and hence a responsible measurement of IP anycast. To illustrate this, we test another feature of MAnycastR. We can configure MAnycastR to go over the hitlist at a specific rate, while maintaining one-second intervals between probes from each Worker. To validate if we can maintain accuracy at a reduced probing rate, we perform a MAnycastR census using a probing rate of 1/8th the 'normal' rate of our daily census. Even at this much-reduced rate, MAnycastR detects the same number of anycast targets.

## 5.6 Anycast prefix size

As mentioned in §4, our methodology scans at a /24 (and /48 for IPv6) granularity as it is the smallest prefix size propagated by BGP. However, BGP announcements seen by route collectors are often less specific. To assess whether such less specific prefixes are anycast in their entirety, we look at all announced IPv4 prefixes in which we detect anycast.

This comparison is done using data from April 27, 2024, where our census finds 12,046 GCD-confirmed anycast /24-prefixes. Using CAIDA's prefix2as dataset [7] we find those /24s to be part of 4,184 BGP announced prefixes. Out of these BGP prefixes 3,827 are entirely anycast (*i.e.*, each contained /24-prefix is anycast). For 70 prefixes there is uncertainty as they contain at least one /24 that is unresponsive for the measurement. The remaining 287 announced prefixes contain unicast /24-prefixes according to our measurement.

This shows scanning at /24 granularity is necessary to avoid overestimating anycast. However, routing may yet be different within such prefixes [28] . This limitation was evident when validating results with *NTT DATA Global IP Network*, a large tier-1 transit provider, who announce their address space at multiple PoPs (*i.e.*, global BGP). However,

their addresses point to single servers (*i.e.*, unicast). The exception being 6 addresses (in a single /24) replicated at all PoPs for their public DNS resolver (*i.e.*, anycast). The census, which probed a unicast address for this prefix, falsely classified the entire /24 as unicast. We call such cases, where a /24 contains both unicast and anycast, partial anycast.

To assess the number of partial anycast prefixes, we perform a GCD measurement at /32 granularity targeting the entire allocated IPv4 address space, totaling four billion targets. We do this using nine VPs spanning multiple continents, at a low probing rate, and over a period of ten days.

While GCD measurements require a large number of VPs for accurate detection, we believe in this case a few VPs suffice for detecting partial anycast as it requires a private backbone with global reach to allow for different routing topologies within an announced prefix.

This scan revealed anycast in 13.4k /24s, of which 1,483 /24s contain partial anycast (including the aforementioned tier-1 prefix). Following this scan we updated our AT list and included a partial anycast flag in our census to avoid overestimating anycast. 1,178 of these partial anycast prefixes are consistently found in our census. We observe most partial anycast prefixes are announced by large CDNs that have global presence. For 305 partial anycast prefixes we find the prefix is entirely unicast the following day. These prefixes mostly belong to Imperva, which we observe to announce temporary anycast in our RIPE Atlas measurements and longitudinal census data. We suspect this is related to their anti-DDoS services. This motivates our decision to measure any cast longitudinally. Future work is to capture short-lived anycast by triggering measurements based on, e.g., BGP announcements observed by route collectors.

## 5.7 Comparing with external sources

There are also external datasets that contain information on anycast. We compare our work against two, a commercial dataset from *IPInfo* and a public dataset from *BGPTools*.

**IPInfo**, a commercial provider of IP address databases, shared their list of detected anycast prefixes with us [22].

First, for IPv6 our census detects anycast in 6.3k /48s whereas IPInfo detects anycast in 2.0k /48s. For IPv4, our census detects anycast in 13.4k /24s and *IPInfo* in 14.0k /24s. Overall, 12.6k /24s are found in both datasets showing high agreement. Inspecting the 0.8k prefixes found only in our census, we find most are detected in few locations and are regional. For the 1.4k prefixes found only by *IPInfo*, we observe 0.6k originate from Imperva and several hundreds from other CDNs offering anti-DDoS services where we suspect temporary anycast. We believe *IPInfo* to include larger numbers of temporary anycast as they take weekly snapshots.

BGPTools also uses an anycast-based approach similar to the first stage of MAnycastR. However, there are two key differences: 1) if BGPTools detects a single address in an announced prefix as anycast, they assume the entire prefix is anycast [4], and 2) they do not use GCD to filter out FPs. We compare BGPTools to MAnycastR for December 20, 2024. On that day, BGPTools marks 3,047 announced prefixes as anycast. Comparing this to MAnycastR, of those 3,047 prefixes our anycast-based stage marks 2,954 as anycast. Of these, 228 prefixes are only detected at two of our VPs, and our GCD stage marks these as FPs, showing BGPTools likely significantly overestimates the number of prefixes that use anycast. Our census finds 13,495 anycast /24-prefixes confirmed by GCD in total for that day. Of these, 9,739 are also covered by the BGPTools prefixes, leaving 3,756 /24-prefixes they miss.

Investigating the prefixes *BGPTools* classifies as anycast, we find 467 are less-specific than a /24. Comparing this to our census, we find that 60 of these prefixes contain a grand total of 8,038 unicast /24s, illustrating the risk of *BGPTools* assumption that entire announced prefixes are anycast.

Looking at IPv6, *BGPTools* detects 1,148 IPv6 prefixes, 1,131 of which are covered by our census while 8 are missed as they are not in our hitlist. That same day we find 6,358 anycast /48s, 1,479 of which were not found by *BGPTools*.

These results reaffirm that IP anycast cannot be generalized to BGP prefixes, that our census has better coverage, and that our pipeline obtains a higher TP rate.

#### 5.8 Ground truth validation

Finally, we validate against public datasets and reach out to operators to obtain ground truth for our census results.

5.8.1 DNS operators. We find all IPv4 and IPv6 prefixes for the Quad9 public resolver, RIPE authoritative nameservers, and all DNS root servers correctly classified as anycast by our census. Interestingly, we observe that G-root is unresponsive to ICMP and TCP for both IPv4 and IPv6. However, MAnycastR is able to detect it using DNS over UDP probing.

Next, we reached out to multiple ccTLD operators as they often deploy regional anycast that is difficult to detect. These include .be, .cz, .de, .dk, .nl, .nz, .ua. The census is able to detect the vast majority of anycast nameservers. The exceptions are two anycast deployments regional to Belgium and the Netherlands not found with GCD, of which one is detected with the anycast-based approach, 3 nameservers regional to New Zealand, one nameserver local to Germany, and 2 IPv6 prefixes not covered by our hitlist. Furthermore, we observe our GCD reported locations closely match reality, exceptions being multiple sites in a single city or nearby cities (e.g., Prague, Bratislava, Vienna) being detected as a single site. Finally, a few anycast operators expanded their deployment during the census which is visible in our longitudinal data.

AS	Organization	IPv4 (/24)		IPv6 (/48)	
396982	Google Cloud	3,627	(1st)	5	
13335	Cloudflare	3,133	(2nd)	284	(3rd)
16509	Amazon	1,286	(3rd)	120	
54113	Fastly	435	(4th)	65	
209242	Cloudflare Spectrum	289	(5th)	3,338	(1st)
19551	Incapsula (Imperva)	2		352	(2nd)
12041	Afilias	221		222	(4th)
44273	GoDaddy	32		122	(5th)

Table 6: Largest ASes originating anycast prefixes.

5.8.2 Hypergiants. Table 6 shows the largest ASes based on the number of IPv4 /24 and IPv6 /48 anycast prefixes they originate. We find that Google Cloud is leading in IPv4 anycast followed by Cloudflare, Amazon, and Fastly. These ASes are large CDNs, often referred to as hypergiants. We color the table based on ground truth validation; green for operator-confirmed ground truth (no FPs), dark-green when fully accurate (no FPs and FNs), and orange when confirmed with public data. Due to the dominance of hypergiants, these results comprise 59% of our IPv4 and 63% of our IPv6 census.

Google provides an *ipranges* dataset [19] that discloses where prefix ranges are announced for Google Cloud resources. We compared our census to this dataset for January 6, 2025. Google's dataset contains 33 globally announced IPv4 prefixes ranging from a /16 to a /24, totaling 3,581 /24-prefixes. Out of these prefixes, 3,572 are detected as anycast in our census. For the 9 missing prefixes, 8 are not in our hitlist and 1 we detect as unicast (which we confirm to be unicast using traceroute). The remaining Google Cloud prefixes we detect as anycast are not listed in the *ipranges* dataset.

Cloudflare, the CDN that ranks first for IPv6 and second for IPv4 anycast in the census, shared ground truth data with us. Their data confirms our census is fully accurate for IPv4 anycast, where we have no FPs and no FNs. For IPv6 they shared we have no FPs, however, our census misses some IPv6 prefixes due to IPv6 hitlist limitations.

Next, we evaluate the prefixes we detect as anycast for Amazon using an *ip-ranges* dataset [2] similar to Google. The list contains 37k /24 prefixes listed as globally announced, these are cases of global announcements but not necessarily anycast as discussed previously. This list includes partitions more specific than /24, even /32 partitions. For the 1,286 /24-prefixes we detect as anycast, 1,123 are listed as being announced globally, 2 are listed as being announced in the east coast of the US, and the remaining 161 are not listed.

We also reached out to Fastly whose prefixes feature prominently in our dataset and obtained ground truth data. Furthermore, they disclosed a traffic engineering implementation that uses *backing anycast*, where a large prefix is announced globally (anycast) and more specific prefixes within route to a single PoP (unicast), which allows for withdrawing the

unicast announcement to fall back on the less-specific anycast announcement. Such traffic engineering allows for rapid redistribution of traffic, e.g., if a unicast prefix announced at a single PoP is the victim of a DDoS attack. For IPv4 Fastly confirmed all /24-prefixes we find are anycast. However, for IPv6 we have a considerable number of misclassifications. After troubleshooting, we discovered that two Ark nodes are located in ASes that miss /48 announcements for the CDN due to these ASes filtering these out or their upstream ASes not propagating the announcements. In the case of these FPs the probed addresses are unicast using a /48 with a less specific backing anycast prefix (the TE approach mentioned above). Due to these nodes being unaware of the /48, they are routed to the nearest PoP, the address is then misclassified as anycast. Filtering out these Ark nodes from our IPv6 results, we find all detected IPv6 anycast to match ground truth.

These findings indicate our methodology can detect ASes that, *e.g.*, do not propagate /48-prefixes, by performing MAny-castR measurements where we announce a specific prefix at a single site. Finally, the TE practice used to switch prefixes from unicast to anycast when a PoP is overloaded motivates the need for a daily census.

## 5.9 Lessons learned

We find the performance of both anycast-based and GCD measurements depend on the number, and geographical and topological diversity of sites. Since the majority of anycast is from *hypergiants* with global deployments, we observe that few nodes on different continents can detect the majority of anycast, as also exemplified by *BGPTools* [5]. However, for detecting regional anycast a larger measurement platform is required. Furthermore, enumeration and geolocation of anycast sites requires a well-distributed measurement platform.

We find our census provides good coverage of the anycast landscape, as confirmed by extensive validation. However, for IPv6 we are restricted by hitlist coverage. Additionally, for enumeration we find our census provides a lower bound as latency-based measurements cannot differentiate between anycast sites in near-proximity.

When using our census, we advise to keep the FP limitation of the anycast-based approach in mind, especially when only two VPs receive responses. As for GCD, we find it is highly accurate but has rare cases of FNs due to regional anycast.

We encourage operators to run their own census using the MAnycastR tool, which supports both anycast- and latency-based measurements. Furthermore, our anycast-based target lists are made available in our public census, making it possible to combine them with other AT lists to improve coverage.

#### 6 CONCLUSIONS AND FUTURE WORK

In this work we presented MAnycast Reloaded (MAnycastR), a clean slate reimplementation inspired by the MAnycast<sup>2</sup> anycast measurement approach of Sommese et al. [30] and by the Great Circle Distance (GCD) approach pioneered by Cicalese et al. with iGreedy [9]. Through extensive validation we show that MAnycastR allows for an efficient, accurate and precise daily anycast census for both IPv4 and IPv6 and with support for transport layer probing using UDP/DNS and TCP. We release this daily census to the community through a public Git repository<sup>1</sup> and release the entire MAnycastR toolchain under a permissive open source licence<sup>4</sup>.

We intend to further extend MAnycastR in future work. Specifically, we want to check responsiveness from a single VP before probing from all VPs, add support for a canary anycast deployment to detect outages, trigger-based detection of temporary anycast – *e.g.*, from BGP route collectors, add GCD support using UDP and more. We are also planning to analyse longitudinal anycast dynamics and work in which we use MAnycastR to detect suspected BGP hijacking.

## **REFERENCES**

- [1] Rafael Almeida, ítalo Cunha, Renata Teixeira, Darryl Veitch, and Christophe Diot. 2020. Classification of Load Balancing in the Internet. In IEEE INFOCOM 2020 - IEEE Conference on Computer Communications. 1987–1996. https://doi.org/10.1109/INFOCOM41043.2020.9155387
- [2] Amazon. 2025. Amazon IP address ranges. https://docs.aws.amazon. com/vpc/latest/userguide/aws-ip-ranges.html. (2025). [Accessed 06-01-2025].
- [3] Todd Arnold, Matt Calder, Italo Cunha, Arpit Gupta, Harsha V. Madhyastha, Michael Schapira, and Ethan Katz-Bassett. 2019. Beating BGP is Harder than we Thought. In Proceedings of the 18th ACM Workshop on Hot Topics in Networks (HotNets '19). Association for Computing Machinery, New York, NY, USA, 9–16. https://doi.org/10.1145/3365609. 3365865
- [4] BGPtools. 2024. GitHub issue discussing coverage of BGPTools census. https://github.com/bgptools/anycast-prefixes/issues/1. (2024). [Accessed 09-01-2025].
- [5] BGPTools. 2025. How bgp.tools detects anycast addresses. https://bgp.tools/kb/anycatch. (2025). [Accessed 06-01-2025].
- [6] Rui Bian, Shuai Hao, Haining Wang, Amogh Dhamdere, Alberto Dainotti, and Chase Cotton. 2019. Towards passive analysis of anycast in global routing: unintended impact of remote peering. SIGCOMM Comput. Commun. Rev. 49, 3 (Nov. 2019), 18–25. https://doi.org/10.1145/3371927.3371930
- [7] CAIDA. 2024. Routeviews Prefix to AS mappings Dataset (pfx2as) for IPv4 and IPv6 caida.org. https://caida.org/catalog/datasets/routeviews-prefix2as. (2024). [Accessed 19-12-2024].
- [8] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, and Jitendra Padhye. 2015. Analyzing the Performance of an Anycast CDN. In Proceedings of the 2015 Internet Measurement Conference (IMC '15). Association for Computing Machinery, New York, NY, USA, 531–537. https://doi.org/10.1145/2815675.2815717

- [9] Danilo Cicalese, Diana Joumblatt, Dario Rossi, Marc-Olivier Buob, Jordan Augé, and Timur Friedman. 2015. A fistful of pings: Accurate and lightweight anycast enumeration and geolocation. In 2015 IEEE Conference on Computer Communications (INFOCOM). 2776–2784. https://doi.org/10.1109/INFOCOM.2015.7218670
- [10] Danilo Cicalese and Dario Rossi. 2018. A longitudinal study of IP Anycast. SIGCOMM Comput. Commun. Rev. 48, 1 (April 2018), 10–18. https://doi.org/10.1145/3211852.3211855
- [11] Kimberly Claffy, Young Hyun, Ken Keys, Marina Fomenkov, and Dmitri Krioukov. 2009. Internet Mapping: From Art to Science. In 2009 Cybersecurity Applications & Technology Conference for Homeland Security. 205–211. https://doi.org/10.1109/CATCH.2009.38
- [12] Cloudflare. 2025. Network edge locations. https://cloudflare.com/network. (2025). [Accessed 06-01-2025].
- [13] Wouter B. de Vries, Salmān Aljammāz, and Roland van Rijswijk-Deij. 2020. Global-Scale Anycast Network Management with Verfploeter. In NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium. 1–9. https://doi.org/10.1109/NOMS47738.2020.9110449
- [14] Wouter B. de Vries, Ricardo de O. Schmidt, Wes Hardaker, John Heidemann, Pieter-Tjerk de Boer, and Aiko Pras. 2017. Broad and load-aware anycast mapping with verfploeter. In *Proceedings of the 2017 Internet Measurement Conference (IMC '17)*. Association for Computing Machinery, New York, NY, USA, 477–488. https://doi.org/10.1145/3131365.3131371
- [15] Xun Fan and John Heidemann. 2010. Selecting representative IP addresses for internet topology studies. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC '10)*. Association for Computing Machinery, New York, NY, USA, 411–423. https://doi.org/10.1145/1879141.1879195
- [16] Xun Fan, John Heidemann, and Ramesh Govindan. 2011. Identifying and Characterizing Anycast in the Domain Name System. *Technical Report* (2011).
- [17] Xun Fan, John Heidemann, and Ramesh Govindan. 2013. Evaluating anycast in the domain name system. In 2013 Proceedings IEEE INFO-COM. 1681–1689. https://doi.org/10.1109/INFCOM.2013.6566965
- [18] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D. Strowes, Luuk Hendriks, and Georg Carle. 2018. Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists. In Proceedings of the Internet Measurement Conference 2018 (IMC '18). Association for Computing Machinery, New York, NY, USA, 364–378. https://doi.org/10.1145/3278532.3278564
- [19] Google. 2025. Google IP address ranges. https://support.google.com/ a/answer/10026322. (2025). [Accessed 06-01-2025].
- [20] Google. 2025. Network edge locations. https://cloud.google.com/vpc/docs/edge-locations. (2025). [Accessed 06-01-2025].
- [21] IP2Location. 2025. IP Address to IP Location and Proxy Information. https://ip2location.com. (2025). [Accessed 08-01-2025].
- [22] ipinfo.io. 2025. Trusted IP Data Provider, from IPv6 to IPv4. https://ipinfo.io. (2025). [Accessed 08-01-2025].
- [23] Thomas Koch, Ethan Katz-Bassett, John Heidemann, Matt Calder, Calvin Ardi, and Ke Li. 2021. Anycast In context: a tale of two systems. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference (SIG-COMM '21)*. Association for Computing Machinery, New York, NY, USA, 398–417. https://doi.org/10.1145/3452296.3472891
- [24] Kurt Erik Lindqvist and Joe Abley. 2006. Operation of Anycast Services. RFC 4786. (Dec. 2006). https://doi.org/10.17487/RFC4786
- [25] Matthew Luckie. 2010. Scamper: a scalable and extensible packet prober for active measurement of the internet. In *Proceedings of the* 10th ACM SIGCOMM Conference on Internet Measurement (IMC '10). Association for Computing Machinery, New York, NY, USA, 239–245. https://doi.org/10.1145/1879141.1879171

<sup>&</sup>lt;sup>4</sup>LOCATION ANONYMISED FOR REVIEW

- [26] MaxMind. 2025. Industry leading IP Geolocation and Online Fraud Prevention. https://maxmind.com. (2025). [Accessed 08-01-2025].
- [27] Mozilla. 2025. Mozilla Public License, version 2.0 mozilla.org. https://www.mozilla.org/en-US/MPL/2.0/. (2025). [Accessed 30-01-2025].
- [28] Kyle Schomp and Rami Al-Dalky. 2020. Partitioning the internet using Anycast catchments. SIGCOMM Comput. Commun. Rev. 50, 4 (Oct. 2020), 3–9. https://doi.org/10.1145/3431832.3431834
- [29] Raffaele Sommese, Gautam Akiwate, Mattijs Jonker, Giovane C Moura, Marco Davids, Roland van Rijswijk-Deij, Geoffrey M Voelker, Stefan Savage, Anna Sperotto, et al. 2021. Characterization of anycast adoption in the DNS authoritative infrastructure. In Network Traffic Measurement and Analysis Conference (TMA'21).
- [30] Raffaele Sommese, Leandro Bertholdo, Gautam Akiwate, Mattijs Jonker, Roland van Rijswijk-Deij, Alberto Dainotti, KC Claffy, and Anna Sperotto. 2020. MAnycast2: Using Anycast to Measure Anycast. In Proceedings of the ACM Internet Measurement Conference (IMC '20). Association for Computing Machinery, New York, NY, USA, 456–463. https://doi.org/10.1145/3419394.3423646
- [31] Roland van Rijswijk-Deij, Mattijs Jonker, Anna Sperotto, and Aiko Pras. 2016. A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements. *IEEE Journal on Selected Areas in Communications* 34, 6 (2016), 1877–1888. https://doi.org/10.1109/JSAC. 2016.2558918
- [32] Vultr. 2024. Vultr Datacenter Locations. https://vultr.com/features/datacenter-locations. (2024). [Accessed 19-12-2024].
- [33] S. Woolf and D. Conrad. 2007. Requirements for a Mechanism Identifying a Name Server Instance. RFC 4892. RFC Editor. https://rfc-editor.org/ rfc/rfc4892.html

#### A GCD USING RIPE ATLAS

We performed GCD using RIPE Atlas for the 23,821 ATs of 16 September, 2024. This measurement lasted three days due to rate-limiting, even with increased spending quotas. Furthermore, it incurred a cost of 37 million RIPE Atlas credits which would take a volunteer hosting a RIPE Atlas probe approximately five years to accumulate. Finally, results show inconsistency in detection and enumeration of anycast due to non-guaranteed availability of RIPE Atlas VPs. For these reasons we find RIPE Atlas unsuitable for the daily census.

However, as discussed in §5.2 we do find that RIPE Atlas achieves better enumeration. However, this comes at a price in terms of probing cost. This can be seen in Figure 8, where we plot the number of PoPs detected for a Cloudflare prefix, with presence in 300+ cities, and the probing cost when decreasing the inter-node distance. Our RIPE Atlas scan was performed using 481 VPs with at least 100km inter-VP distances to maximize geographical coverage. By increasing the inter-VP distance up to 1,000km we reduce the number of VPs. We find that there is a linear increase in enumeration capabilities, whereas the probing cost shows an exponential increase.

#### B INCREASING GCD COVERAGE

As mentioned, the  $GCD_{Ark}$  measurement used more VPs as it was performed in the development environment of Ark. The regular GCD measurement targeting the ATs uses 163 VPs, whereas the  $GCD_{Ark}$  measurement targeting the full hitlist used an additional 64 VPs totaling to 227. Figure 9 plots the number of sites detected behind anycast prefixes for the different number of VPs. We find that the additional VPs increases enumeration capabilities from  $\sim$ 55 to  $\sim$ 65, an increase of 18%. The probing cost, which is linear with the number of VPs, increases 39%. Unlike RIPE Atlas, we observe

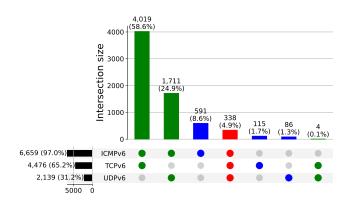


Figure 7: MAnycastR detection of anycast candidates for ICMPv6, TCPv6, and UDPv6.

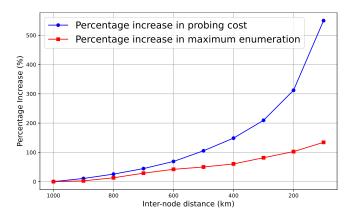


Figure 8: Percentage increase in probing cost and enumeration capability when decreasing inter-node distance.

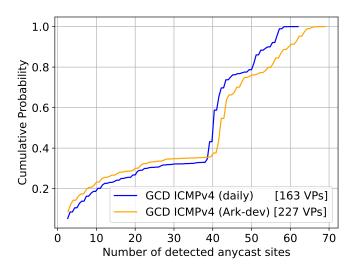


Figure 9: Plotting enumeration capabilities of the current MAnycastR GCD measurement and that with the development *Ark* nodes.

that results remain consistent and daily measuring of ATs remains feasible despite the increase in VPs.

As the *Ark* platform grows, we expect our census to increase performance in detection, enumeration, and geolocation of anycast using the GCD approach. Future work is to selectively use Ark VPs based on geographical coverage, similar to how we selected RIPE Atlas VPs, to reduce the probing cost of our measurement.

# C MEASURING ANYCAST WITH CHAOS

## C.1 Detection

As mentioned in §2.3, prior work attempted to detect and enumerate anycast using CHAOS queries. We build on this

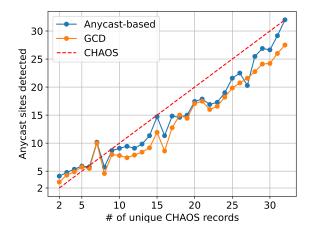


Figure 10: Enumeration counts for the three methodologies: CHAOS, anycast-based, and GCD-based, plotted against the number of unique CHAOS record founds, for IPv4 name servers.

prior work and leverage CHAOS queries as an opportunity to perform a side-by-side comparison in isolation of the anycast-based approach in MAnycastR and the built-in GCD approach. We take our nameserver hitlist as a starting point. We send CHAOS queries to each target from each of our 32 VPs according to RFC 4892 [33] to the IPv4 and IPv6 addresses on the hitlist and record the responses. Then we run a separate synchronised anycast-based measurement with 1 second offsets against the hitlist as well as a separate GCD-based measurement. In other words: we run both the anycast-based and GCD measurement toward all addresses using the MAnycastR deployment. Of the 161 K nameservers probed, we detect 2,762 to be anycast using the anycast-based approach of which 2,371 are also found using GCD.

We also observe 414 nameservers that are unresponsive to ICMP and TCP, hence the GCD method cannot detect them. 50 of these prefixes are detected to be anycast using the anycast-based method, we observe they give strong indication of being TPs as they are captured at > 3 VPs. These 50 include prefixes from well-known operators, such as eBay and Oracle, highlighting the utility of the added protocol support in MAnycastR.

# C.2 Enumeration

As mentioned, the MAnycastR tool has built-in support for both the anycast-based approach and the GCD approach. Using the MAnycastR deployment, with 32 VPs, we measure the site enumeration capabilities of both approaches when using the same VPs. We target nameservers that support CHAOS records, and measure the number of distinct CHAOS

Prefix size	Occurrence	Anycast	Unicast	Unresponsive
/11	1	1,026	1,845	5,321
/13	1	256	0	1,792
/14	7	841	3,844	2,483
/15	2	84	372	568
/16	16	976	1,262	1,858
/17	5	273	211	156
/18	4	135	54	67
/19	9	70	87	131
/20	221	3,378	33	125
/21	16	65	32	31
/22	51	175	8	21
/23	134	213	21	34
/24	2,580	2,247	269	64
Total	3,047	9,739	8,038	12,651

Table 7: BGP prefixes classified as anycast by *BGPTools* grouped by size and its occurrence. We count the number of anycast, unicast, and unresponsive /24s according to GCD.

responsive values observed under the assumption that different values indicate different sites, Figure 10 plots the name server IPv4 prefixes by number of unique CHAOS records observed (x-axis) against the site counts found with the different methodologies (y-axis). We observe two things. First, for low numbers of distinct CHAOS records, both the anycast- and GCD-based approaches estimate a slightly higher number of sites. We do not have a firm explanation for this but speculate that this maybe a case of name server infrastructures where CHAOS records are used to differentiate between colocated servers at a single location (values such as 'auth1' and 'auth2' suggest this), but not between anycast sites. Second, we observe that the anycast-based approach consistently approximates the assumed CHAOS "ground truth" more closely suggesting the anycast-based approach may be slightly better at estimating a lower bound for the number of anycast

sites, especially for cases where the anycast-based approach receives responses at many vantage points.

## **D** EXTERNAL DATASET

As mentioned in §5.7, *BGPTools* uses the anycast-based approach to detect anycast. When an address is classified as anycast they assume the entire BGP prefix announcing this address is anycast. In this section we show this assumption does not hold by comparing both daily censuses using data from the 20th of December 2024.

In Table 7 we show the BGP prefixes detected as anycast in the *BGPTools* census grouped by size. We see prefixes range from /11 (that contains 8,192 /24s) to /24 in size. Next, we list the occurrence of each prefix size, we observe the most occurring prefix size is /24 followed with /20. For these prefix sizes we list the number of anycast, unicast, and unresponsive /24s found in our census using GCD.

First, for the 2,580 BGPTools /24-prefixes we find 2,247 are GCD-confirmed, 269 not GCD-confirmed, and 64 are unresponsive. We suspect the 269 not GCD-confirmed prefixes are largely FPs of the anycast-based approach. Looking at the total, we observe BGPTools classifies 3,047 BGP prefixes as anycast that include 9,739 GCD-confirmed anycast /24s; 8,038 unicast /24s; and 12,651 unresponsive /24s. For the single /13 prefix we count 256 anycast /24s. Investigating this prefix we observe it contains a /16 that is entirely anycast, whereas the remainder of the /13 is ICMP unresponsive. Investigating the 7 /14s, that have the most unicast /24s, we find 6 are prefixes announced by Google Cloud. Using the aforementioned Google ipranges dataset, we confirm these unicast /24s are not listed as being announced globally. These results reaffirm that GCD confirmation is necessary to filter out FPs of the anycast-based approach and that their assumption to classify entire BGP prefixes as anycast is wrong.