Anycast tooling:

For operators and researchers

GMI-AIMS-5, San Diego, USA

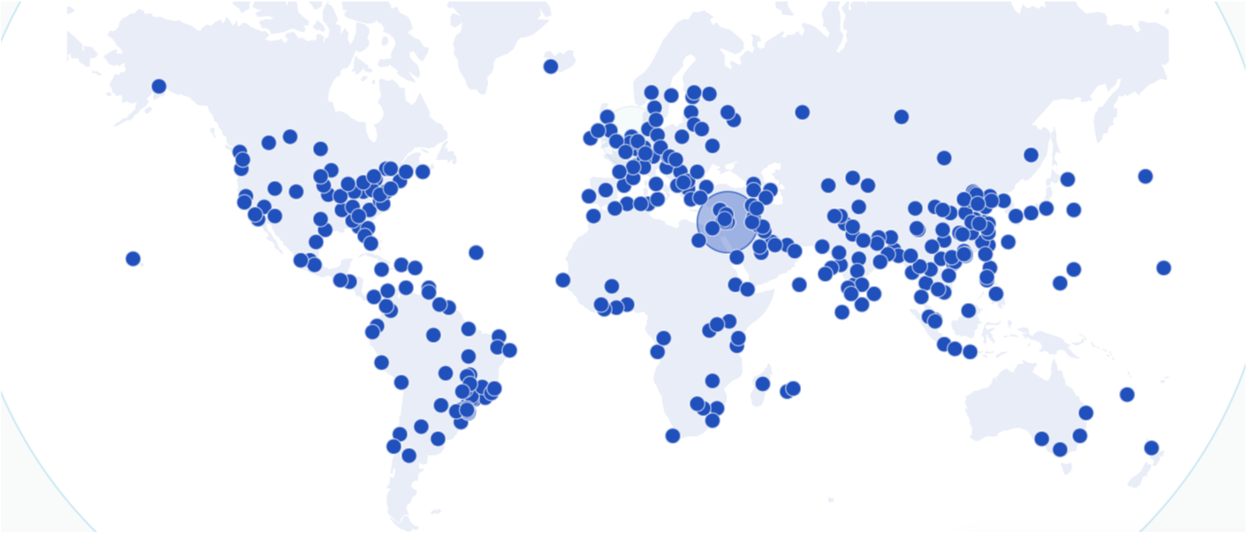
FEB, 2025

Remi Hendriks

UNIVERSITY
OF TWENTE.

Motivation

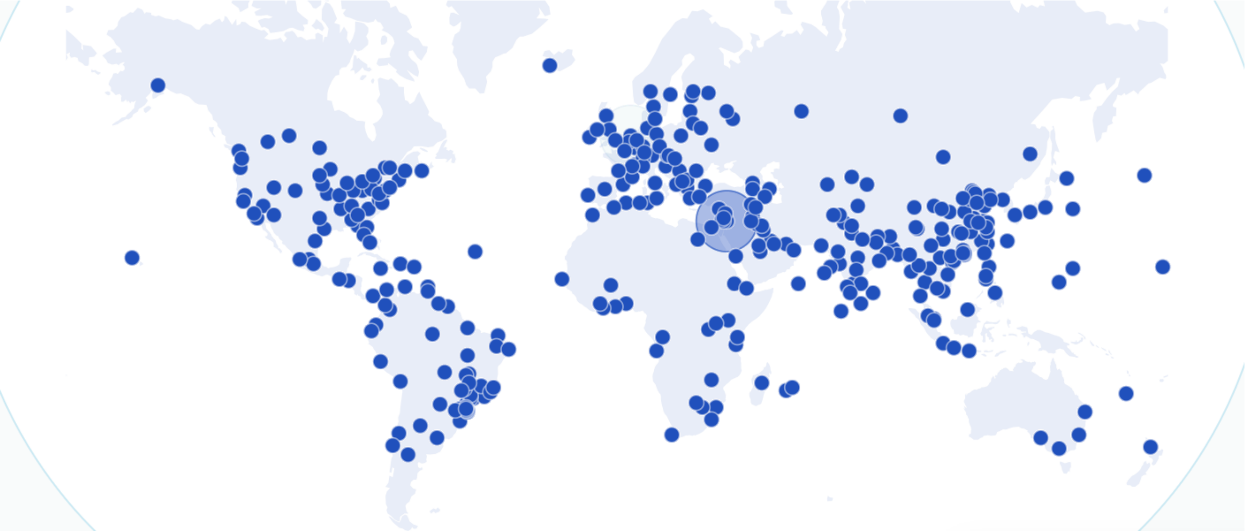
- Anycast: replicating a service at multiple locations using a single shared IP address
  - Querying 1.1.1.1 from New Zealand -> reach server in New Zealand
  - Querying 1.1.1.1 here -> reach Cloudflare server in San Diego



Cloudflare's anycast network

Motivation

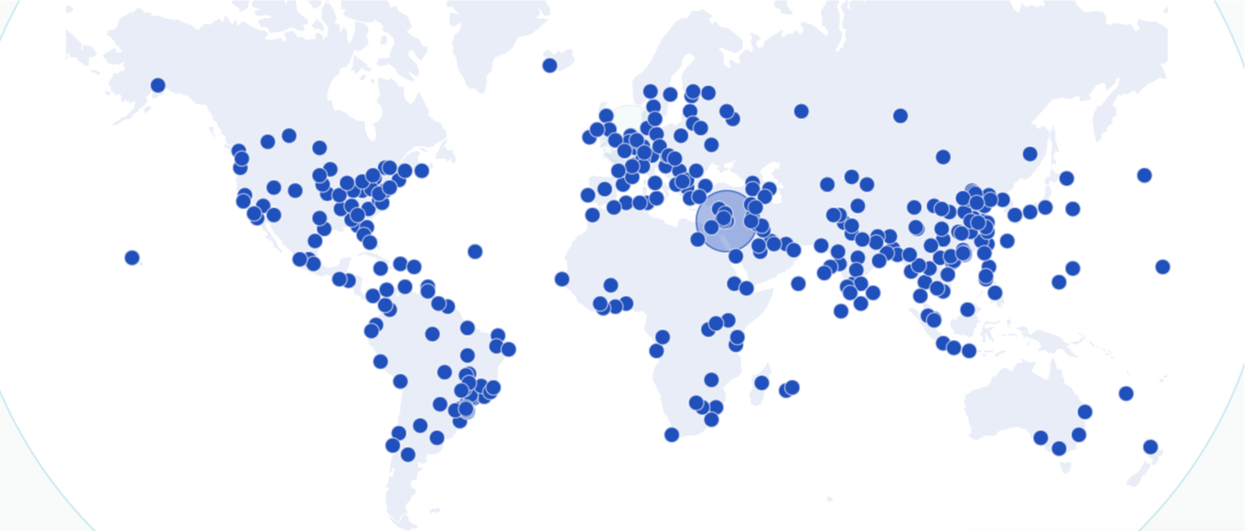
- Anycast: replicating a service at multiple locations using a single shared IP address
  - Querying 1.1.1.1 from New Zealand -> reach server in New Zealand
  - Querying 1.1.1.1 here -> reach Cloudflare server in San Diego
- Used for critical Internet infrastructure (e.g., DNS)
- Used by CDNs for a large variety of services
- Used to provide DDoS mitigation services



Cloudflare's anycast network

Motivation

- Anycast: replicating a service at multiple locations using a single shared IP address
  - Querying 1.1.1.1 from New Zealand -> reach server in New Zealand
  - Querying 1.1.1.1 here -> reach Cloudflare server in San Diego
- Used for critical Internet infrastructure (e.g., DNS)
- Used by CDNs for a large variety of services
- Used to provide DDoS mitigation services
- Why?
  - ○ Proven technique
  - ○ Reduces latency, load-balances traffic
  - ○ Most importantly, improves resilience



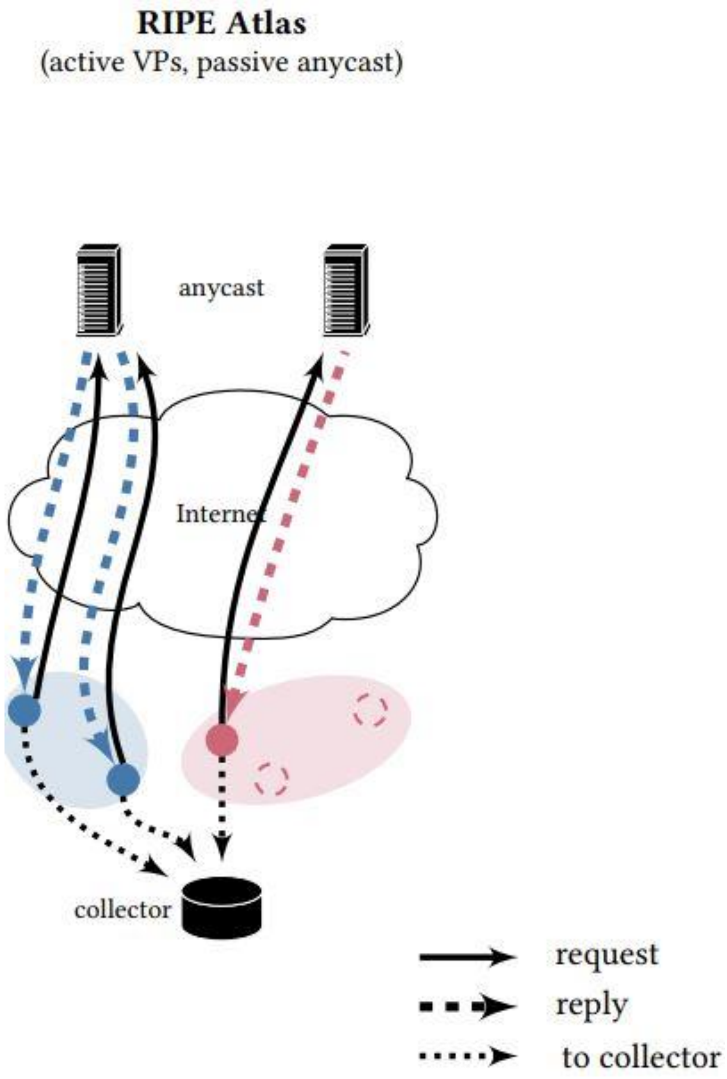
Cloudflare's anycast network

Problem statement

- Anycast relies on BGP to route clients to nearest PoP
  - BGP not designed for anycast routing
  - BGP not performance aware
  - *E.g.,* remote-peering may send traffic to different continents
  - Load-balancing and route flips cause anycast routing instability (short- and long-term)
- For these reasons, anycast requires active Traffic Engineering (TE)
- To make these TE decisions, performance metrics are needed

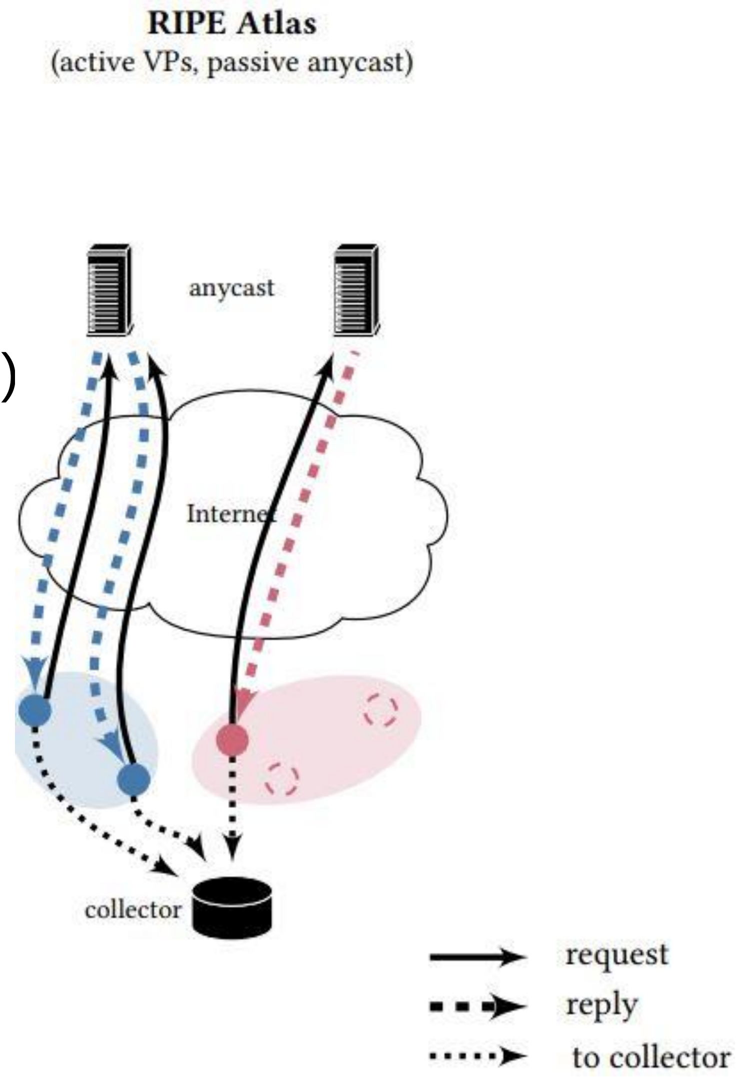
Measuring anycast

- Passive traffic analysis
  - Requires passive traffic data
  - Measures after applying changes



Measuring anycast

- Passive traffic analysis
  - Requires passive traffic data
  - Measures after applying changes
- External active measuring (e.g., RIPE Atlas, Ark)
  - Can measure proactively
  - Limited to the coverage of the probing platform



Measuring anycast

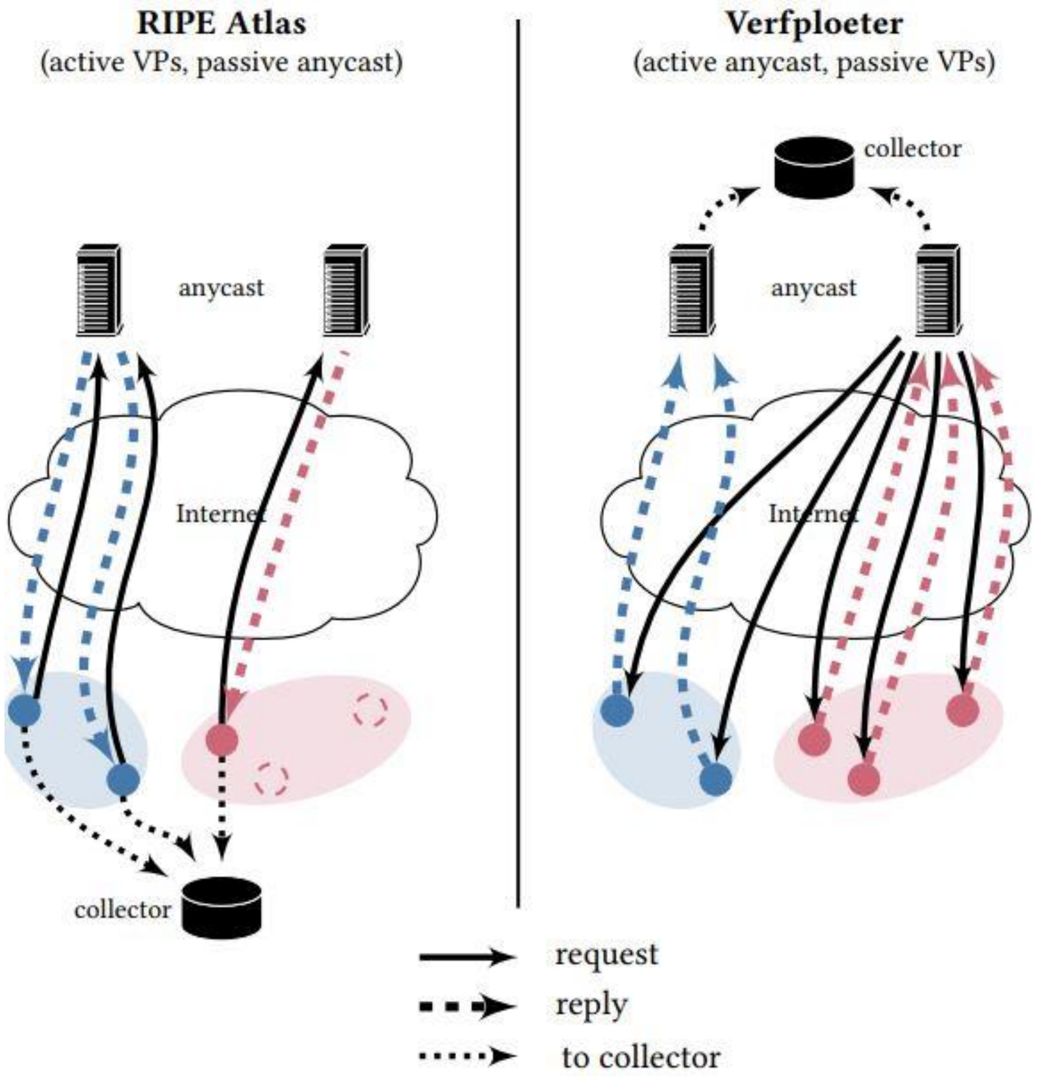
Verfploeter *[1]*

- Active anycast measuring

○ How?

  ○ Probe target with anycast source IP

  ○ Listen on all anycast sites for probe reply

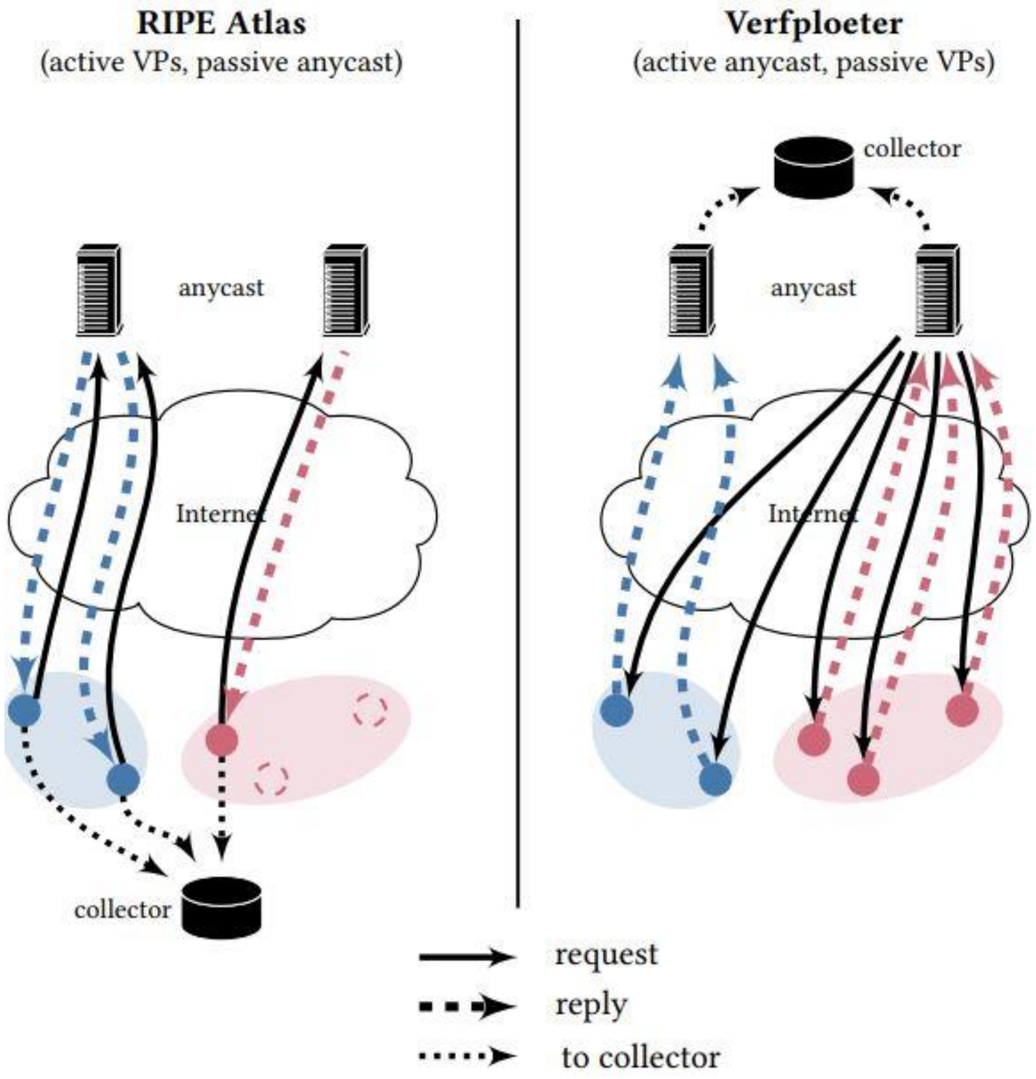


[1] De Vries et al. "Broad and load-aware anycast mapping with verfploeter." IMC’17

Measuring anycast

Verfploeter *[1]*

- Active anycast measuring
- How?
  - Probe target with anycast source IP
  - Listen on all anycast sites for probe reply
- Allows for catchment mapping
  - *I.e.,* which site 'catches' which part of the Internet
- Coverage of ~4 million /24s
  - **ICMP-responsive targets ISI hitlist*
- Methodology used by CloudFlare

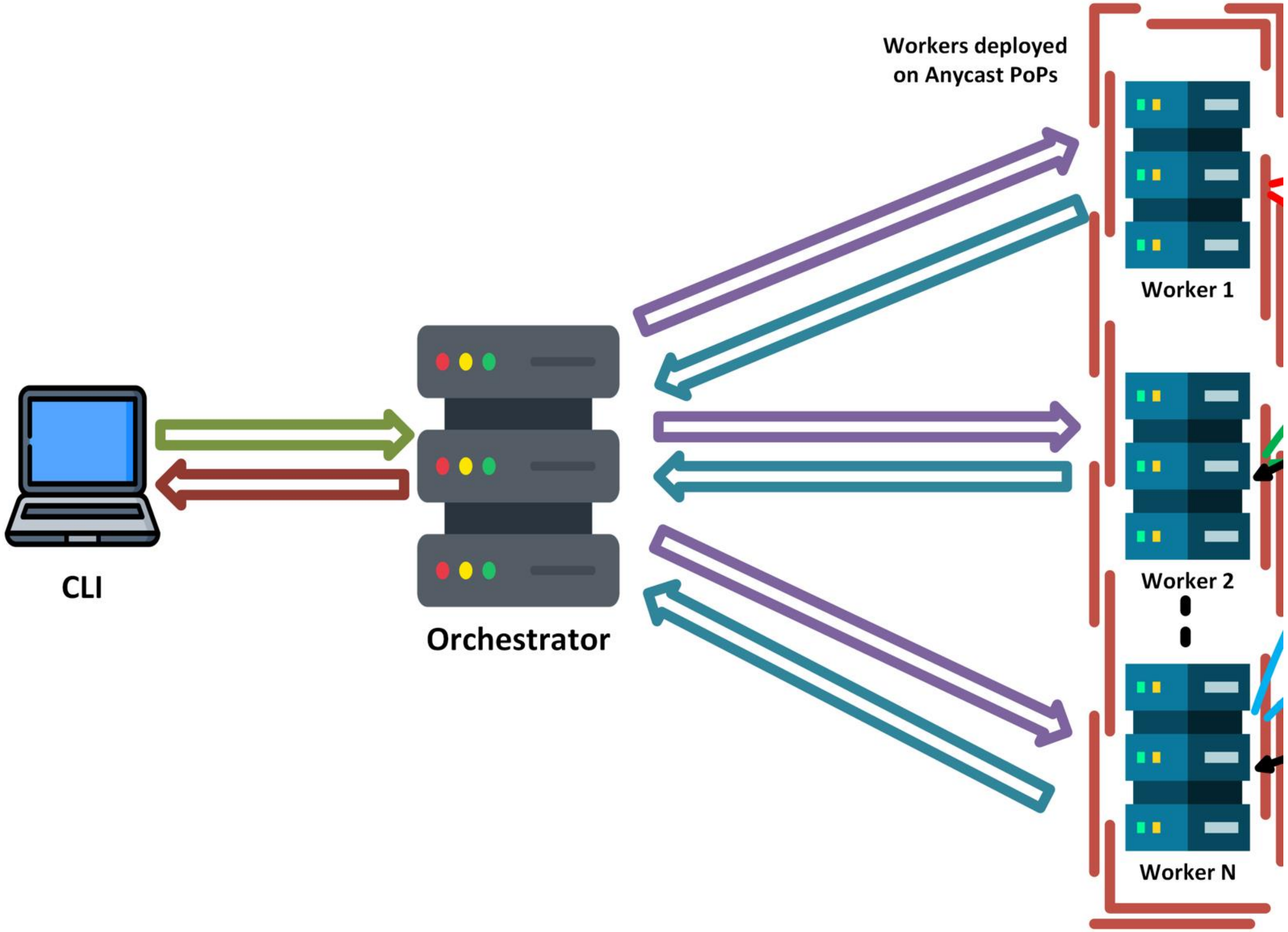


[1] De Vries et al. "Broad and load-aware anycast mapping with verfploeter." IMC'17

Our tooling

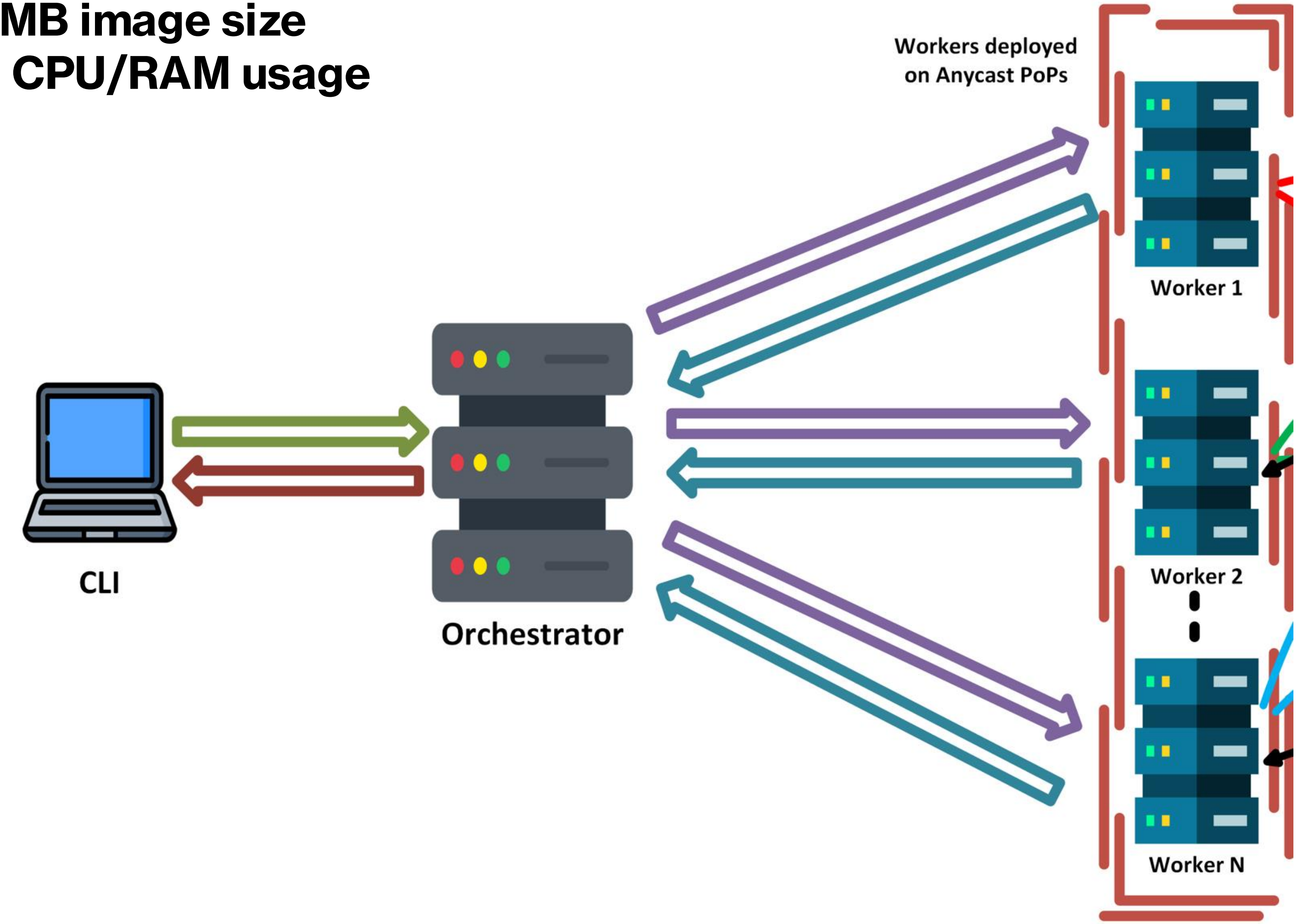
- Allows for unicast and anycast measurements
  - Including Verfploeter’s catchment mapping
- Designed as a ‘Swiss knife’
  - Many (mostly optional) configurable parameters
  - Configuration files (for complex measurements)
  - Large variety of supported measurements

Our tooling
System design



Our tooling
System design

130 MB image size
Low CPU/RAM usage

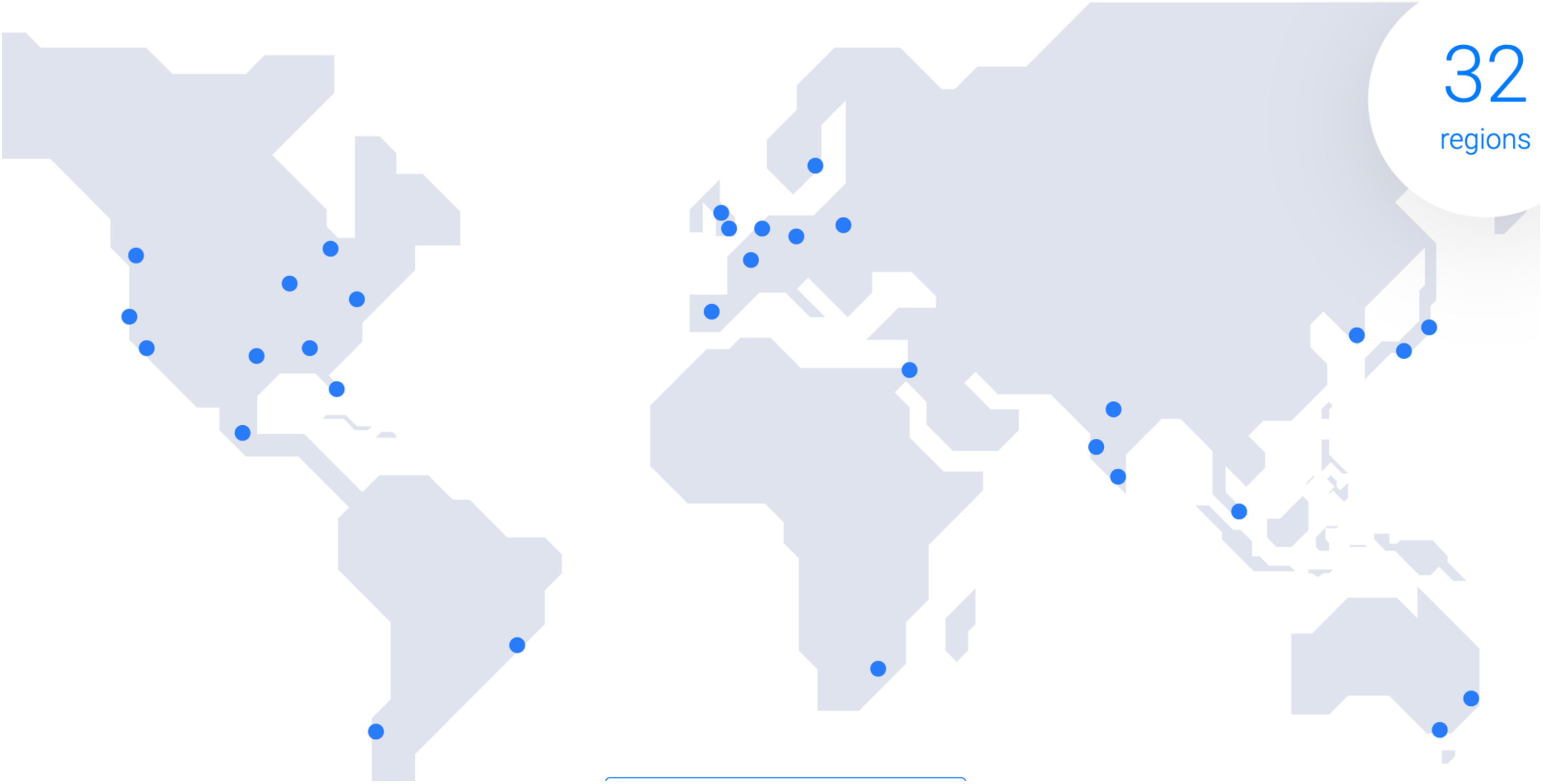


docker

Measurement setup

Deployed using Vultr (32 PoPs)

5.9 million /24-prefix targets (ISI hitlist)

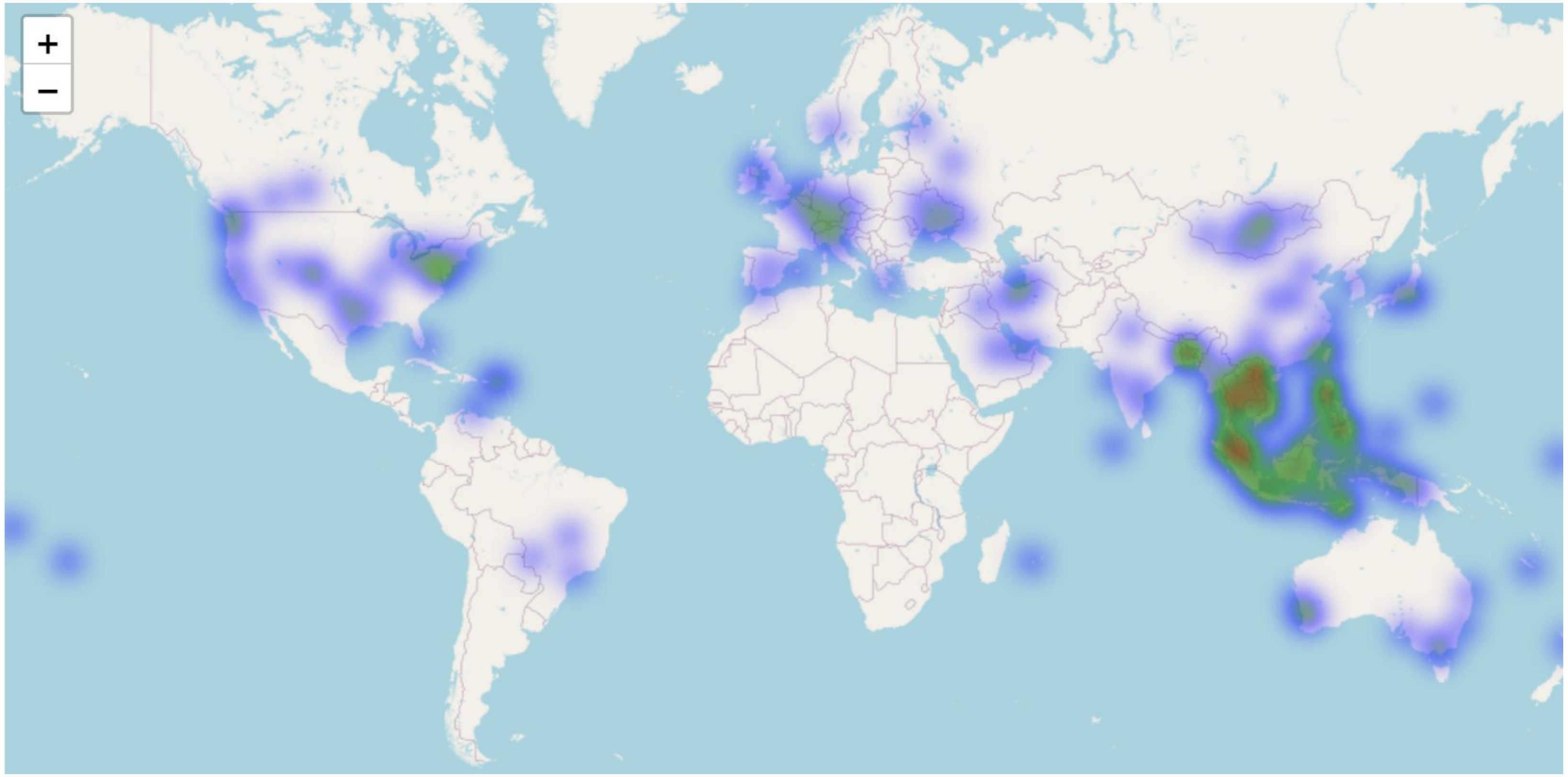


Verfploeter
Divide-and-conquer

- Improved Verfploeter using a divide-and-conquer approach
  - Divides hitlist among PoPs
  - Spreads probing burden among PoPs (including their upstreams)
  - Speeds up measurements significantly (with a factor of # of PoPs)
- Allows IPv4 catchment mapping (5.9 million targets) in 3 minutes
  - Using a modest probing rate of 1,000 pps (at each PoP)
  - Would be 98 minutes with traditional Verfploeter approach

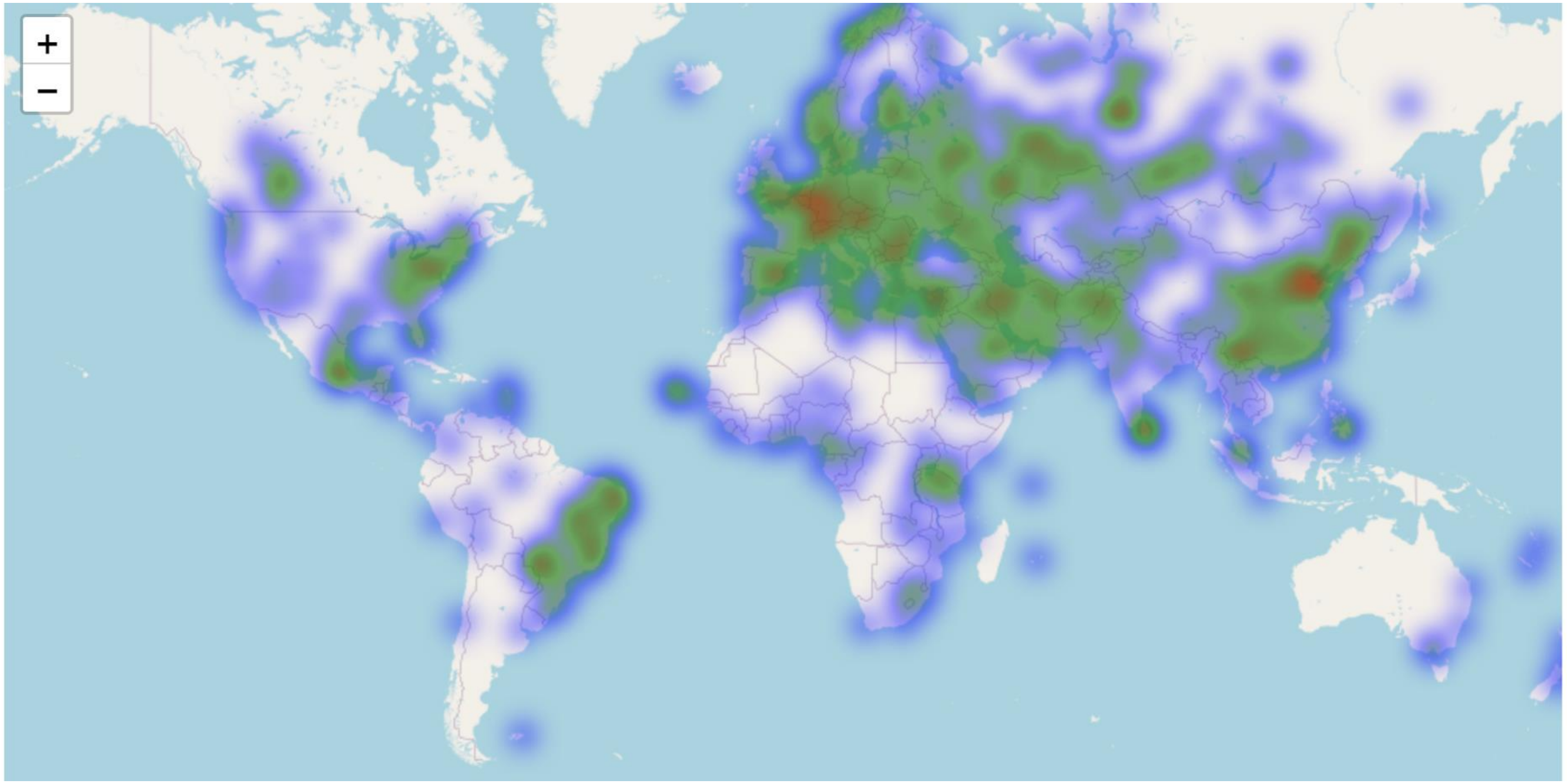
Verfploeter
Catchment mapping

Singapore (mostly good)



Verfploeter
Catchment mapping

Frankfurt (bad)



Protocol support

- UDP, TCP, ICMP supported
  - Extends coverage (not limited by ICMP-responsive hosts)
  - Answers concern that ICMP catchments do not hold for TCP/UDP anycast services
- IPv6 support
  - Lack of research in IPv6 anycast
  - IPv6 anycast routing is different (*e.g.*, HE a tier-1 for IPv6 only)

Multi-address probing

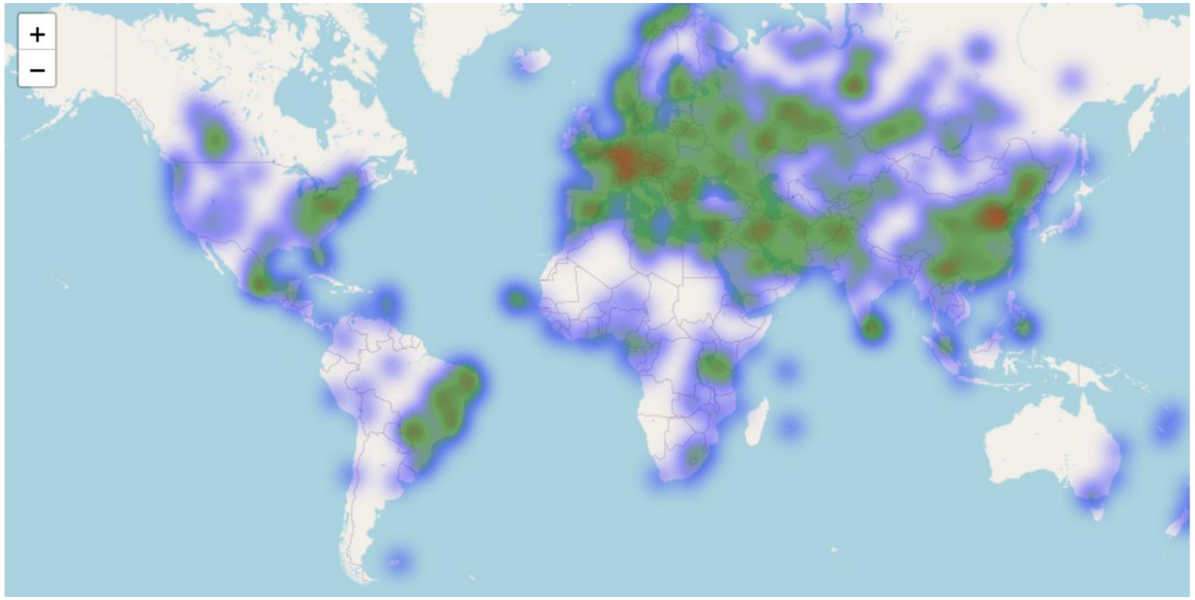
- Tool can measure with multiple addresses/port values simultaneously;
- Vary flow header to trigger load-balancing
  - See which regions may be load-balanced among different PoPs
  - We find load-balancing affects 4% of probed targets
  - Critical when *e.g.,* flagging spoofed traffic using catchment data

Multi-address probing

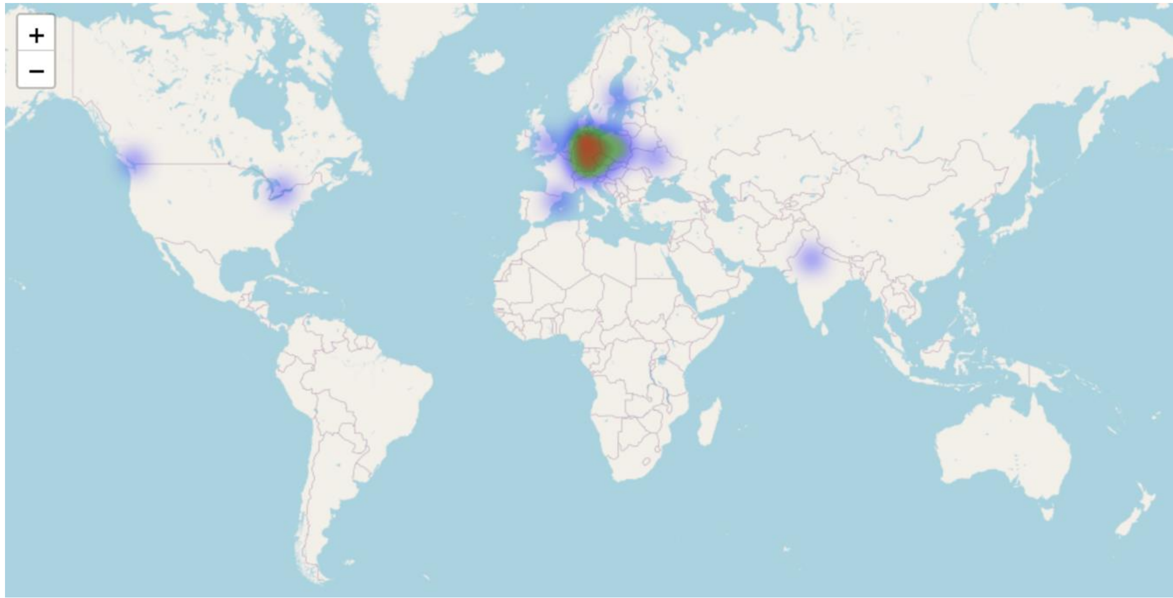
- Tool can measure with multiple addresses/port values simultaneously;
- Vary flow header to trigger load-balancing
  - See which regions may be load-balanced among different PoPs
  - We find load-balancing affects 4% of probed targets
  - Critical when *e.g.,* flagging spoofed traffic using catchment data
- Measure ’control’ and ‘experiment’ prefix simultaneously
  - *E.g.,* what if PoP Amsterdam goes offline? What if we prepend our announcement at Frankfurt?
  - Side-by-side comparison of ‘normal’ and ‘varied’ case

Multi-address probing
Prepending de-fra

No prepends (control)



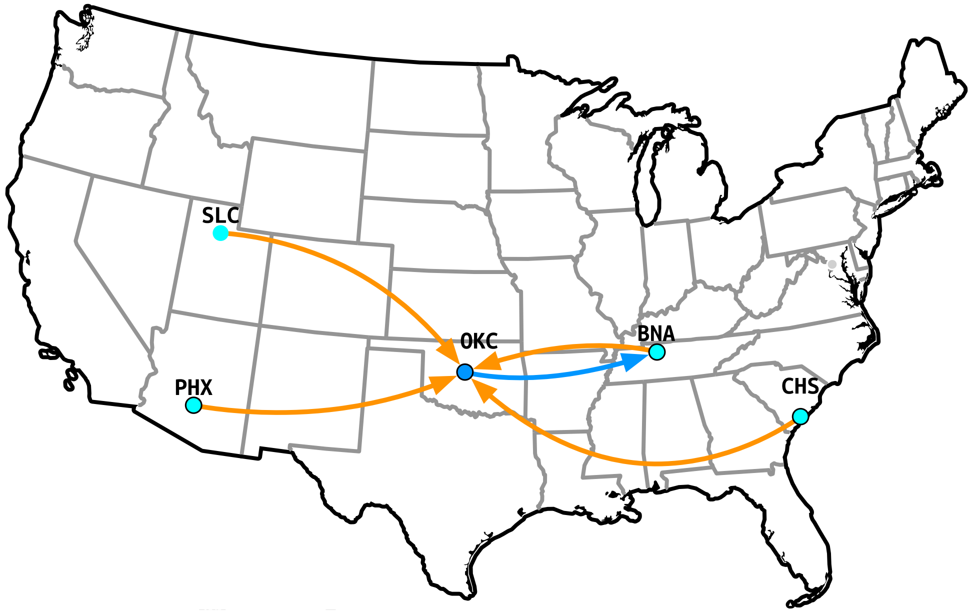
1 prepend (experiment)



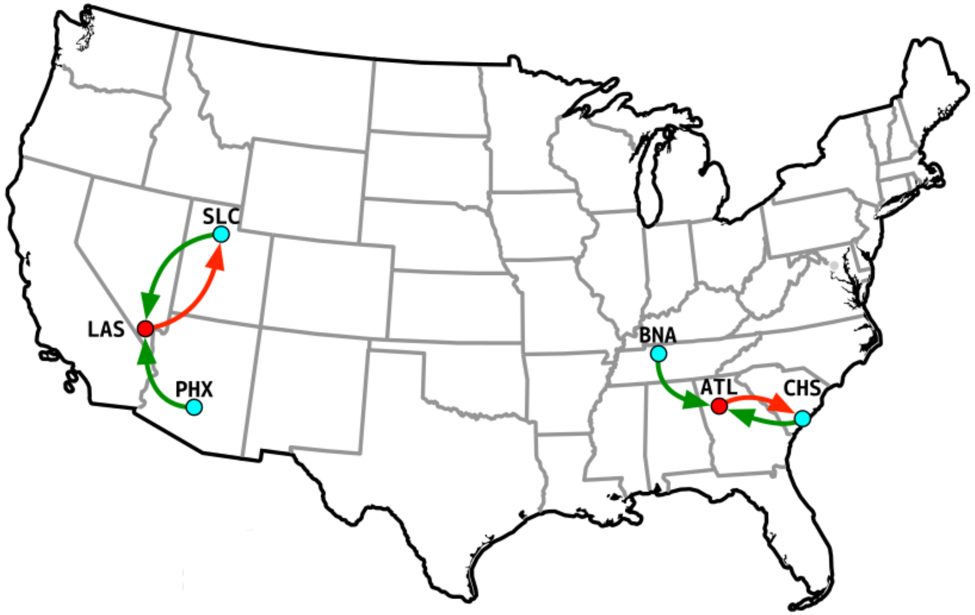
Multi-client probing

- Probe from multiple PoPs simultaneously
- Allows operators to perform their own anycast-census using MAnycast^2 methodology [2]
    - More than one PoP receiving replies -> likely anycast

Unicast



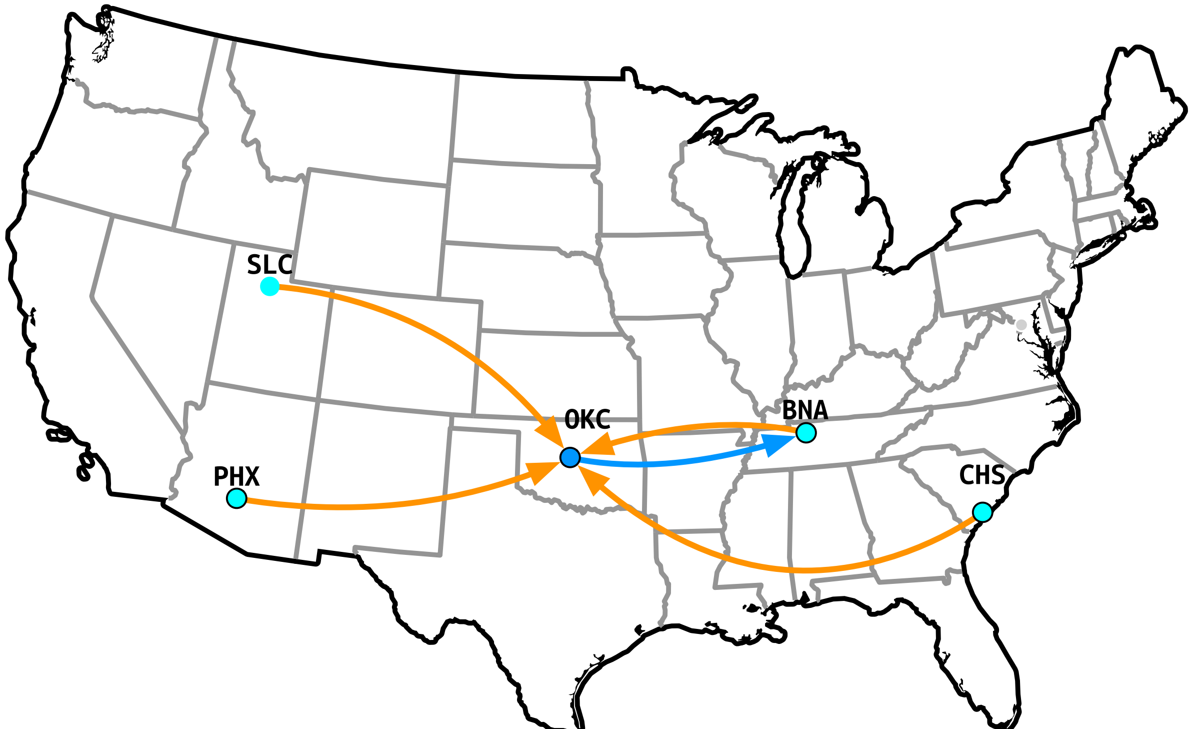
Anycast



[2] Sommese et al. "Manycast2: Using anycast to measure anycast." IMC'20

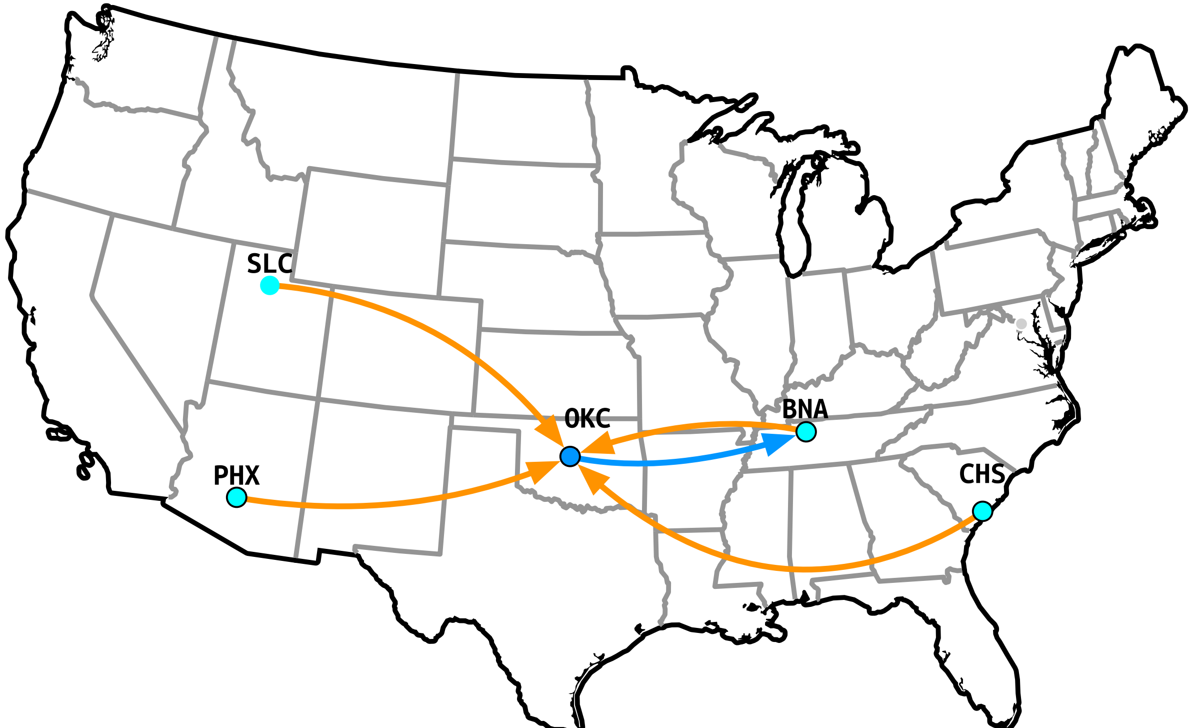
Multi-client probing

- Also, allows for measuring anycast latency
  - Ping one) which PoP does this network route to?
  - Ping two) measure from PoP to network (receiver == sender)

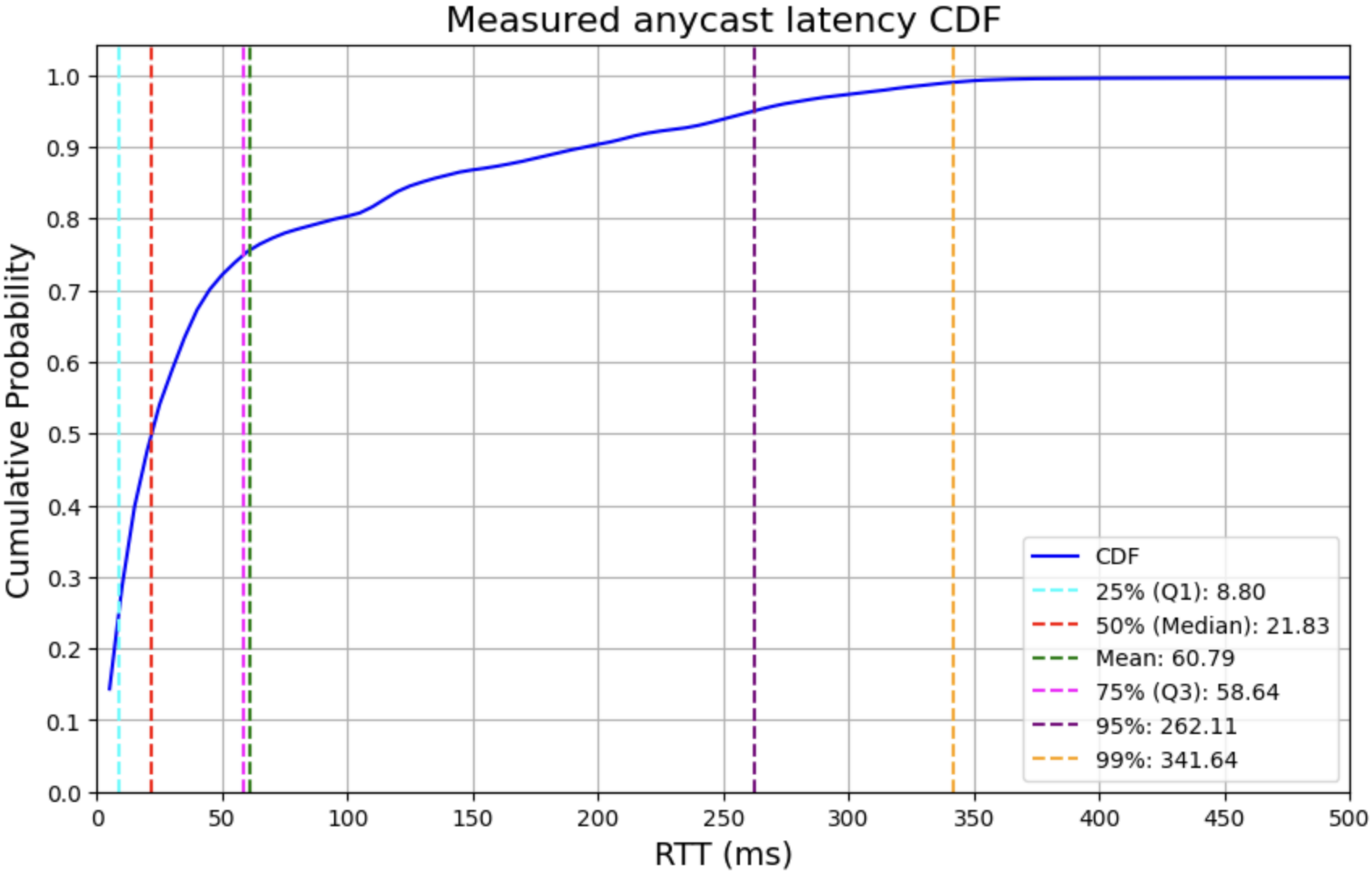


Multi-client probing

- Also, allows for measuring anycast latency
  - ○ Ping one) which PoP does this network route to?
  - ○ Ping two) measure from PoP to network (receiver == sender)
- ○ Latency data validated with passive DNS over TCP traffic (ccTLD)



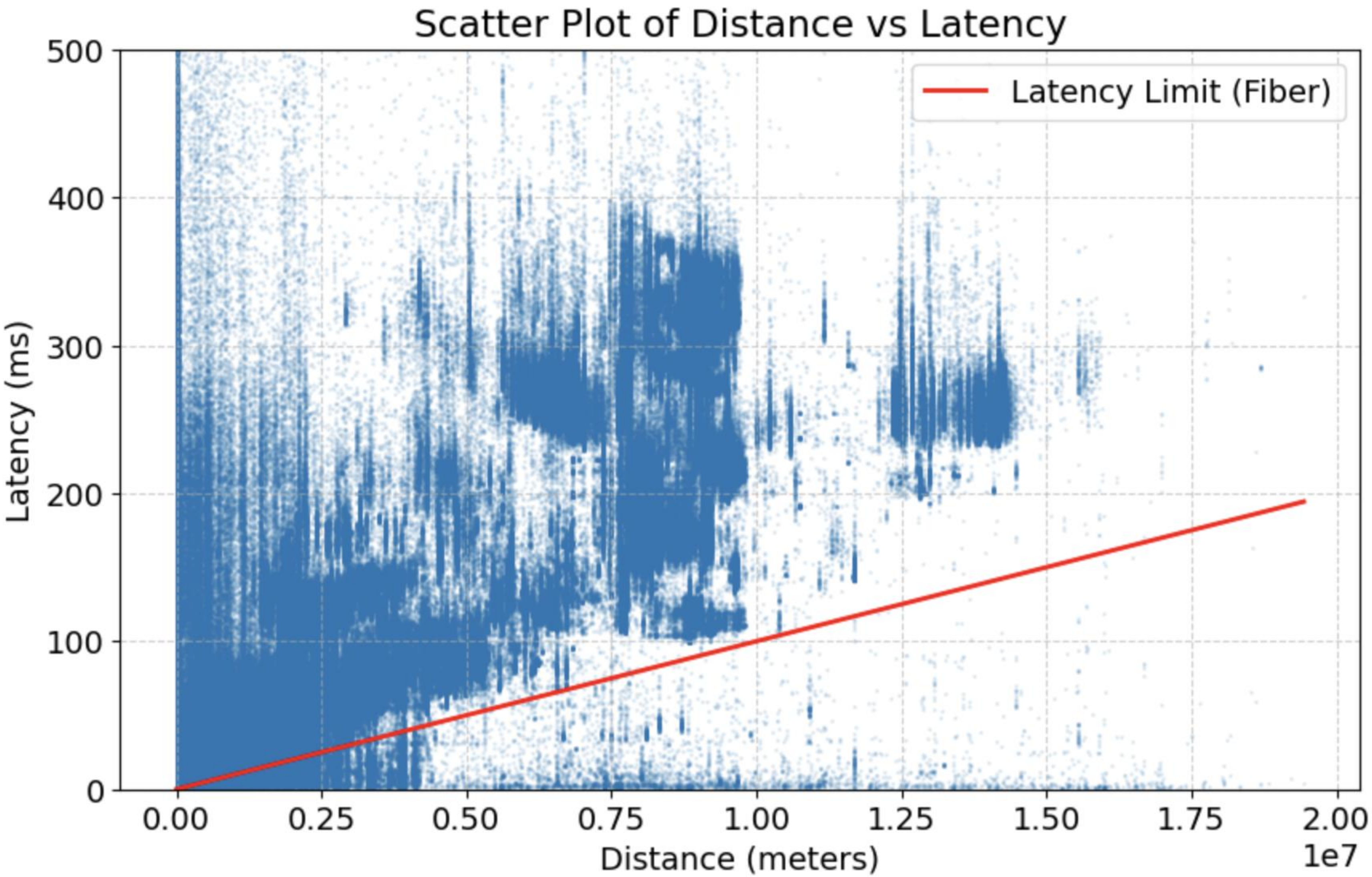
Multi-client probing



Problem with catchment mappings

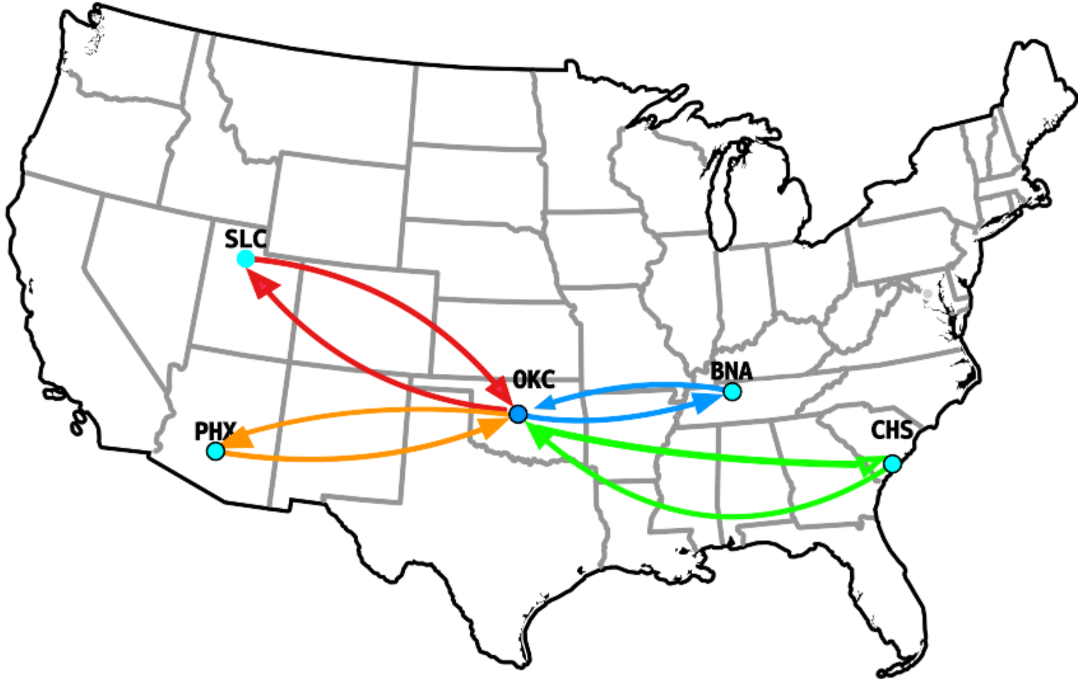
- Latency important metric for most anycast operators
- Catchments can be misleading
  - Geographical proximity does not guarantee optimal routing
  - Client may still suffer from a long path

Problem with catchment mappings

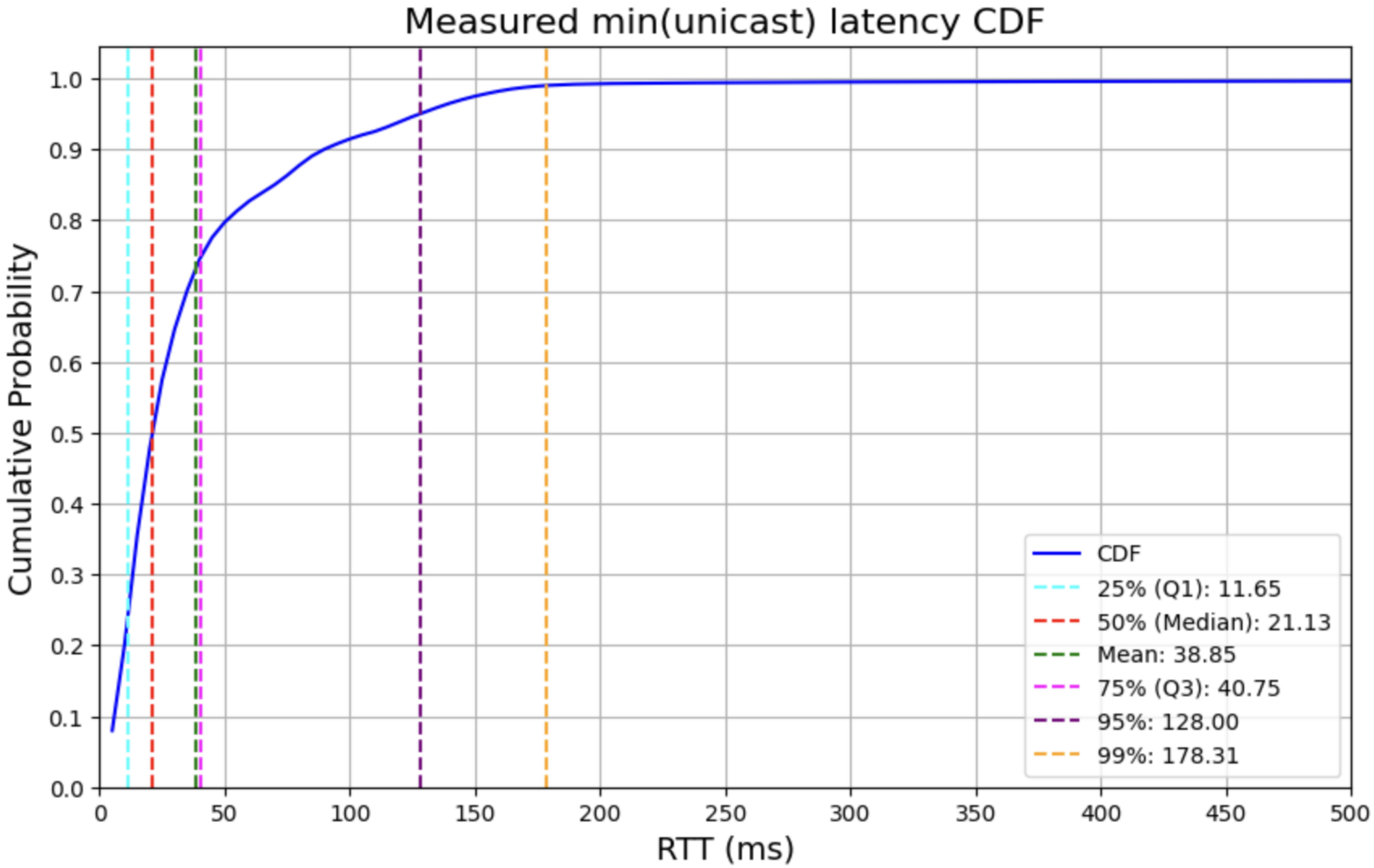


Unicast probing

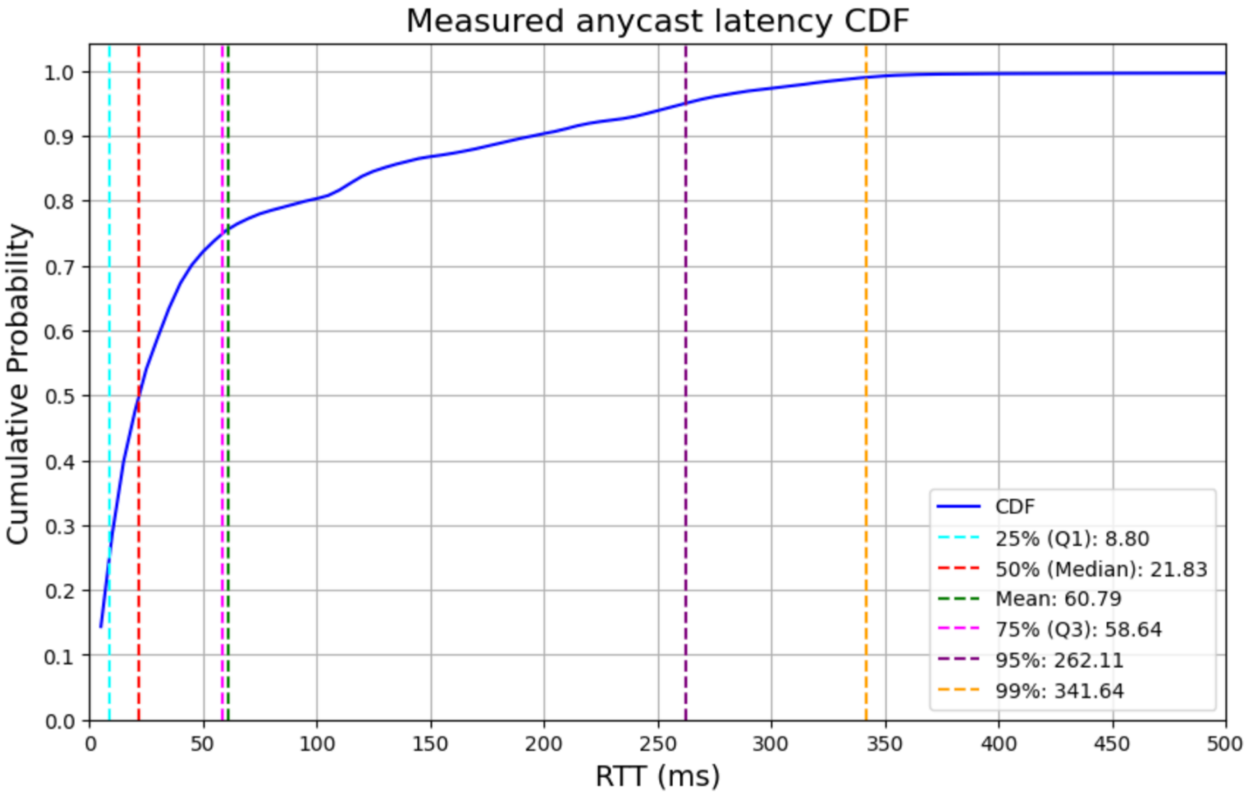
- Allows for probing with unicast IPs
  - Probe target from all PoPs with unicast IP
  - Latency data to all PoPs
  - Obtain nearest (optimal) anycast site based on lowest latency

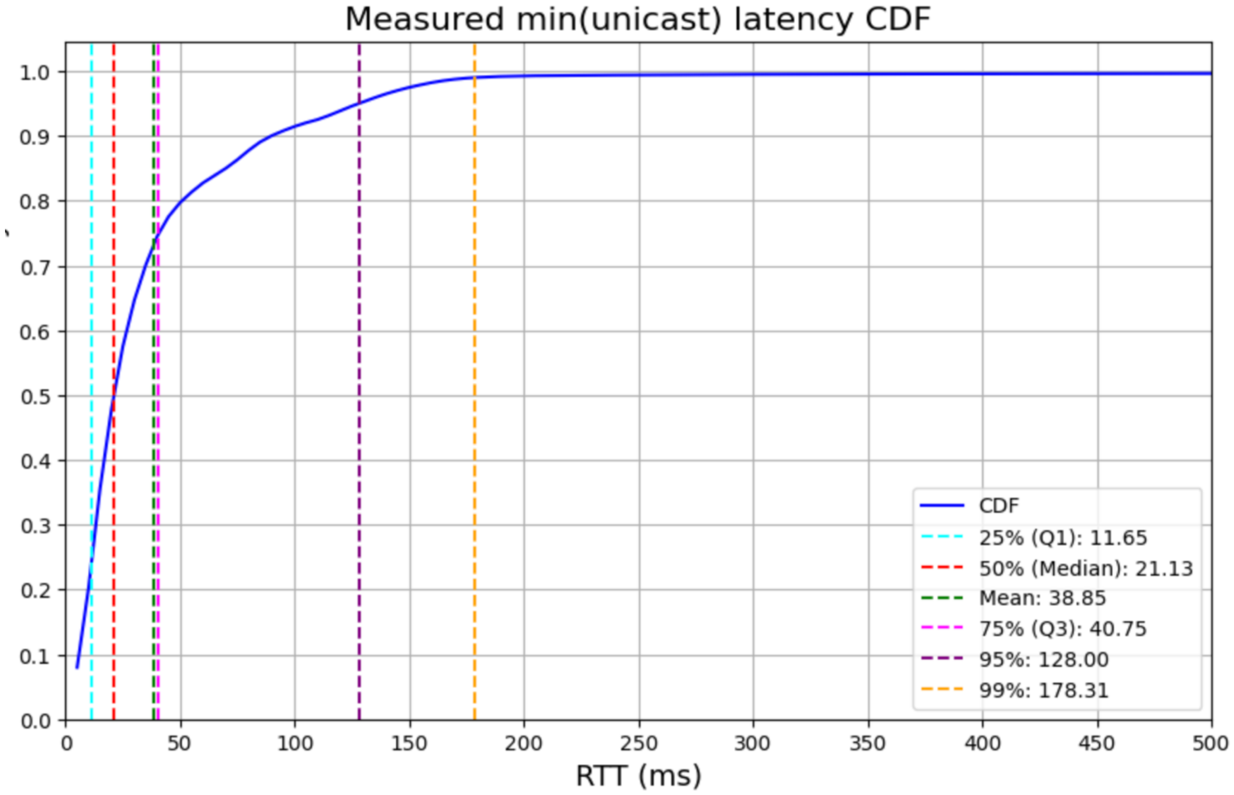


Unicast probing



Comparing latencies





'Optimal' deployment

Site	Mean anycast latency (ms)	Catchment	Mean optimal latency (ms)	Optimal catchment
Frankfurt	83	482k	53	29k
Seoul	63	410k	32	299k
Tokyo	82	322k	66	404k
New York	30	354k	20	302k
Amsterdam	33	143k	28	278k
Atlanta	27	92k	32	199k

Summary

- Tool initially created to measure external anycast deployments (anycast census)
- Now, extended to measure anycast deployments themselves
- Currently used in production for a ccTLD anycast deployment

- Measure actual anycast performance
  - Divide-and-conquer approach to Verfploeter
  - Anycast latency
- Measure ‘optimal’ performance
  - Using unicast latency measurements

Future

- Publicly release tooling
- Paper
- Share tooling with operators
  - (hopefully gain data in return)
- User-friendly dashboard for analyzing results