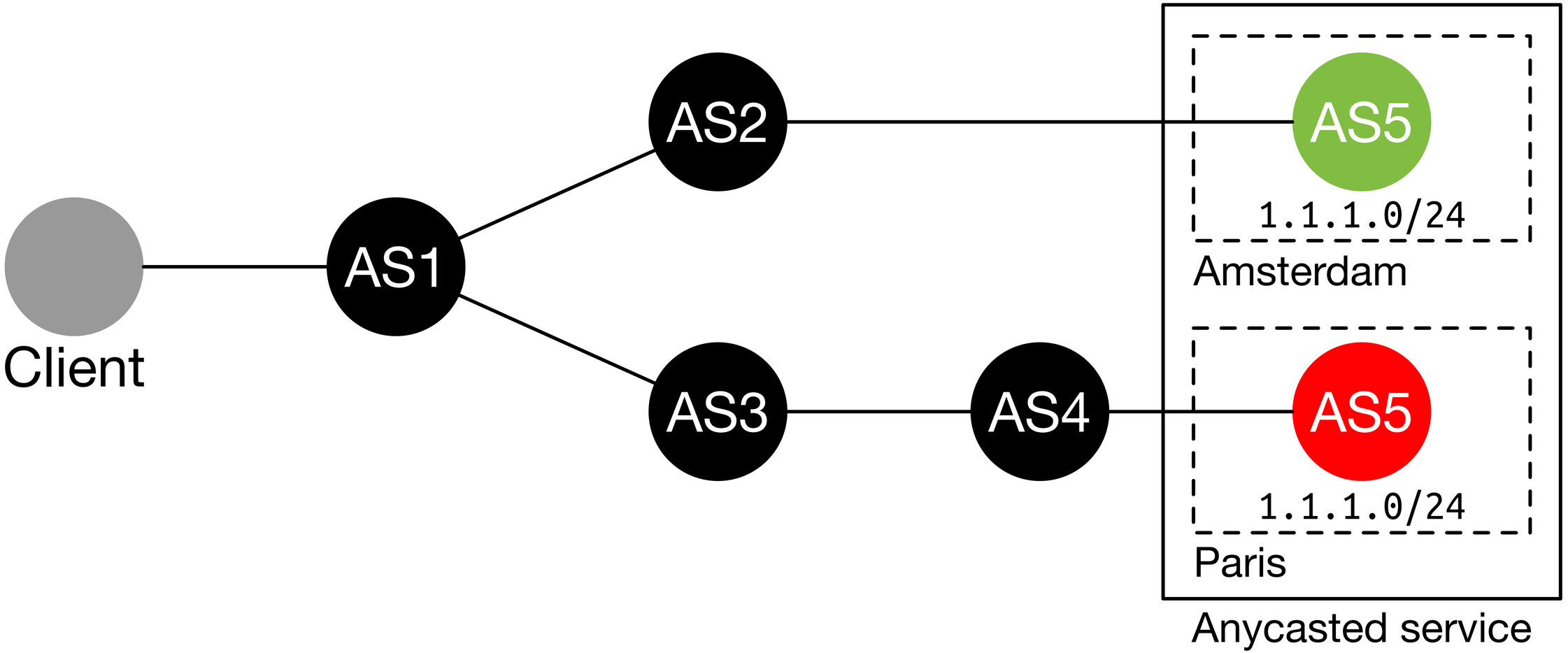
Anycast measurements at CDN scale

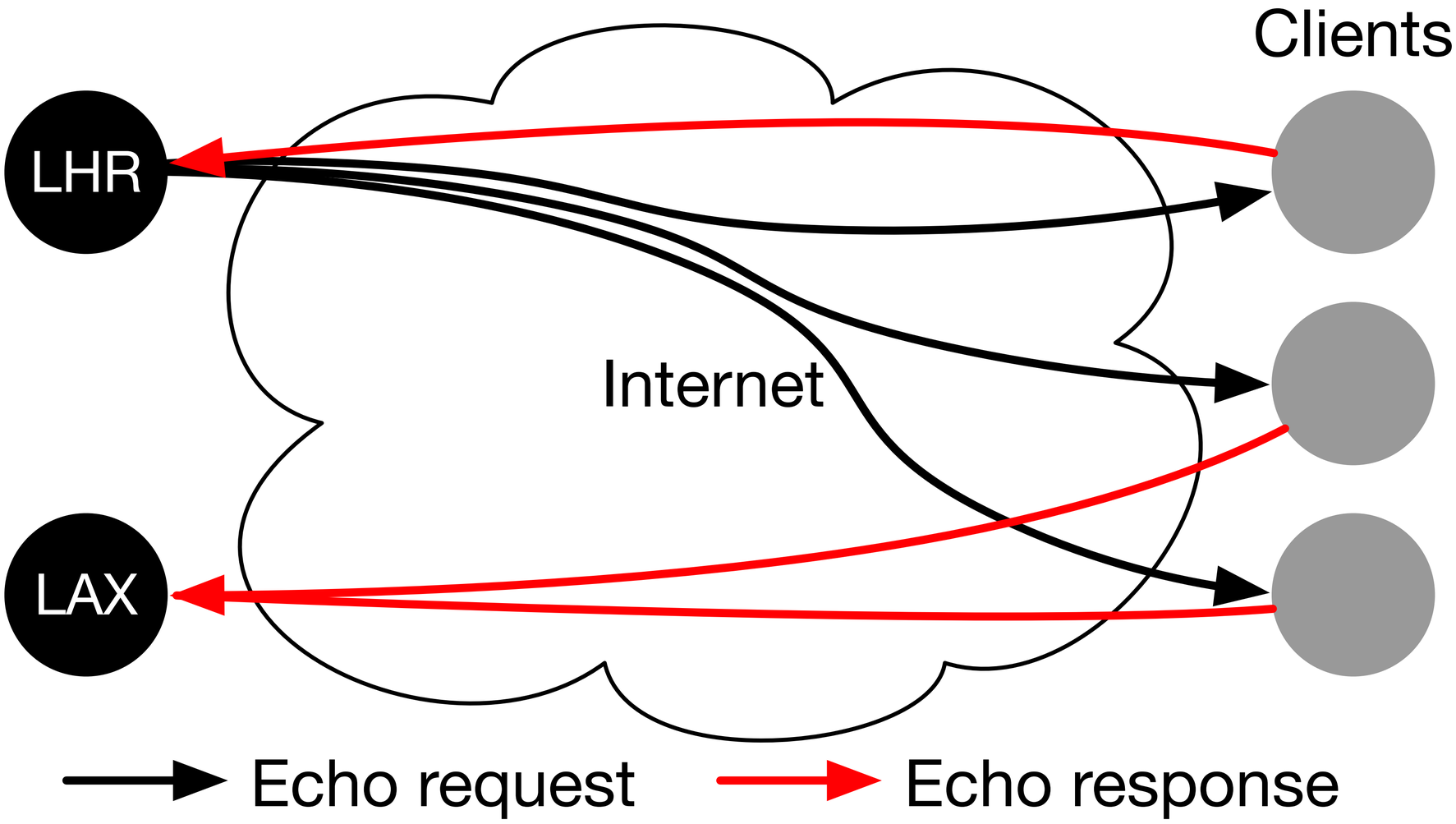
Introduction

- In 2017, we developed "*Verfploeter*", an active measurement method to determine anycast measurements
- *Verfploeter* was tested on the *Tangler* testbed and used to measure the anycast deployment of DNS B Root
- Both of these lacked scale; *Tangler* has 9 PoPs, B Root currently has 3
- In 2018, we deployed *Verfploeter* on a very large CDN with over 190 anycast PoPs; this talk discusses how we did this and what we learned

Anycast in 1 slide



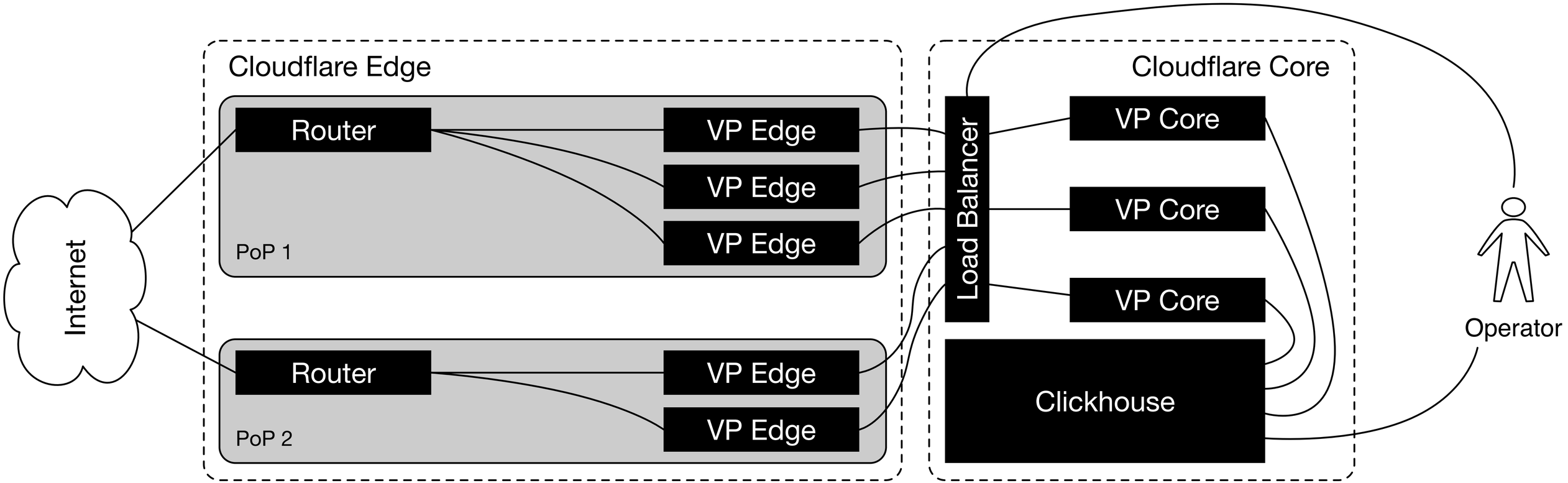
Verfploeter in 1 slide



Cloudflare CDN

- We deployed *Verfploeter* at Cloudflare, a large anycast CDN
- Some numbers on Cloudflare:
  - Over 190 PoPs worldwide
  - Over 30Tbps aggregate link capacity
  - Announcing over 700 prefixes using anycast covering 1.5M+ IPv4 addresses

Deploying *Verfploeter*



Case studies

- We tested *Verfploeter* using three case studies:

  1. Planning anycast maintenance and outages

  2. Identifying and troubleshooting connectivity issues

  3. Detecting spoofed attack traffic

Case 1: planning anycast

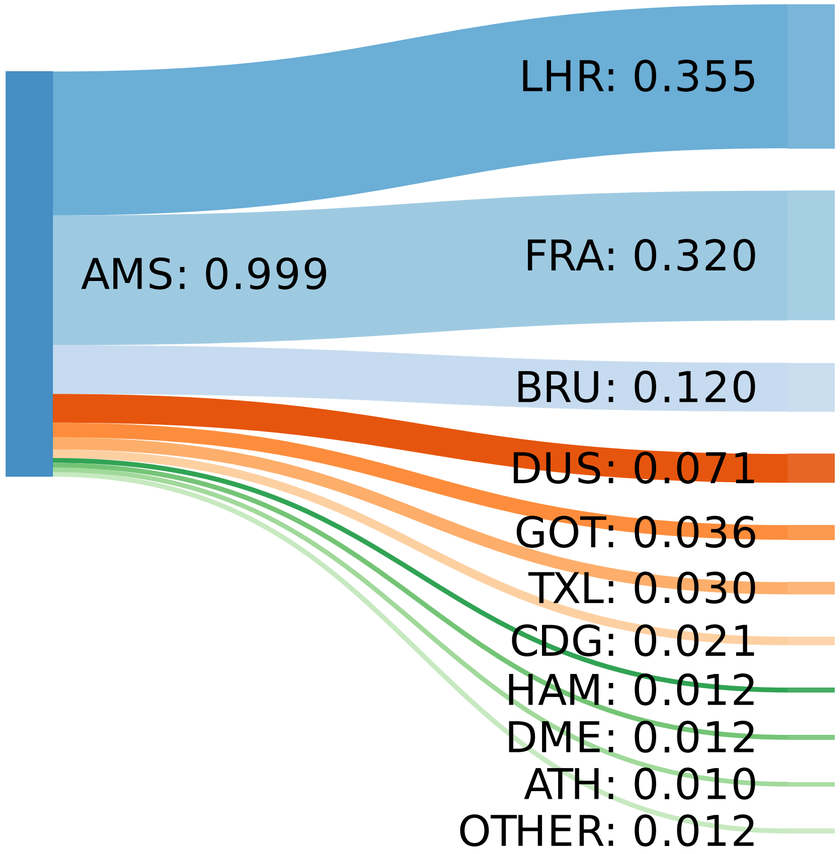
- The goal of anycast is to automatically route traffic to a close (in terms of the network) point of presence
- This balances the load, improves latency and creates resilience against DDoS attacks
- But what if a PoP is down, due to maintenance, service disruption or because it caves in under attack?
- With Verfploeter we can *map and predict where traffic goes* if a PoP is down

PoP takedown measurement

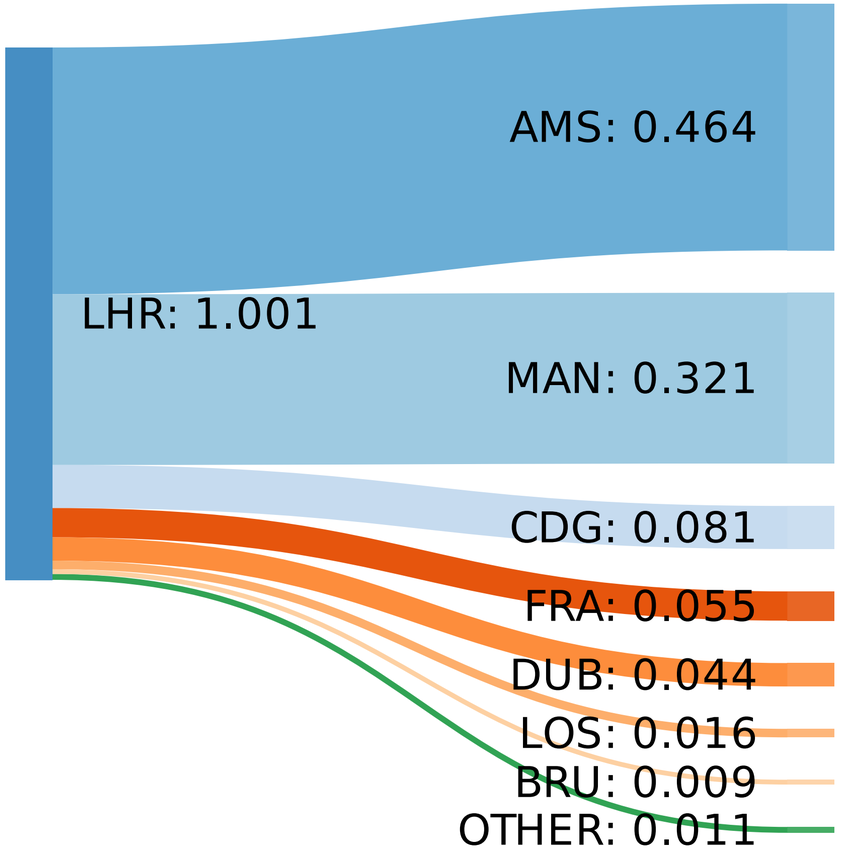
#	PoP(s) offline	Count	Response fraction
P0	*None*	3.49M	0.57
P1	AMS	3.44M	0.56
P2	LHR	3.30M	0.54
P3	CDG	3.42M	0.56
P4	AMS, LHR	3.45M	0.56
P5	AMS, CDG	3.50M	0.57
P6	one per measurement 182 measurements different PoP each measurement	≈3.5M	≈0.55

takeaway: response rate not affected by PoP down

Single PoP down



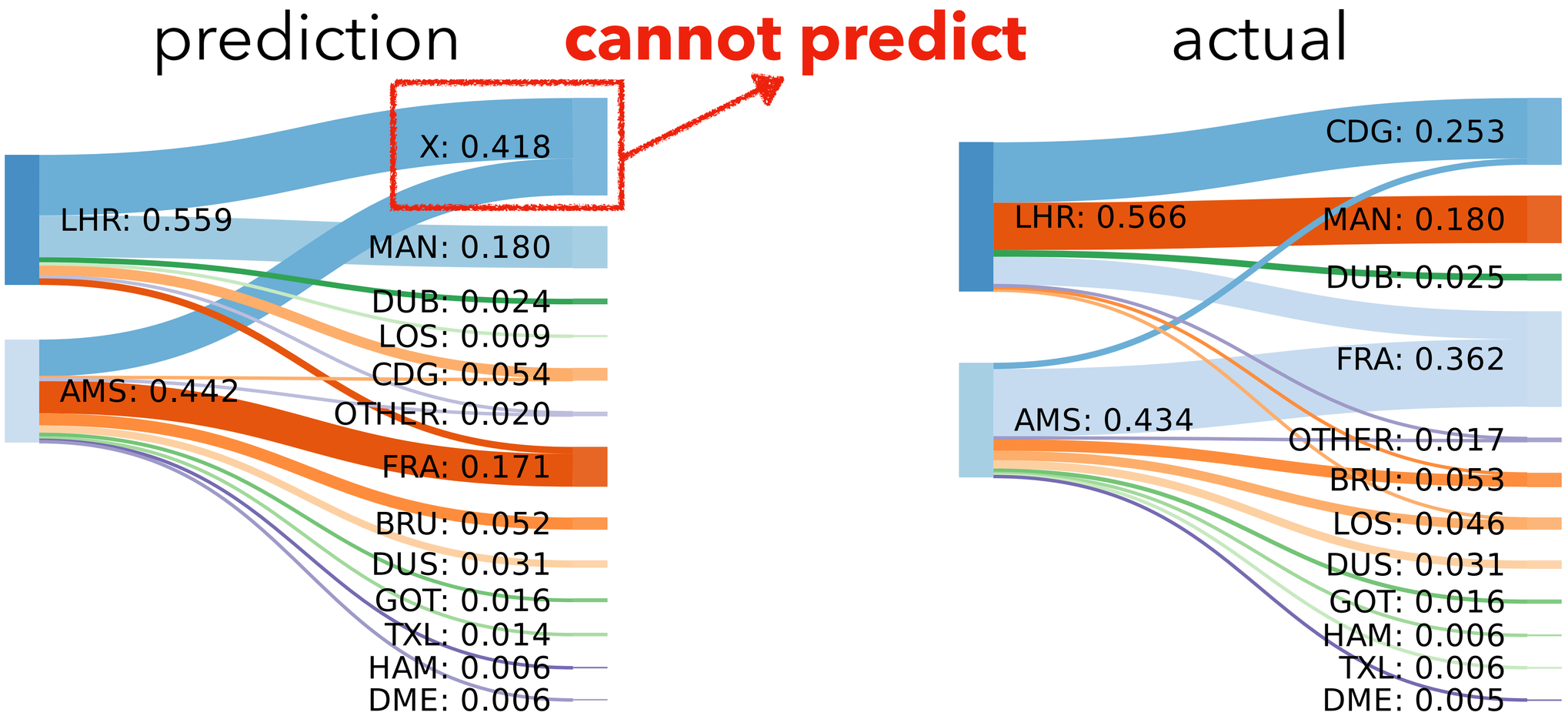
AMS down



LHR down

takeaway: most traffic re-routes to few close PoPs

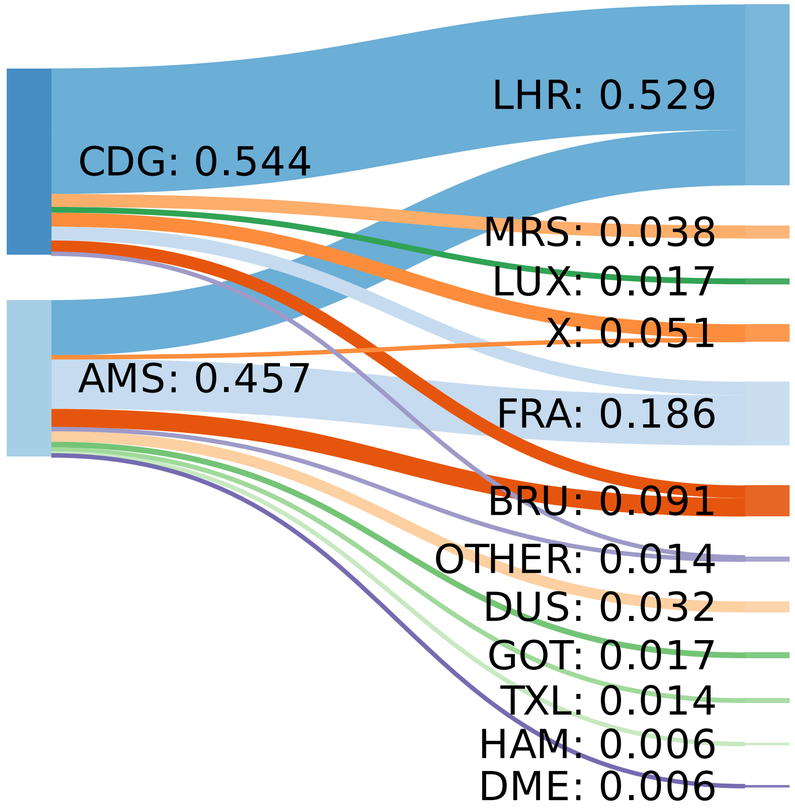
Two PoPs down (1)



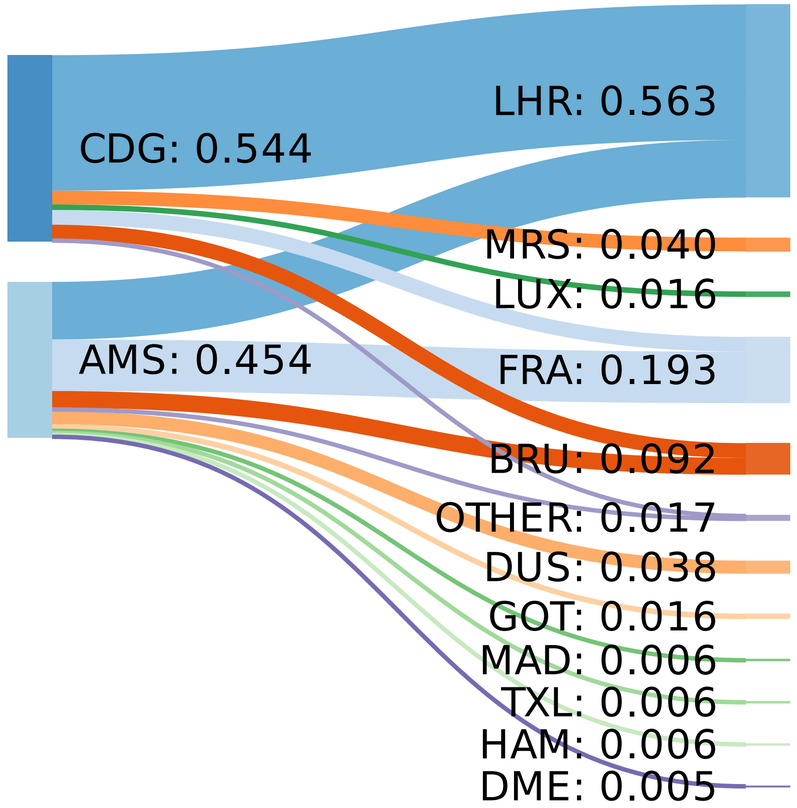
takeaway: prediction hard with dependent PoPs

Two PoPs down (2)

prediction



actual



takeaway: prediction very accurate

Case 2: troubleshooting

- For unicast we commonly use "ping" to troubleshoot connectivity issues
- With anycast, this is, of course, no longer possible
- With *Verfploeter*, however, we can restore this capability
- Cloudflare operates the 1.1.1.1 public DNS resolver; given the "special" nature of this address, and its use in default and example configurations, there were connectivity issues
- We analysed these using *Verfploeter*

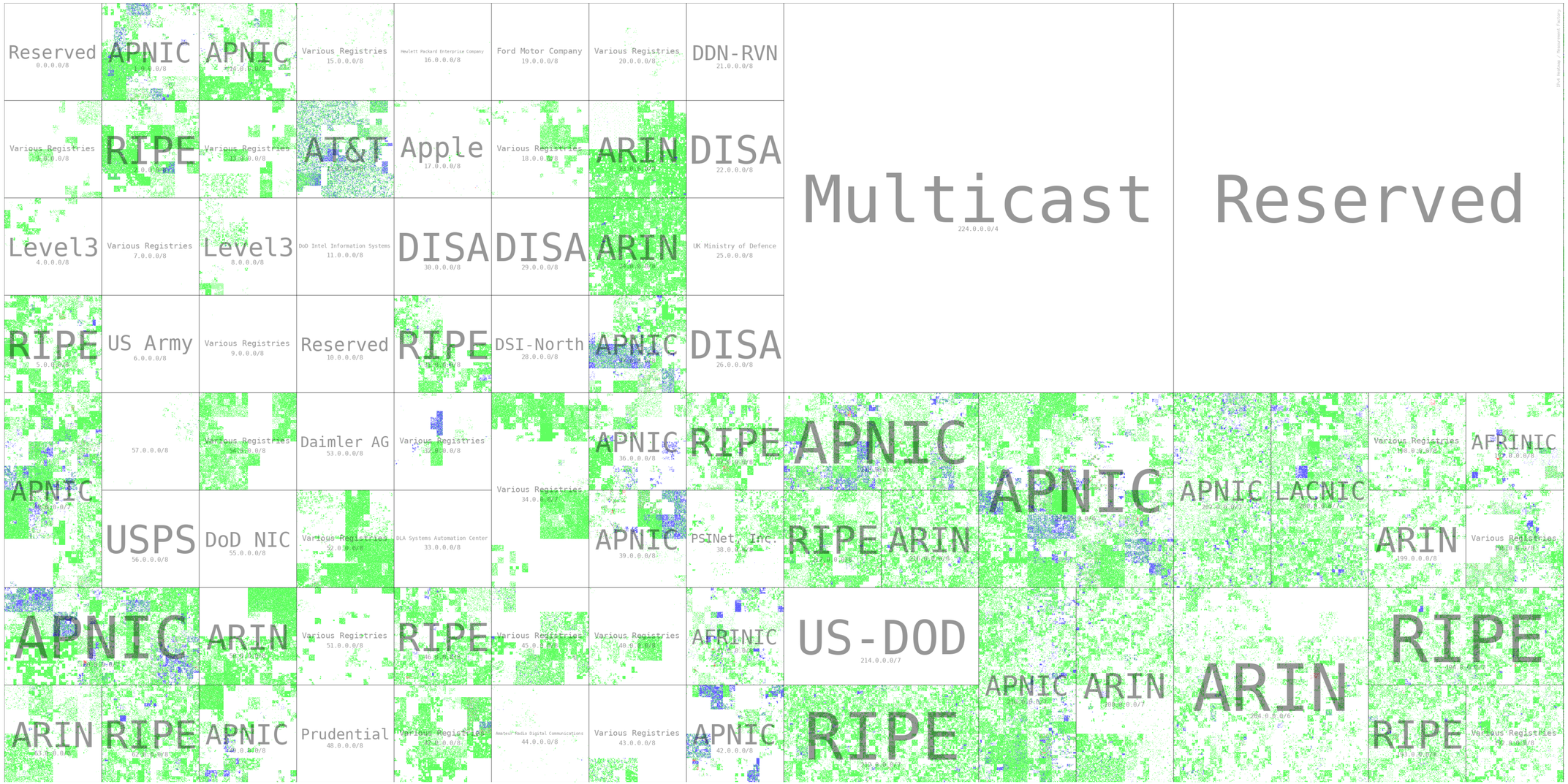
Reachability comparison

Source	Count	Response fraction
1.0.0.1	3.47M	0.56
1.0.0.1	3.49M	0.57
1.1.1.1	3.28M	0.53
1.1.1.1	3.28M	0.53
104.23.98.190	3.48M	0.57
104.23.98.190	3.5M	0.57
Combined		
1.0.0.1	3.58M	0.58
1.1.1.1	3.36M	0.55
104.23.98.190	3.59M	0.58

IPs	Count	Fraction
1.0.0.1, 1.1.1.1, 104.23.98.190	3,324,062	0.917
1.0.0.1, 104.23.98.190	232,160	0.064
104.23.98.190	18,526	0.005
1.1.1.1, 104.23.98.190	17,508	0.005
1.0.0.1, 1.1.1.1	16,473	0.005
1.0.0.1	8,125	0.002
1.1.1.1	6,707	0.002

takeaway: only 1.1.1.1 suffers significant reachability issues

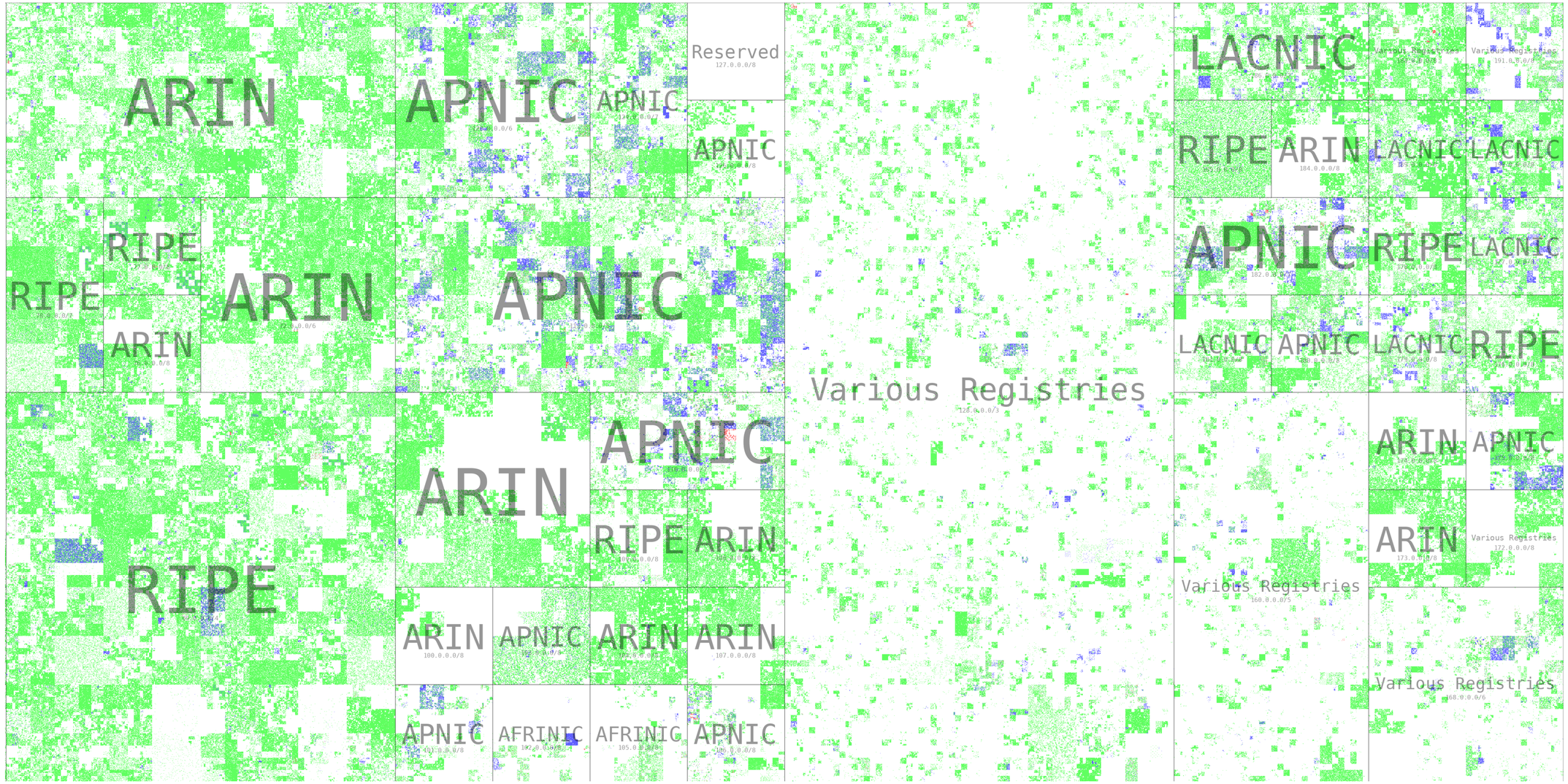
1.1.1.1 vs. 1.0.0.1



blue = 1.0.0.1 reachable but 1.1.1.1 is not

UNIVERSITY OF TWENTE.

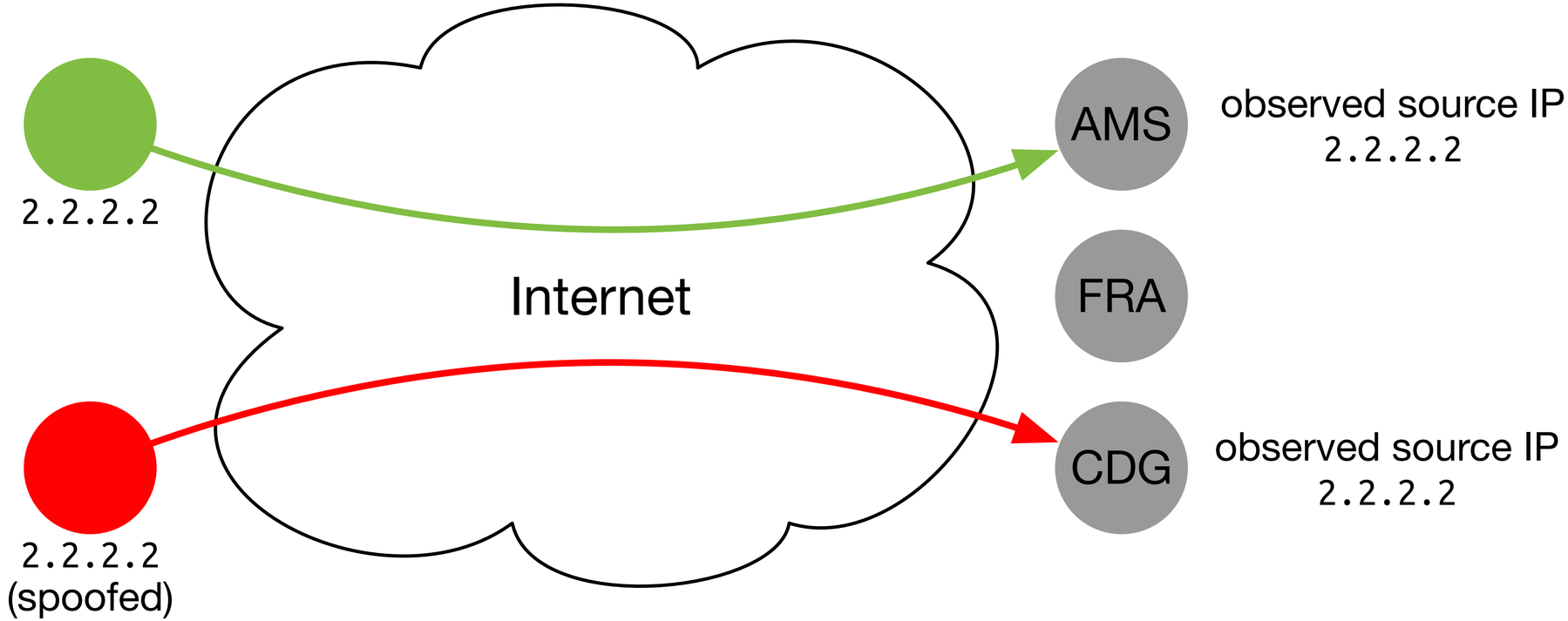
1.1.1.1 vs. 1.0.0.1



blue = 1.0.0.1 reachable but 1.1.1.1 is not

UNIVERSITY OF TWENTE.

Case 3: spoofed DDoS

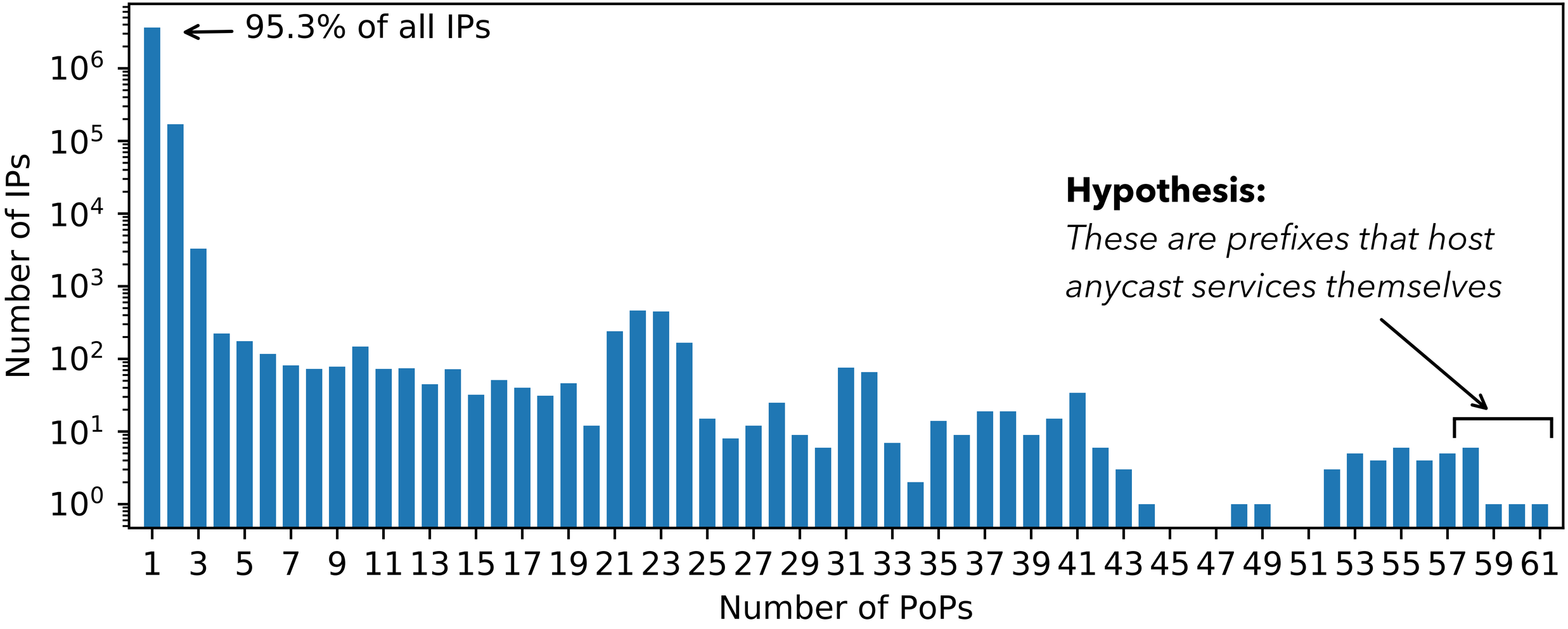


- Hypothesis:
  We can detect spoofed traffic because most of it will arrive in a PoP where we are not expecting to receive it, due to routing from spoofing source

Side-step: PoP affinity

- Before we can test this hypothesis, we need to verify two assumptions
- First, we need to check that the (vast) majority of /24 prefixes are consistently routed to the same PoP
- Second, we need to check that /24 prefixes are routed to the same PoP, regardless of the origin PoP of the *Verfploeter* measurement

Measured PoP affinity



(based on 191 measurements, each from a different PoP)

Detecting anycast

- Top 9 prefixes from long tail of figure on previous slide:

AS8068 = Microsoft, AS26415 = ICANN, AS42 = PCH

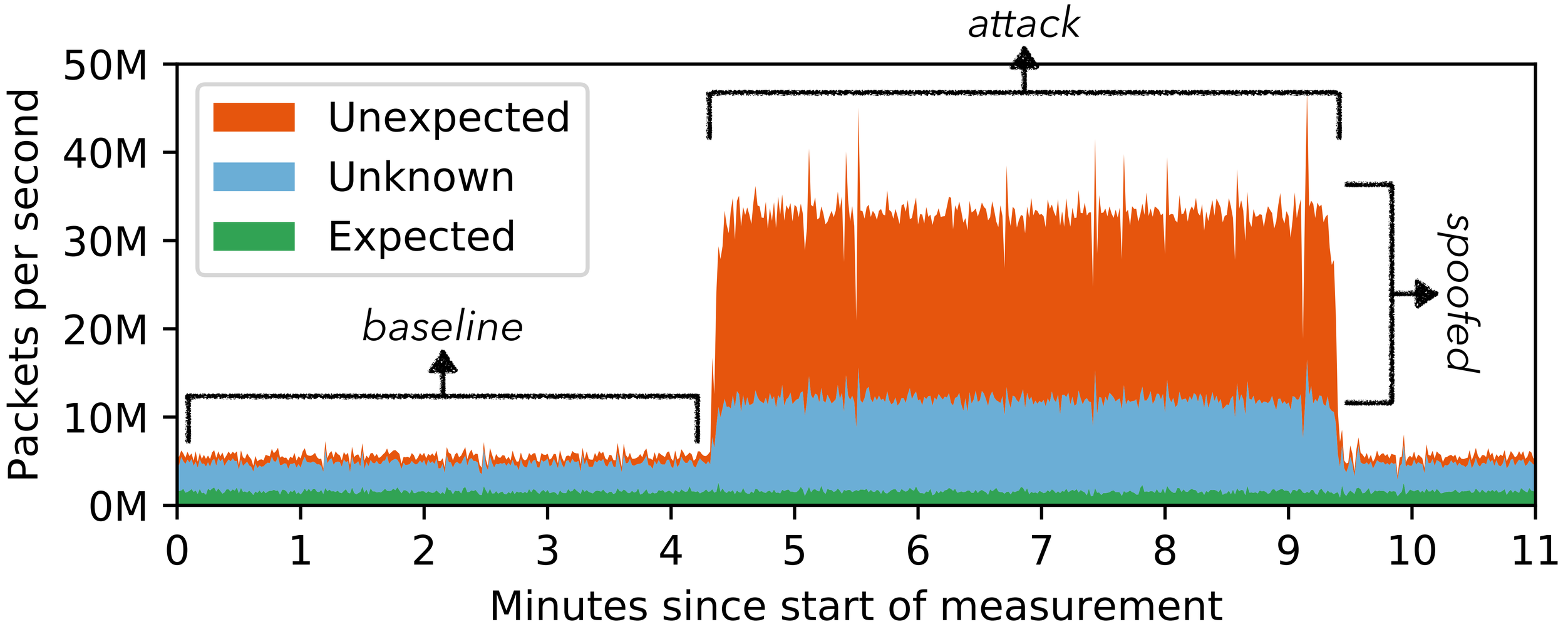
Prefix	PoPs	ASN	Rev. Hostname
192.58.128.0/24	61	26415	j.root-servers.net
204.61.216.0/23	60	42	ns.anycast.woodynet.net
192.33.14.0/24	59	26415	b.gtld-servers.net
189.201.244.0/23	58	42	e.mx-ns.mx
204.19.119.0/24	58	42	c.ns.apple.com
200.108.148.0/24	58	42	c.dns.ar
206.51.254.0/24	58	42	lns61.nic.tr
13.107.4.0/24	58	8068	ns1.c-msedge.net
194.0.17.0/24	58	42	e.nic.ch

- Spoiler alert: all of these are anycast services

Detecting spoofed DDoS

- To recap, we now know that we can expect traffic from the vast majority of prefixes to ingress the Cloudflare network at a single PoP
- This means we can test our hypothesis that we can identify spoofed traffic based on mapped anycast catchments
- To test this, we applied the method to a real spoofed SYN-flood attack that lasts for 11 minutes and generates almost 50Mpps at its peak

Detection results



Conclusions

- We showed that *Verfploeter* scales to large-scale global deployments
- It adds value for large operators; Cloudflare is now using this in production
- We demonstrated that we can accurately detect a significant fraction of the spoofed traffic in an attack based on expected anycast catchments (this is a highly significant result that we intend to study further)

Further reading

- Paper to be presented at NOMS 2020 (20-24 April, Budapest, Hungary)
- Contact us if you want a pre-print
- More info on the SAND project: https://www.sand-project.nl

Global-Scale Anycast Network Management with Verfploeter

Wouter B. de Vries
Design and Analysis of Communication Systems
University of Twente
Enschede, The Netherlands
w.b.devries@utwente.nl

Salman Aljammaz
Cloudflare
s@cloudflare.com

Roland van Rijswijk-Deij
Design and Analysis of Communication Systems
University of Twente
Enschede, The Netherlands
r.m.vanrijswijk@utwente.nl

***Abstract*—Anycast has become a valuable tool for network operators. It plays a vital role in making the DNS root system globally highly available and resilient to stresses from e.g. DDoS attacks. Content delivery networks use it to direct clients to local caches, and to absorb attack traffic. Yet managing an anycast network is far from simple. Earlier work studying a DDoS attack on the DNS root system, for example, shows that even highly distributed anycast networks can be overwhelmed.**

To manage an anycast service, it is vital to know the catchment of points of presence (PoPs) of the service. In earlier work, we introduced "Verfploeter" a novel active measurement method to determine anycast catchments using ICMP messages. Unlike previously existing approaches, Verfploeter is unbiased, accurate and can be executed directly by the anycast operator without the need for external vantage points. We demonstrated the efficacy of Verfploeter on a testbed and small anycast service.

In this paper, we take the next step and deploy Verfploeter on one of the world's largest anycast networks, the Cloudflare CDN with 192 PoPs worldwide. We perform three real-world case studies on network planning (what happens when PoPs are switched on or off), troubleshooting (reachability issues of an anycasted prefix) and security (detecting spoofed attack traffic). Using these three case studies, we show that Verfploeter is highly suitable for such a large-scale operation and gives operators vital insights that allow them to improve network management practices of their anycast service.

***Index Terms*—Anycast, Routing, Measurements, Active, Monitoring, BGP, Security, Troubleshooting, Network Planning**

I. Introduction

Service operators use IP anycast to provide increased resilience, lower latency, and increased throughput for their services. Anycast is a technique, enabled by BGP, that allows physically and geographically distinct systems to be addressable with a single IP-address/IP-prefix. This allows services to be scaled horizontally at different locations, by adding more and more systems.

Examples of services that make use of anycast are the DNS root servers and Country-code Top Level Domain (ccTLD) DNS servers (e.g. `.nl`). Historically it was assumed that anycast is only suitable for connectionless protocols, since each packet can potentially reach a different anycast instance. DNS, largely dependent on UDP, is therefore a suitable candidate for anycasting. It has since been shown that Internet routing is stable enough to allow anycast to work for both connectionless and connection-oriented protocols, such as TCP [1], [2]. Nowadays, many large Content Delivery Networks (CDNs) also utilize anycast, such as Microsoft/Bing, Verizon/Edgecast, Akamai, and Cloudflare.

In earlier work [3] we introduced a novel methodology to measure the catchments (i.e. which client will be served by which site) of anycast services, called "Verfploeter". Key advantage of this methodology is that it does not require any external Vantage Points (VPs) such as RIPE Atlas probes, but instead relies on ICMP-responsive Internet hosts. By sending ICMP Echo Requests to many hosts on the Internet, and collecting the responses, we can accurately establish the catchment of a service for the full IPv4 Internet, or a part thereof. Unlike an approach based on external vantage points, Verfploeter does not suffer from bias due to the distribution of these points.

In previous work, we showed how Verfploeter performs from a deployment on a testbed, and, on a limited scale, the B root DNS server (which has just three anycast sites). In contrast, in this paper we describe a global-scale deployment in one of the worlds largest anycast CDNs. We discuss the challenges of deploying Verfploeter in an anycast network of this scale (with 192 global points-of-presence). Then, we show how Verfploeter can help large-scale anycast operators manage their network through three use cases:

Firstly, we show how Verfploeter's detailed catchment information helps manage changes in the configuration of the active sites of an anycast service. For example, what would happen if large site A is taken down, in terms of the shift in clients to other sites. We argue that this is important since depending on the shift of traffic, one or more of the other sites might attract traffic exceeding its maximum capacity. This is also particularly useful for planned maintenance.

Secondly, we show how Verfploeter can be used to regain traditional ICMP-based troubleshooting capabilities. For example, traditionally connectivity issues are confirmed using `ping`, i.e. by sending an ICMP Echo Request packet. However, in the case of anycast, the response to this packet will likely end up in a different location. From the viewpoint of the sender of the request packet this would appear as a timeout. Using Verfploeter these packets are matched regardless of the location where it is received, in essence allowing an asymmetric ping.

Lastly, we show how Verfploeter can be used to detect

Questions?