Anycast tooling:

For operators and researchers

GMI-AIMS-4, MADRID, SPAIN

7 NOV, 2024

Remi Hendriks

UNIVERSITY OF TWENTE.



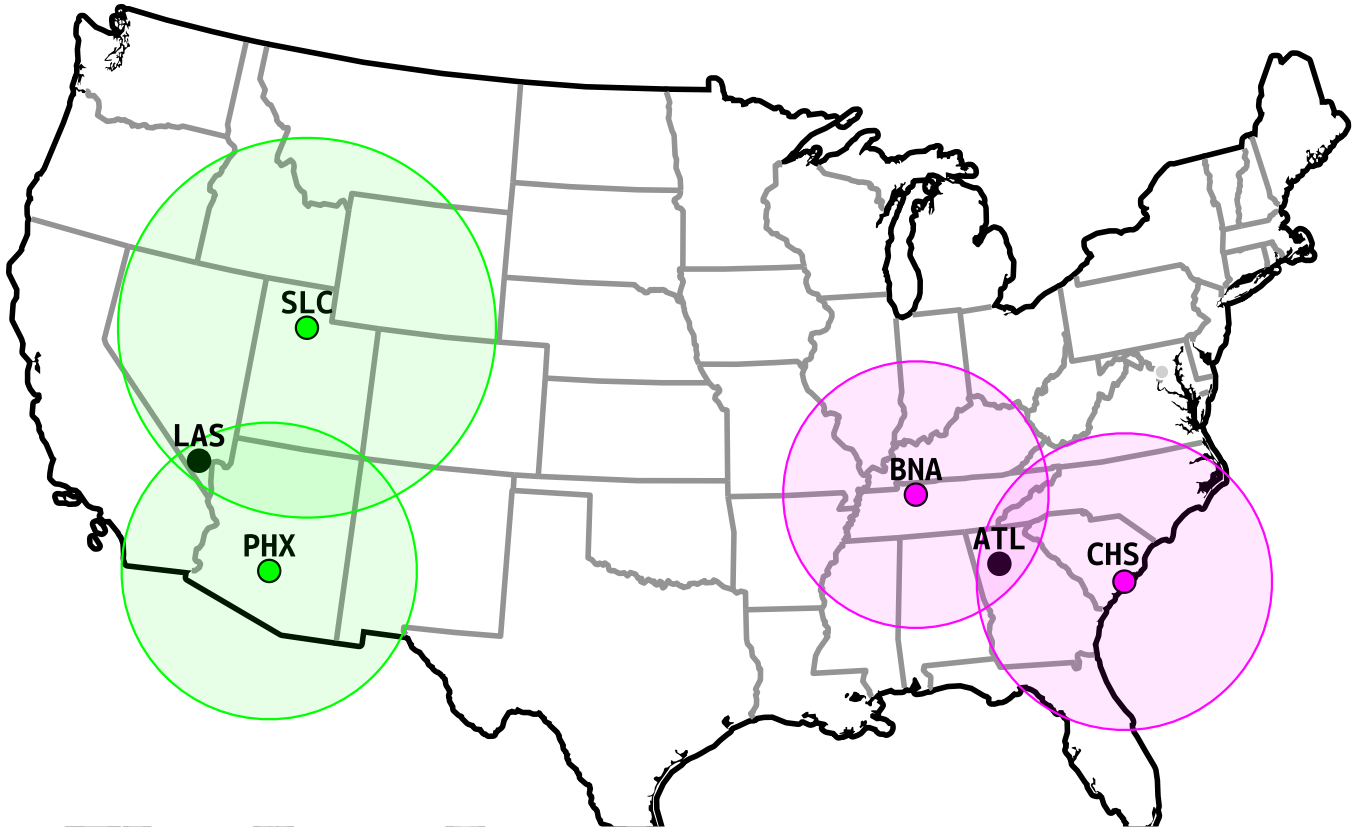


Previously

- Presented daily anycast census
  - Uses a latency-based approach
  - And an anycast-based approach

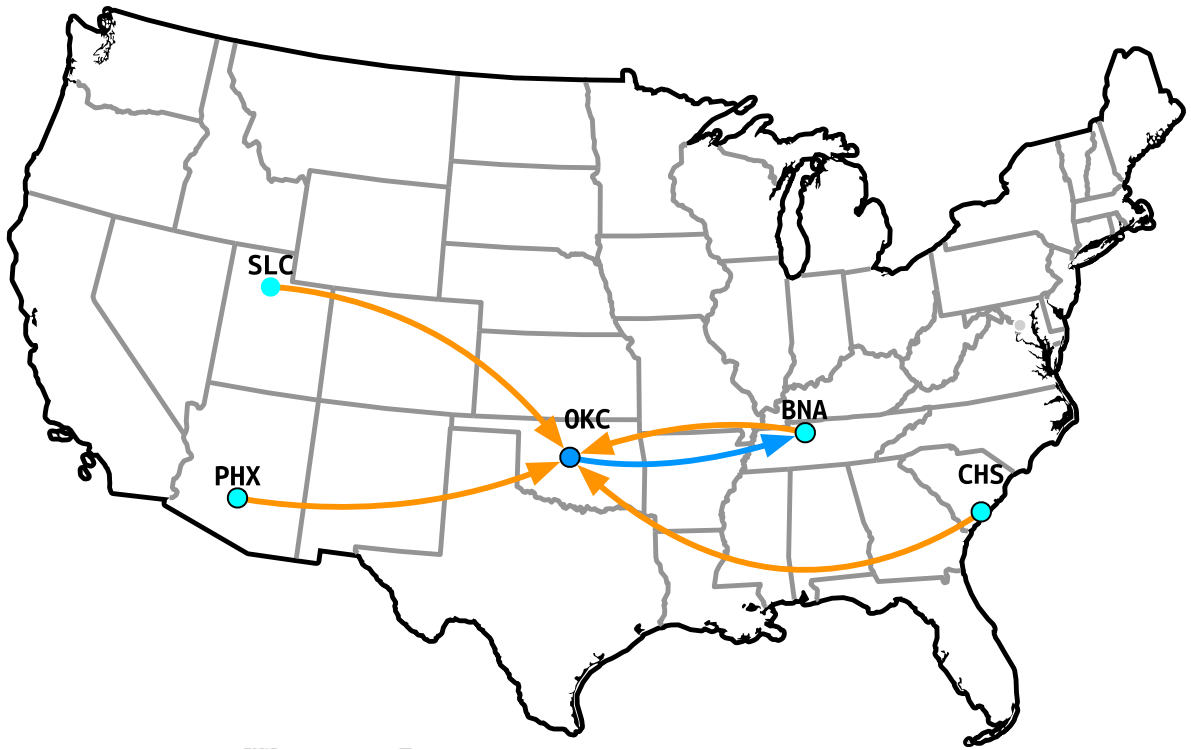
Short recap
Latency-based

- Platforms
  - Archipelago
  - Our anycast testbed (32 VPs)
- Implemented with Scamper
  - Running on testbed

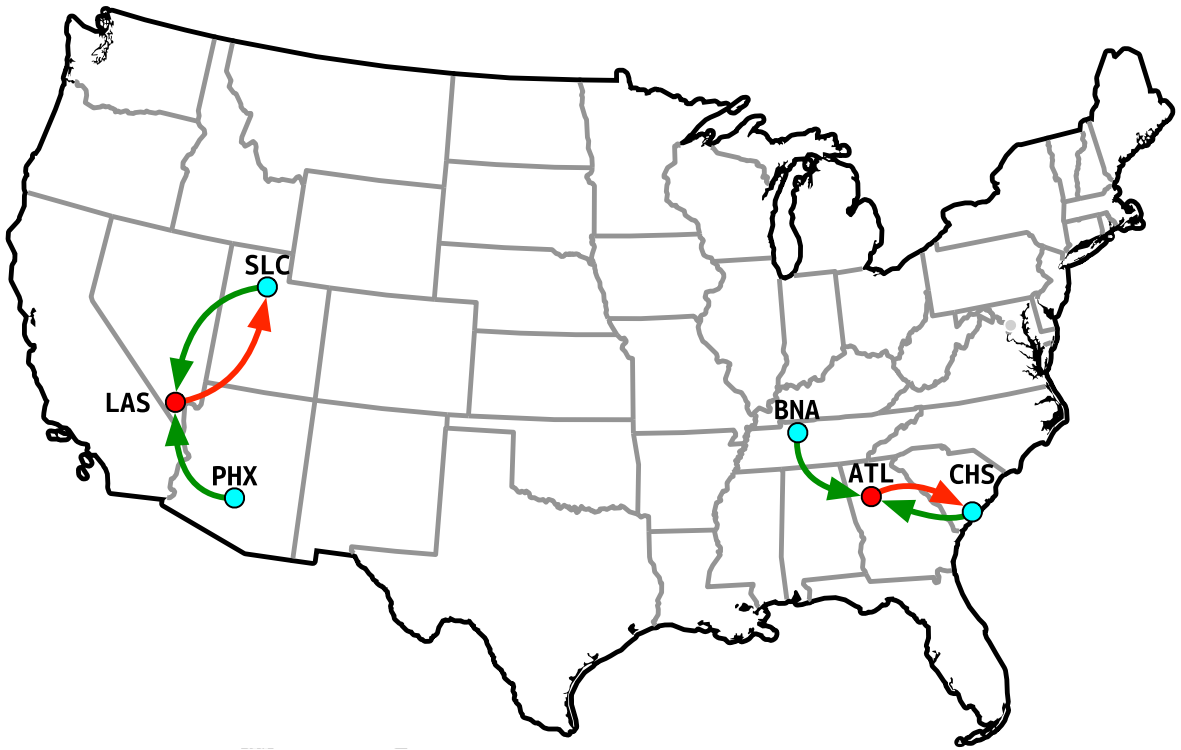


Short recap
Anycast-based

Unicast



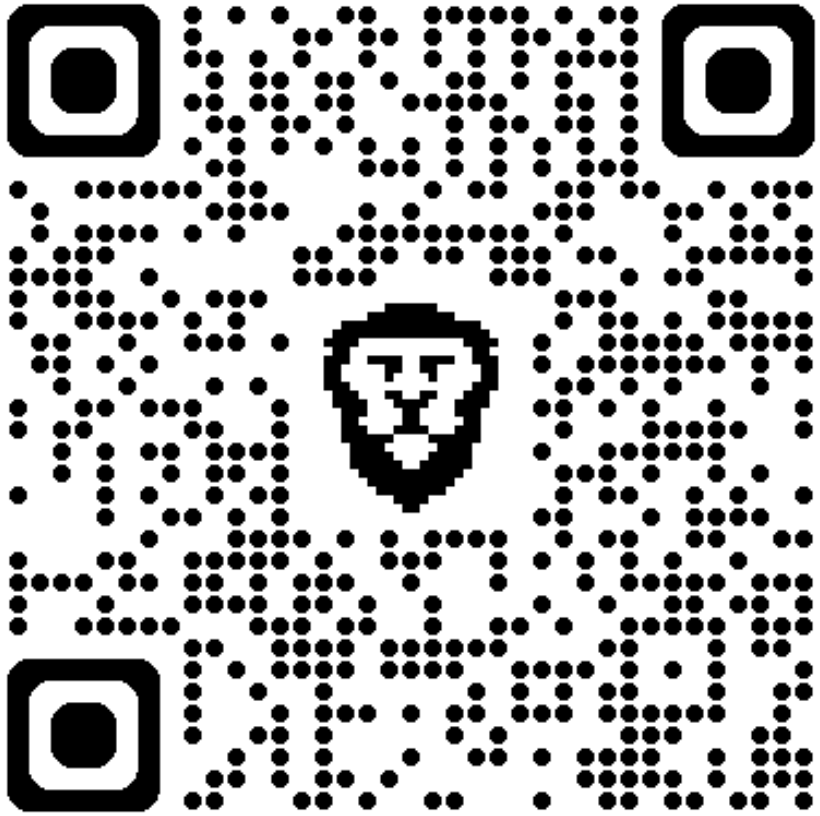
Anycast



Census update

- Census now publicly available
  - With 7 months of daily data
- User-friendly dashboard in development
- Resubmitting to Conext

- Why not bgp.tools ?
  - We also provide enumeration and geolocation
  - Their dataset uses only anycast-based approach
    - (With only 3 sites)



github.com/anycast-census/anycast-census

Now

- Anycast measurement tool (significant improvement of Verfploeter)
  - Developed for census
  - Expanded to measure anycast internally
- Used at Cloudflare (largest anycast platform we know of)
  - Own version, developed by author of Verfploeter paper
- Currently used at SIDN (.nl operator)
  - Our version
  - Active collaboration
- Interest from Fastly
- Will be publicly released



https://www.cloudflare.com/en-gb/network/

Motivation

- Research
  - Tool used for several papers we are working on
  - Hopefully, tool useful for other researchers in their work
- Operators
  - Collaborate with anycast operators
  - Allow operators to improve their services
- Improve Anycasted services -> improve Internet resilience

Tool use cases

- Actively measure how people route to your anycast deployment
- Measure latency
- Measure best-case latency
- Anycast Paris traceroute
- IPv6/TCP/UDP support (gap in anycast research)
- Much more ...

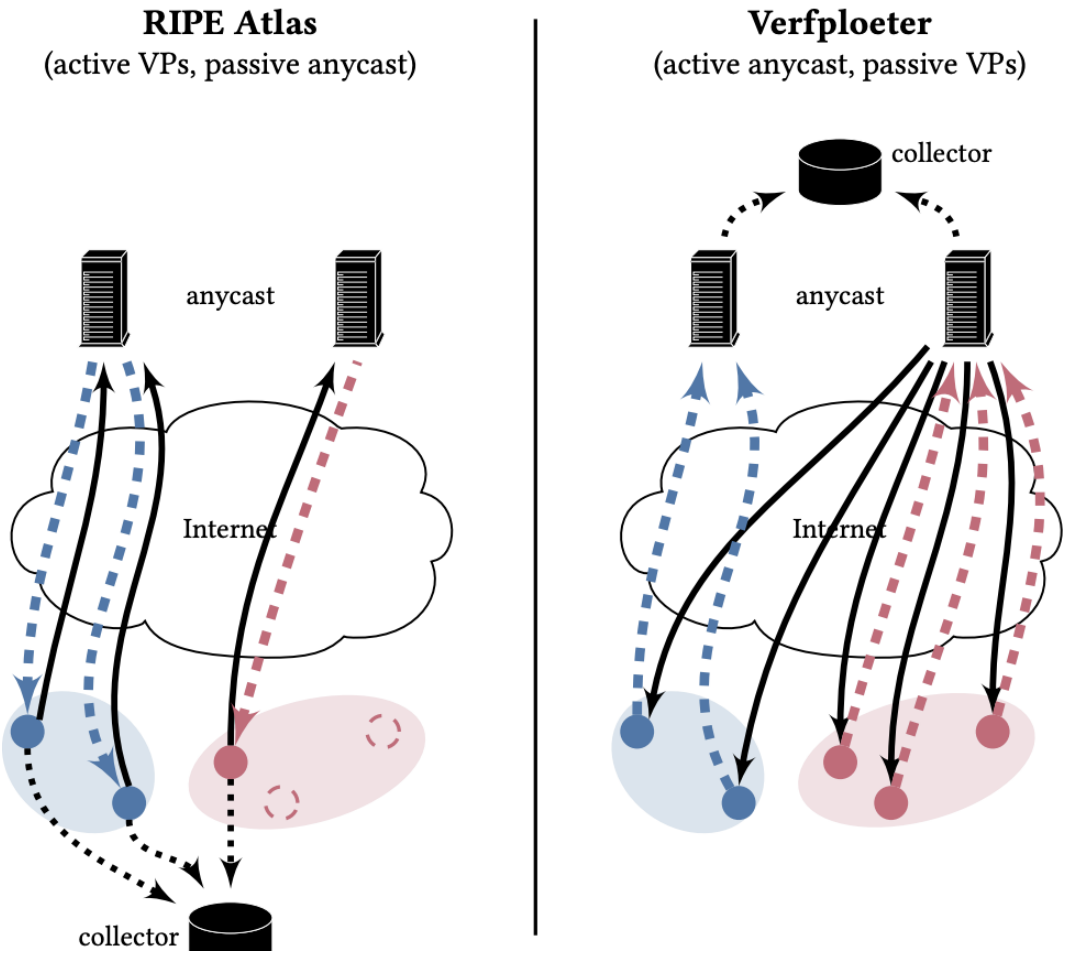
What we need

- Expand our anycast testbed
  - We have our own prefix and AS
  - We need VMs where we can announce it
- Especially interested in economically developing regions

Appendix slides

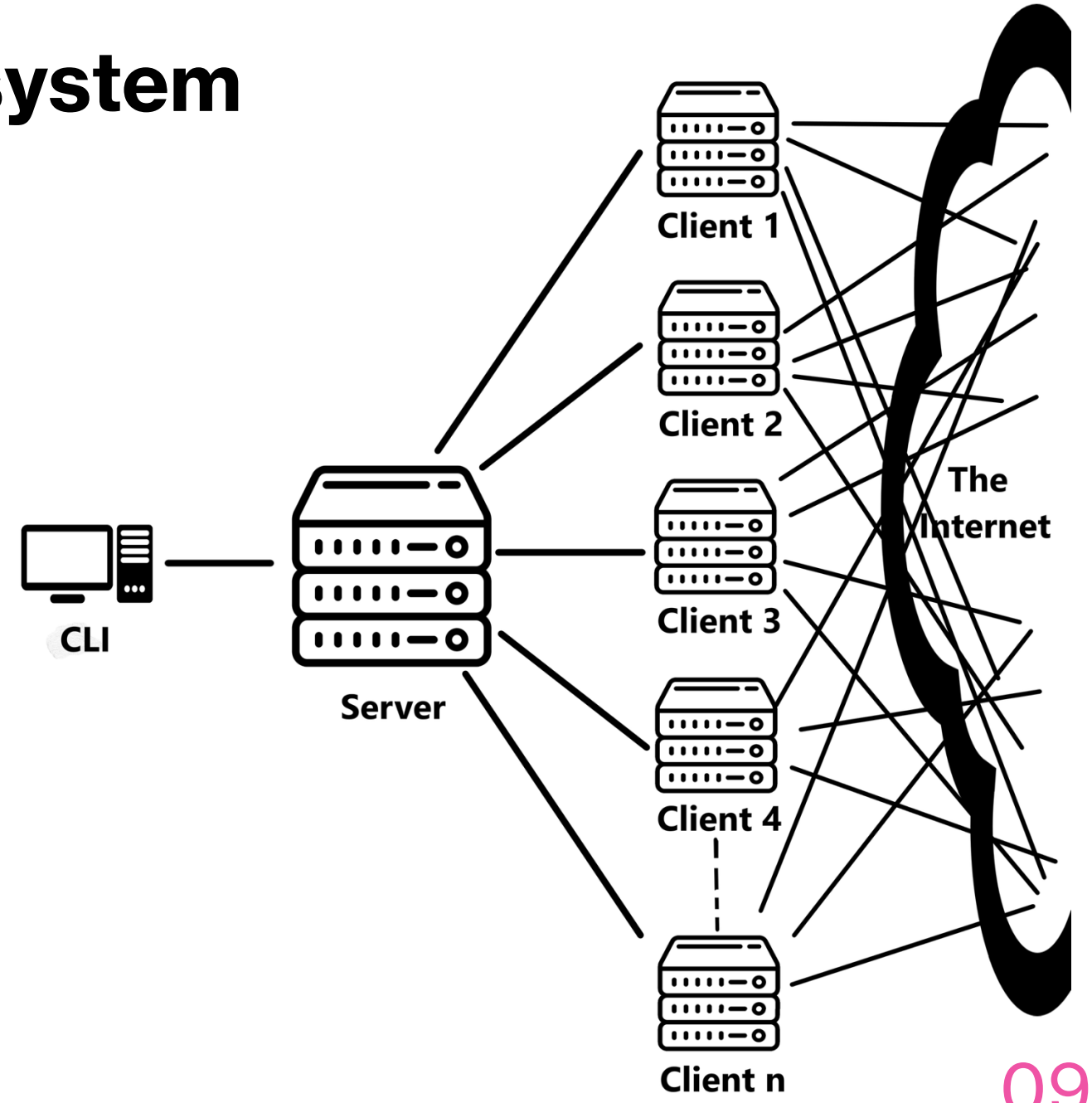
Verfploeter

- Maps anycast catchment
  - i.e., the set of addresses routing to each site
  - Useful for identifying load-distribution
- Uses ping responsive hosts (10^6)



The system

- CLI defines measurement
- Server orchestrates measurement
- Clients probe Internet



Features
Multi-protocol support

- IPv6 supported
  - Gap in anycast performance research
- Added TCP, UDP support
  - Higher coverage and responsiveness
- DNS A & TXT support

Features
Multi-protocol support (results)

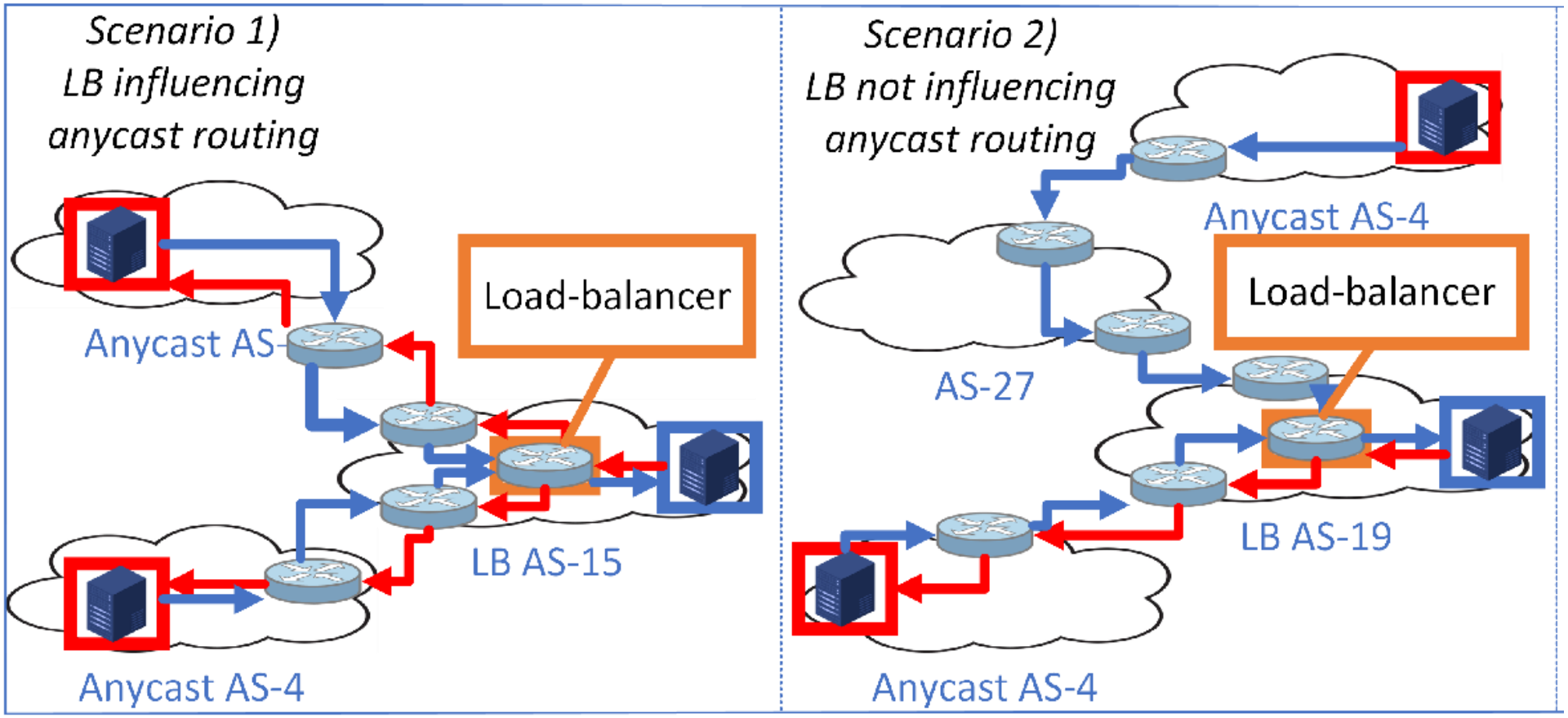
- Compared ICMP with TCP and UDP
  - Anycast routing consistent among protocols
  - ICMP measurements are valid for anycast TCP/UDP/.. services
- Census
  - Able to detect anycasted prefixes that are not ICMP responsive
- CHAOS
  - Useful for troubleshooting Anycast to Anycast (DNS) routing

Features
Multi-'Origin'

- Origin 3-tuple: (src IP, src & dst port)
  - Configurable by CLI
- Supports multi-origin probing
  - Synchronously probe with multiple origins
  - Probe with multiple prefixes
    - Assess effectiveness of changes
    - What if site X goes offline

Features
Multi-'Origin'

- Detects load-balancer induced routing instability



Features Multi-'Origin' (results)

- Large scale study of Load-Balancing versus Anycast
- Submitted to TNSM
- Main takeaways:
  - Vast majority of Load-Balancing is flow-based
  - We observe large latency differences among LB paths
  - Traffic engineering needs to account for these cases
  - Can cause false spoofed classifications

Features
Divide-and-conquer

- Hitlist divided among vantage points
  - Significantly speeds up probing process (multiplication factor of # of VPs)
  - Distributes measurement burden among sites (and upstreams)
  - Allows for catchment mappings in minutes (or even seconds)
  - Speed and ease allows for frequent mappings
    - *e.g.,* for spoofed traffic detection

```
remi@manycast:~$ ~/dev/stable/manycast-extended/target/debug/manycast cli -s 192.87.172.29:50001 start -a 145.116.218.1 data/v420240813.csv 1 -o out/ --shuffle -u edu.nl/9qt8h --divide
[Main] Executing CLI version git-1240117
[CLI] Connecting to Controller Server at address 192.87.172.29:50001
[CLI] Probes will be sent out from all clients
[CLI] Performing ICMP/ping measurement targeting 5,651,598 addresses, with a rate of 1,000, and an interval of 1
[CLI] Clients send probes using the following origin: source IP: 145.116.218.1, source port: 62321, destination port: 63853
[CLI] This measurement will be divided among clients (each client will probe a unique subset of the addresses)
[CLI] This measurement will take an estimated 2.96 minutes
[CLI] Measurement started at 15:18:15
⠒ [00:00:18] [####>-----------------------------------] 19/177 (2m)
```

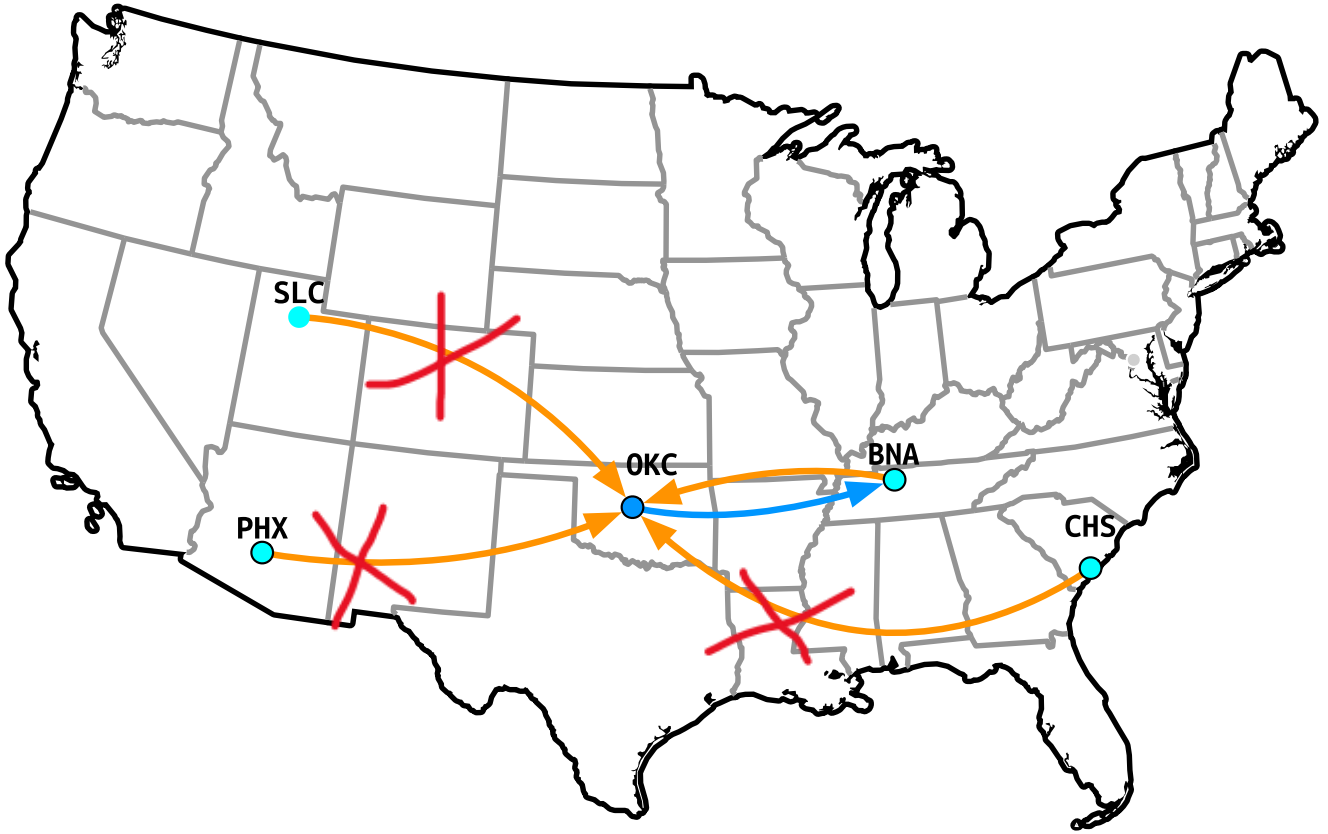
Catchment mapping of 5.7 million addresses in 3 minutes (using a probing rate of 1,000 at each anycast site)

Features
Distributed synchronized probing

- Synchronous probing from all VPs
  - Just like ping (1-second intervals) **interval configurable*
  - Reduces FP rate in census
  - Allows for measuring ingress dependent routing policies (Microsoft)

Features
Distributed synchronized probing

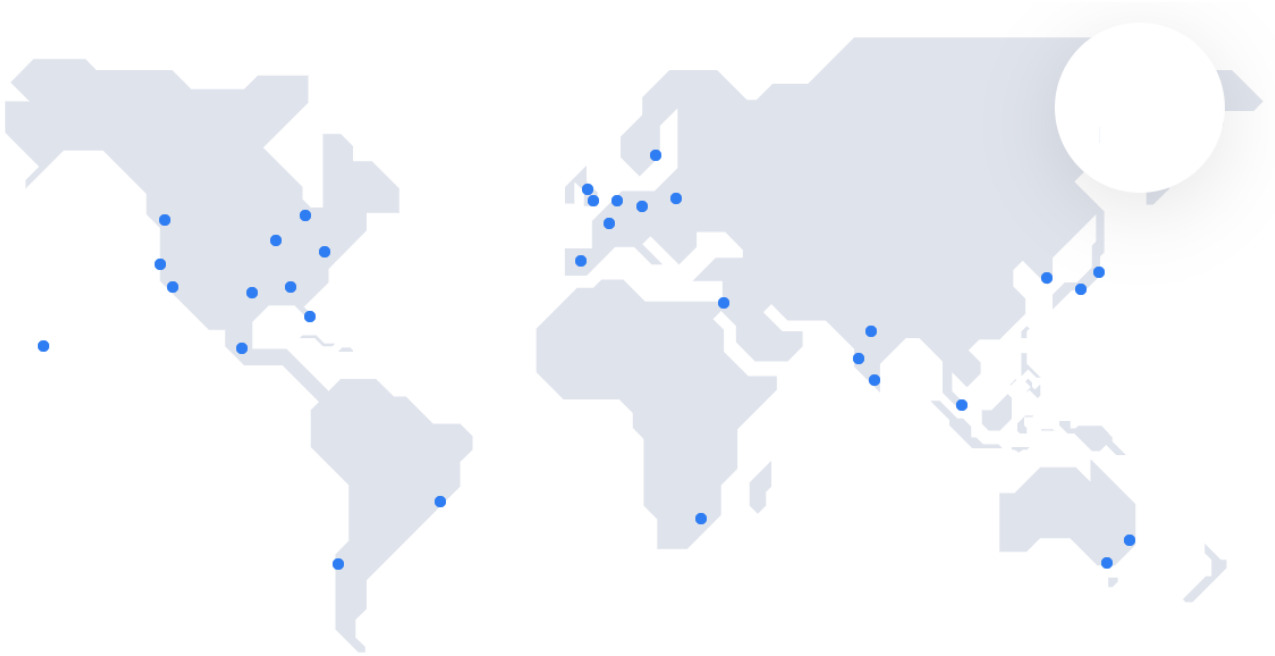
- Allows for measuring anycast RTT



Features
Distributed synchronized probing

```
Latency to the USC/ISI hitlist

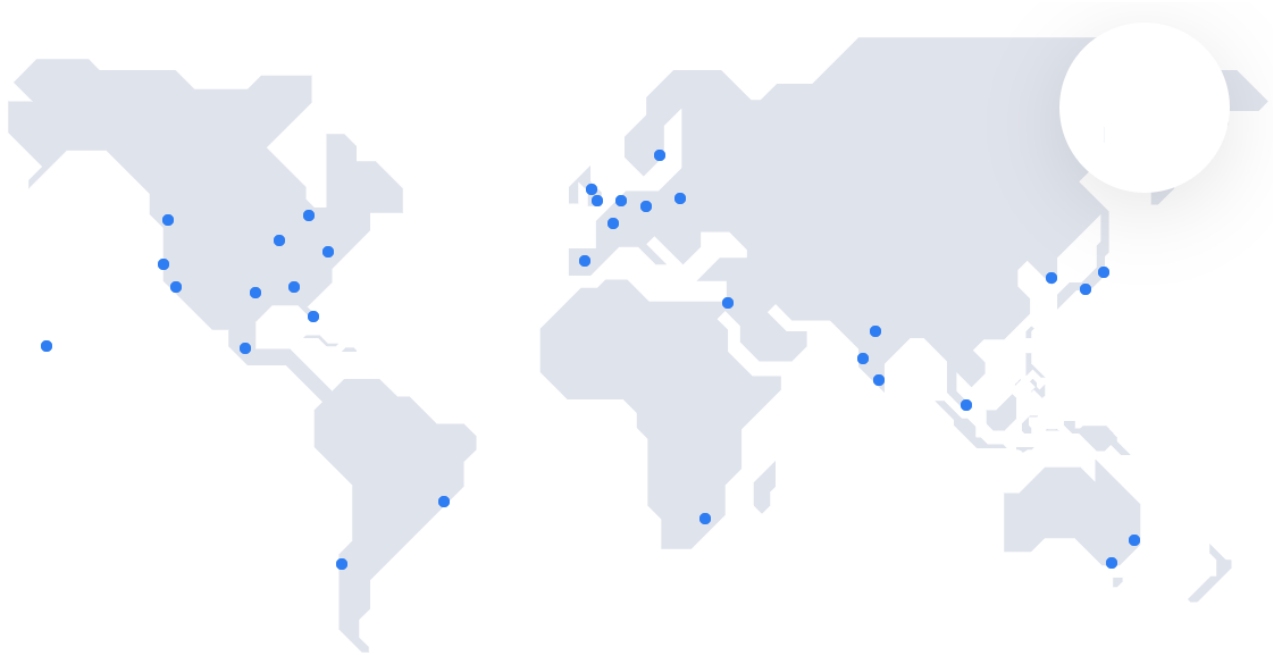
Average   65.7 ms
25%       9.8 ms
50%       23.5 ms
75%       74.1 ms
```



Our anycast deployment (32 locations)
(deployed using Vultr)

Catchment analysis
Anycast RTT data (example)

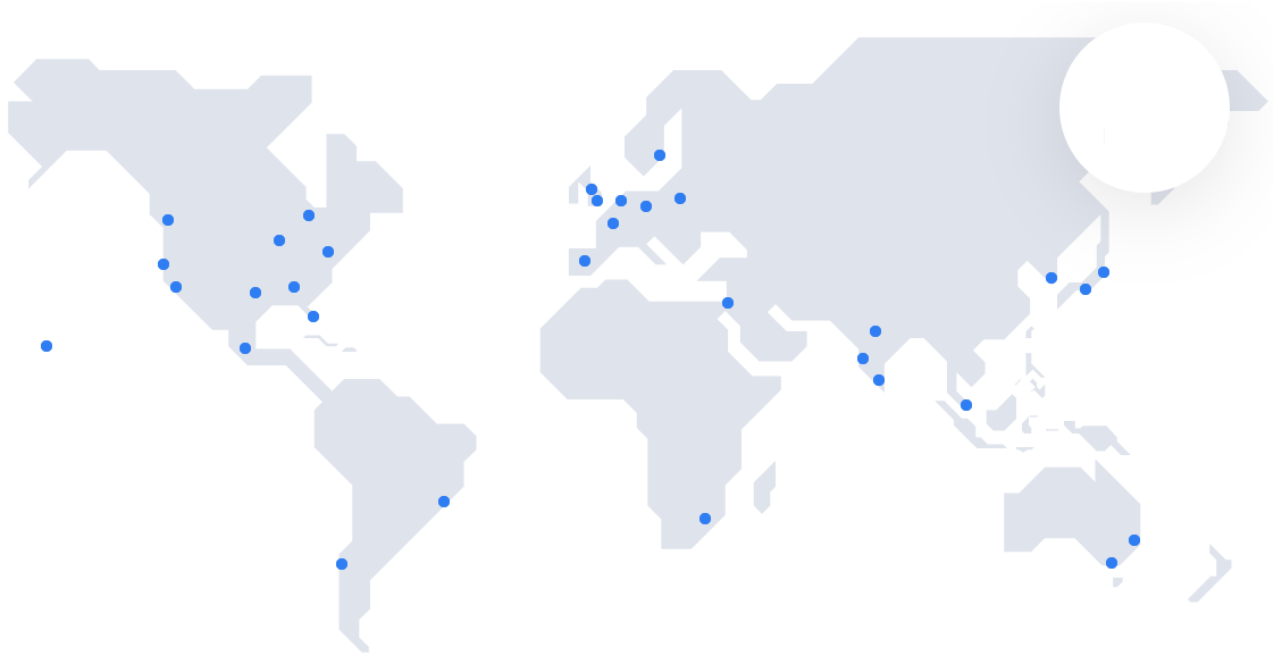
Site	Mean RTT (in ms)	Catchment size
de-fra	111.3	385,264
kr-icn	60.8	379,341
us-ewr	29.3	268,475
in-bom	146.1	264,222
jp-nrt	88.3	233,544
us-ord	29.3	194,599
us-mia	109.2	150,184
us-lax	51.3	146,907
br-sao	101.6	146,285
gb-lhr	33.1	140,632



Our anycast deployment (32 locations)

Catchment analysis
Anycast RTT data (example)

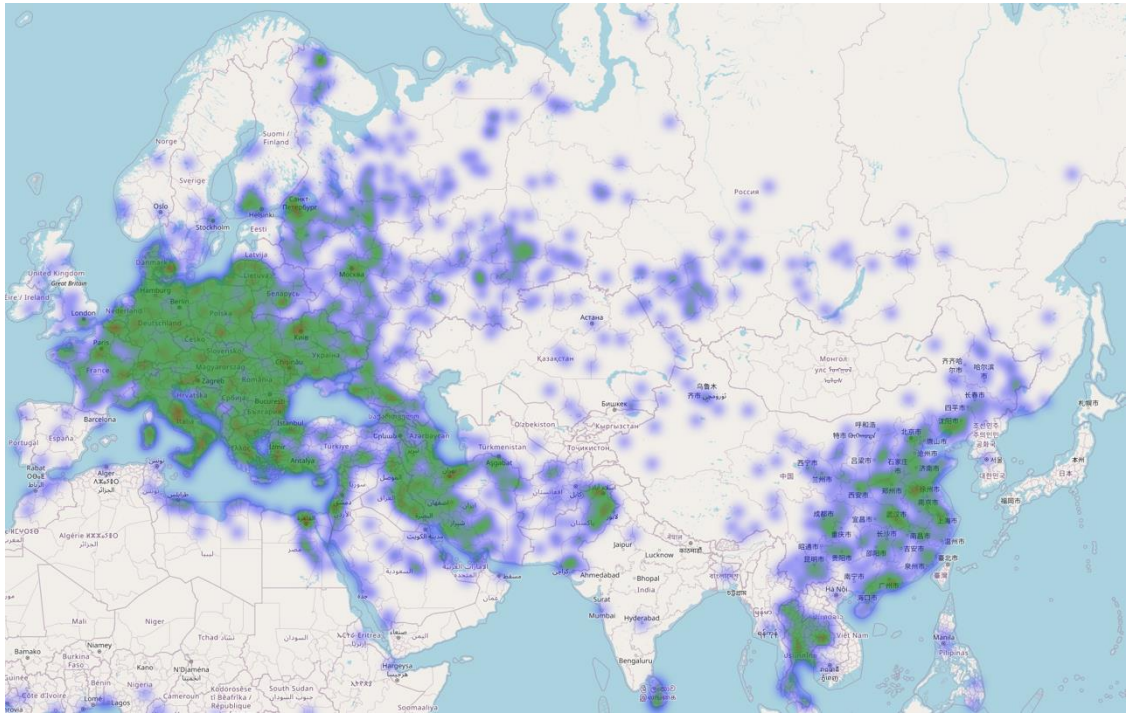
Site	Mean RTT (in ms)	Catchment size
de-fra	111.3	385,264
kr-icn	60.8	379,341
us-ewr	29.3	268,475
in-bom	146.1	264,222
jp-nrt	88.3	233,544
us-ord	29.3	194,599
us-mia	109.2	150,184
us-lax	51.3	146,907
br-sao	101.6	146,285
gb-lhr	33.1	140,632



Our anycast deployment (32 locations)

Catchment analysis
Anycast RTT data (example)

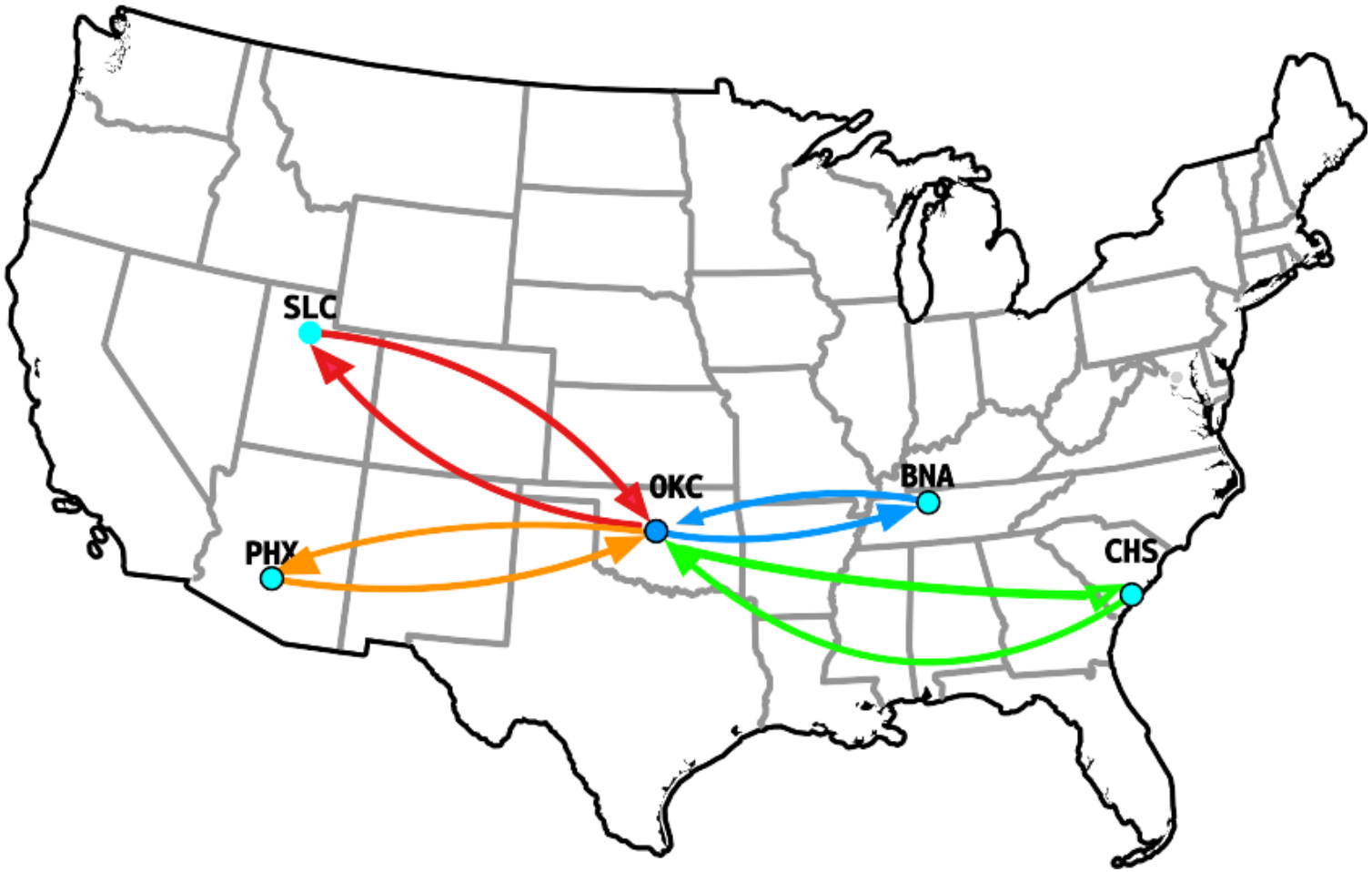
- Zooming in on de-fra
- Mapping target locations using *ip2location*
- Large number of Chinese prefixes (99k)
  - Most from a few ASes



Features
Unicast probing

- System allows for probing with unicast IPs from anycast sites
  - Can synchronously probe from all VPs with unicast IP
  - Allows for finding closest site, and lowest achievable RTT for each target

Features
Unicast probing



Catchment analysis
Anycast RTT data vs. best-case unicast RTT

- Filter on min($RTT_{unicast}$) -> best-case unicast RTT
  - I.e., the site that has the lowest RTT towards a target
  - The highest achievable performance (in terms of latency) for the anycast deployment
- Compare with $RTT_{anycast}$ -> $\Delta = RTT_{anycast} - min(RTT_{unicast})$
  - Data on actual anycast performance compared to best-case unicast routing
  - Detect regions/ASes that experience sub-optimal anycast routing

Catchment analysis
Anycast RTT data vs. best-case unicast RTT

- min($RTT_{unicast}$) mean is 39.4 ms

Catchment analysis
Anycast RTT data vs. best-case unicast RTT

- $\min(RTT_{unicast})$ mean is 39.4 ms
- $RTT_{anycast}$ mean is 65.7 ms

Catchment analysis
Anycast RTT data vs. best-case unicast RTT

- $\min(RTT_{unicast})$ mean is 39.4 ms
- $RTT_{anycast}$ mean is 65.7 ms
- $\Delta = RTT_{anycast} - \min(RTT_{unicast}) =$ 26.3 ms

Catchment analysis
Anycast RTT data vs. best-case unicast RTT

- $\min(RTT_{unicast})$ mean is 39.4 ms
- $RTT_{anycast}$ mean is 65.7 ms
- $\Delta = RTT_{anycast} - \min(RTT_{unicast})$ = 26.3 ms
  - Substantial latency inflation due to sub-optimal anycast routing
  - Large gain in performance (+40%) possible using *e.g.,* BGP prepending, selective announcements, ...

Catchment analysis
Anycast RTT data vs. best-case unicast RTT

- $\min(RTT_{unicast})$ mean is 39.4 ms
- $RTT_{anycast}$ mean is 65.7 ms
- $\Delta = RTT_{anycast} - \min(RTT_{unicast}) =$ 26.3 ms
  - Substantial latency inflation due to sub-optimal anycast routing
  - Large gain in performance (+40%) possible using *e.g.,* BGP prepending, selective announcements, ...

- Future work:
  - Automated detection and solving of sub-optimal anycast routing

Features
Other significant contributions

- TLS support
  - Security requirements of SIDN
- Many options (show image of --help)
- Anycast Paris traceroute (currently broken)
- Responsive host finder

Thoughts and feedback

- Possible measurements for anycast operators
- Research ideas, using anycast probing system
- Collaboration
- We can easily perform scans, if you share a list of addresses you'd like to have checked for anycast

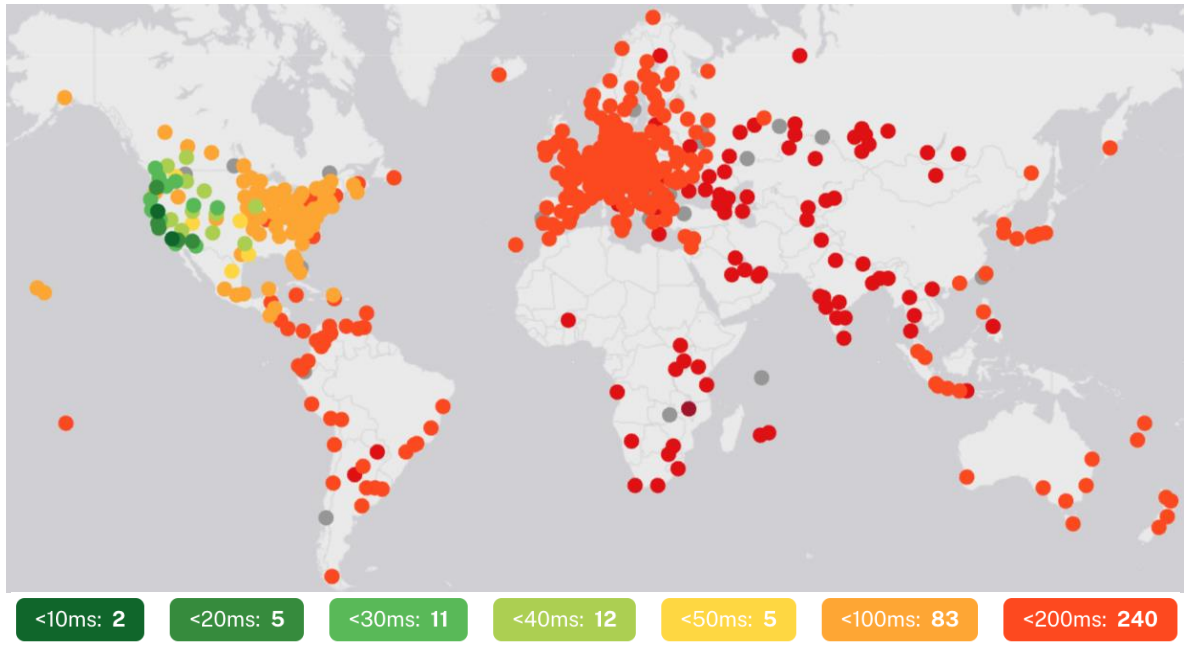
Future

- eBPF
- Support reverse traceroute
- Simplify deployment for operators
- Support for automated data analysis
- Automatically solve sub-optimal routing

Appendix slides

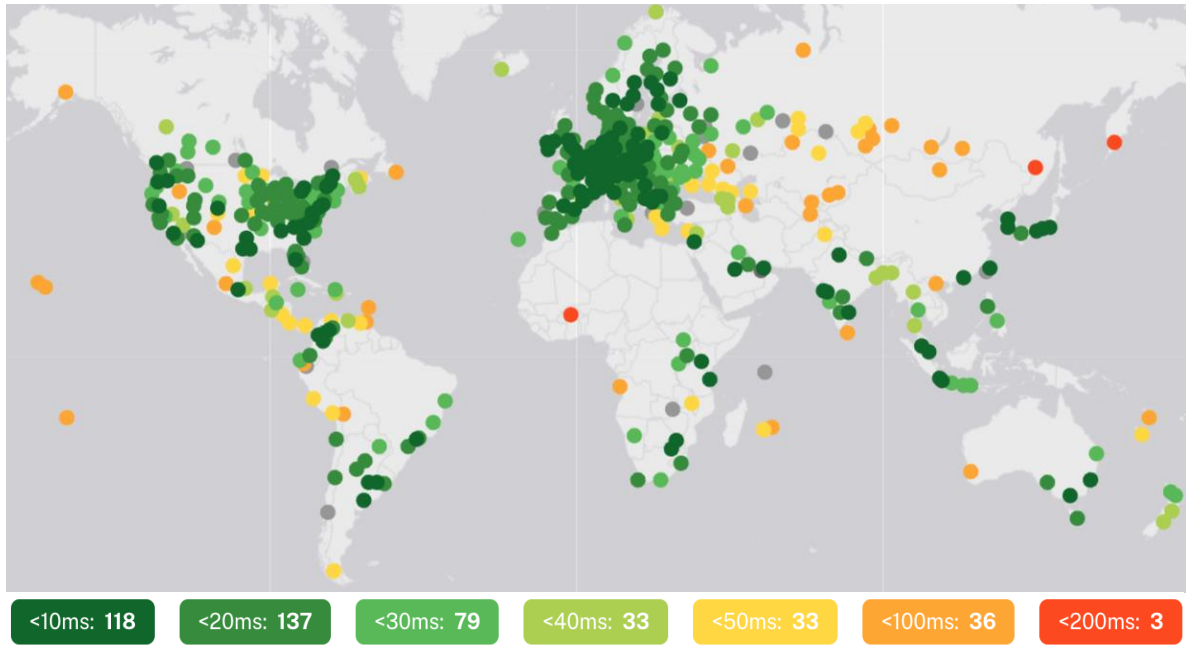
GCD measurement
RIPE Atlas example

Unicast



https://atlas.ripe.net/measurements/79147535/

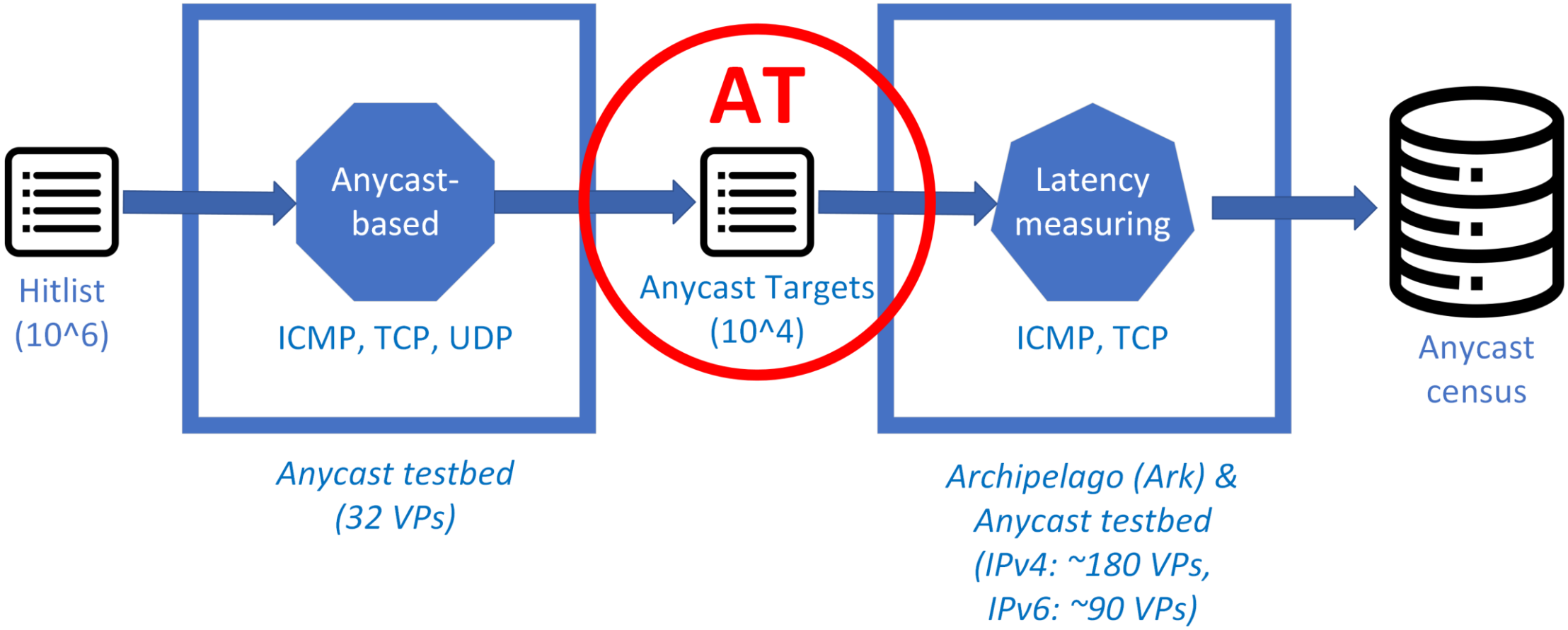
Anycast



https://atlas.ripe.net/measurements/79137270/

RIPE NCC
RIPE Atlas

Pipeline


Hitlist
(10^6)
Anycast-based
ICMP, TCP, UDP
AT
Anycast Targets
(10^4)
Latency measuring
ICMP, TCP
Anycast census
Anycast testbed
(32 VPs)
Archipelago (Ark) &
Anycast testbed
(IPv4: ~180 VPs,
IPv6: ~90 VPs)

Results
GCD-confirmed prefixes

AS	Organization	IPv4	IPv6
396982	Google Cloud	3,345	3
13335	Cloudflare	3,131	162
16509	Amazon	1,235	86
54113	Fastly	438	56
15169	Google	282	6
209242	Cloudflare Spectrum	234	2,836
19551	Incapsula	2	292
12041	Afilias	222	207
44273	GoDaddy	31	122

Table 6: Largest ASes originating anycast prefixes for IPv4 and IPv6.

Results
Anycast-based by protocol

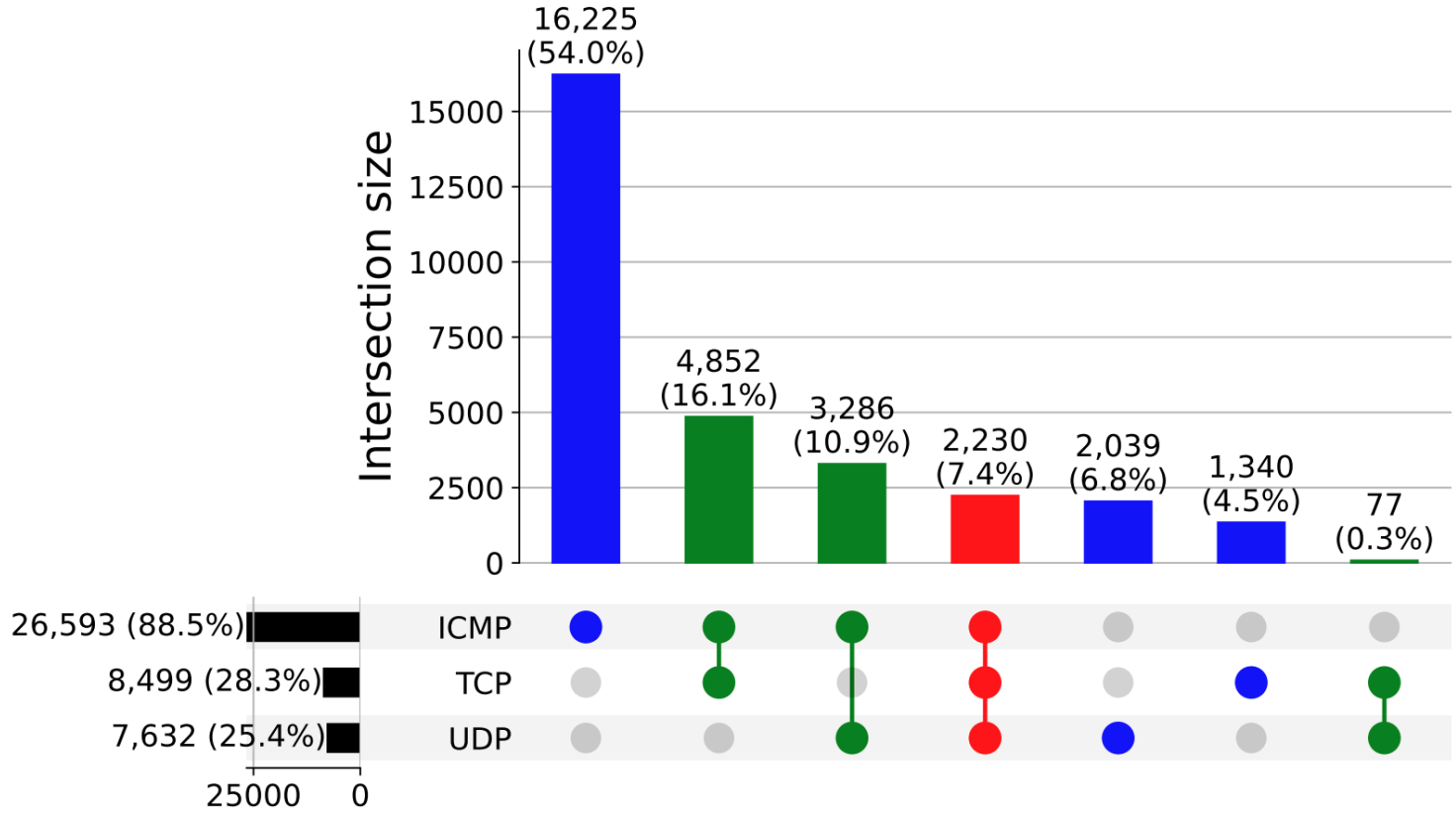
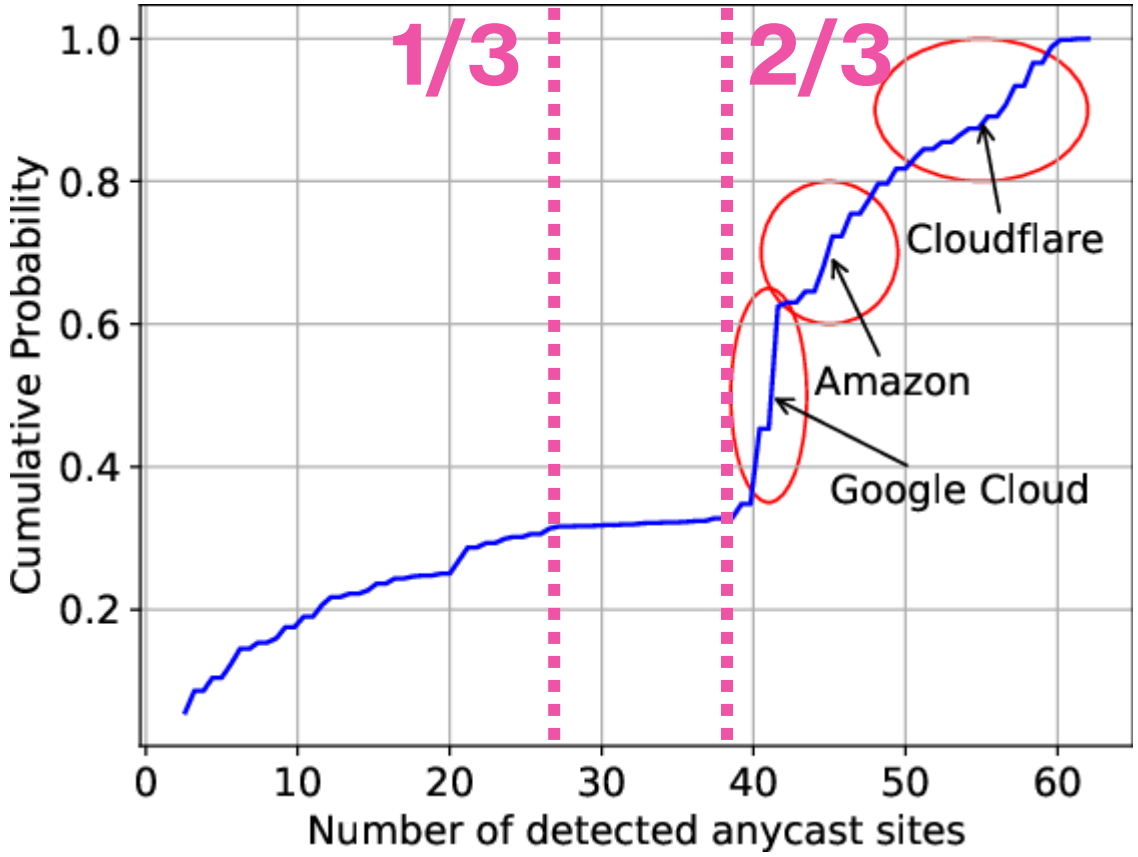


Figure 6: MAnycastR detection of anycast candidates for ICMPv4, TCPv4, and UDPv4.

Results
Enumerating deployments



Enumeration:

- Lower bound of real deployment
- Fairly accurate for small deployments (few sites)
- Indication of size for large deployments (many sites)

Appendix slides
Daily census output snippet classification + enumeration

File: YYYY/MM/DD/YYYY-MM-DD_v4.json


```
{"prefix": "1.0.0.0/24", "characterization":
    {"MAnycastICMPv4": {"anycast": true, "instances": 26},
    "MAnycastTCPv4": {"anycast": true, "instances": 26},
    "MAnycastUDPv4": {"anycast": null, "instances": 0},
    "iGreedyICMPv4": {"anycast": true, "instances": 60},
    "iGreedyTCPv4": {"anycast": true, "instances": 26}
    }
},
...
```

Appendix slides
Daily census output snippet locations

File: YYYY/MM/DD/YYYY-MM-DD_v4_locations.json


```
 {
   "prefix": "1.0.0.0/24",
   "count": 60,
   "instances": [
    {
      "marker": {
        "city": "Honolulu",
        "code_country": "US",
        "id": "HNL",
        "latitude": 21.3187007904,
        "longitude": -157.9219970703
      }
    },
...
```

Appendix slides
MAnycastR FNs breakdown

# of sites receiving	Candidate anycast	TPs	FPs	TP rate
2	14,966	620	14,346	4.14%
3	421	347	74	82.42%
4	307	284	23	92.51%
5	400	364	36	91.00%
5-10	918	901	17	98.15%
10-15	831	825	6	99.28%
15-20	3,890	3,886	4	99.90%
20-25	2,288	2,288	0	100.00%
25-32	2,572	2,572	0	100.00%
Total	26,593	12,087	14,506	45.45%

Table 3: Anycast-based ICMPv4 TP rate per number of sites receiving replies, taking GCD_{Ark} as ground-truth.

Appendix slides
MAnycastR IPv6 ATs by protocol

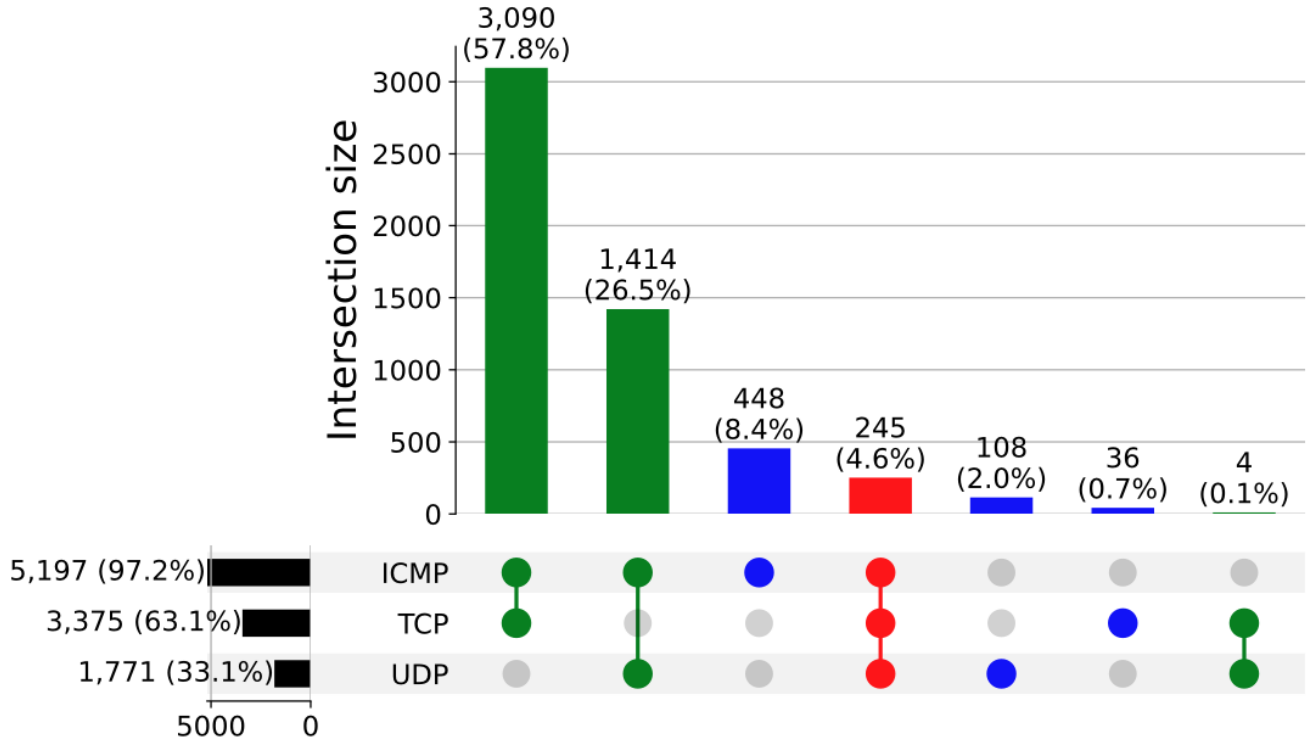


Figure 7: MAnycastR detection of anycast candidates for ICMPv6, TCPv6, and UDPv6.