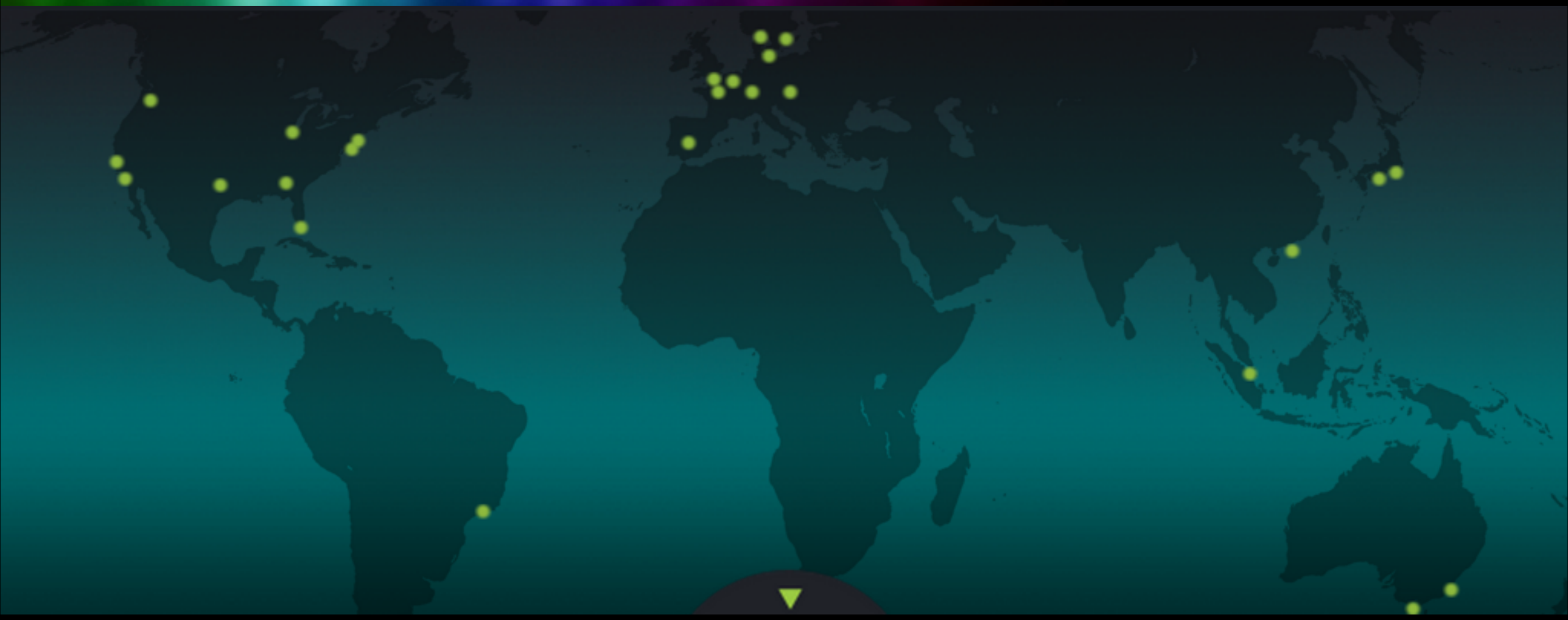
IPv6 and Path MTU problems in AnyCast networks

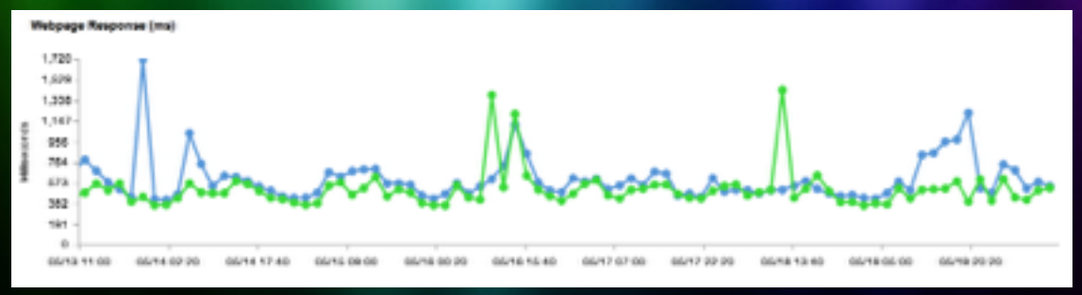
Hossein Lotfi
Director of Performance Engineering
EdgeCast Networks Inc.

About EdgeCast



- Performance Oriented Content Delivery Network with world-wide presence
- HTTP Caching Platform for Static Content
- Application Delivery Network for Dynamic Content (Lots of TCP Optimizations)
- Streaming
- DNS Platform

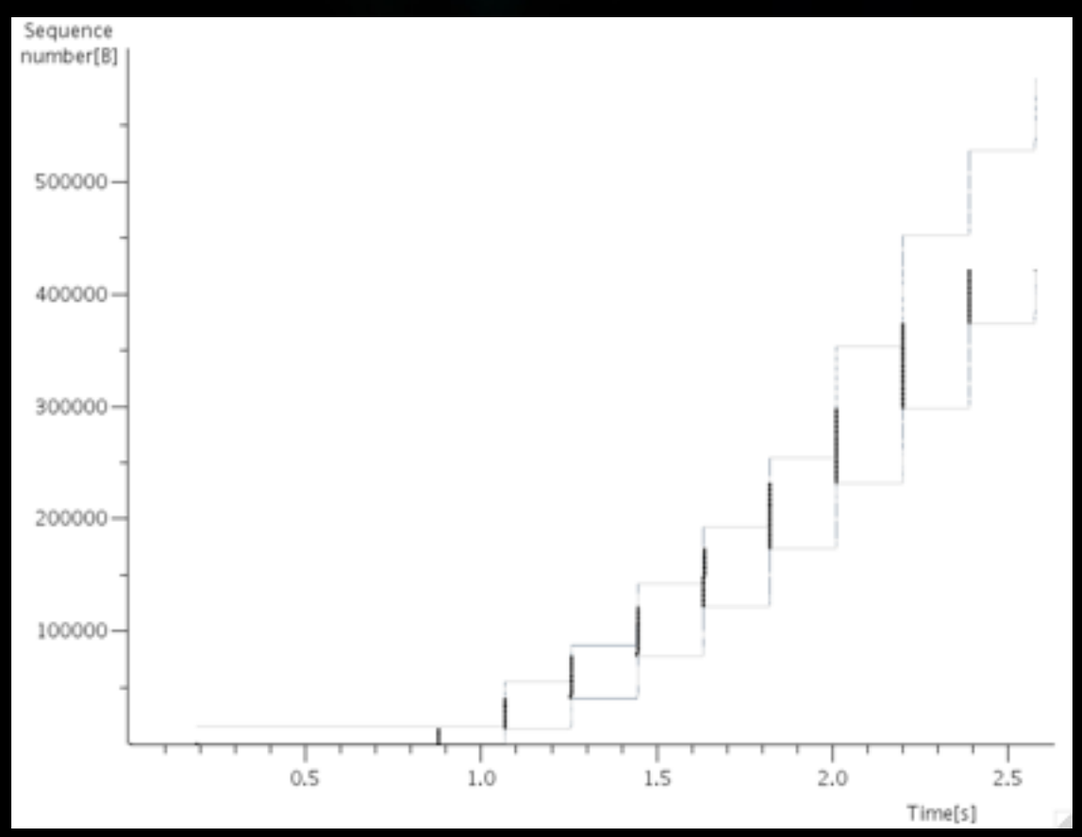
What performance team does at EdgeCast



Analyze Performance Trends and come up with ideas to make the CDN faster

Performance Optimization of:

- **Network Stack**
- **Routing**
- **Kernel**
- **Application**



TCP Optimizations

we work on the most complex cases. they usually mean find needle in a haystack

We will talk about:

- In the first half of this presentation we will explain How we tested our network prior to IPv6 Launch
- Broken flows that we discovered and signs of a bigger problem with AnyCast flows
- Our troubleshooting process
- Possible solutions

Testing methodology:

- Synthetic monitoring
- Real User monitoring
  - RIPE Atlas
  - Internal RUM
  - TCP_INFO

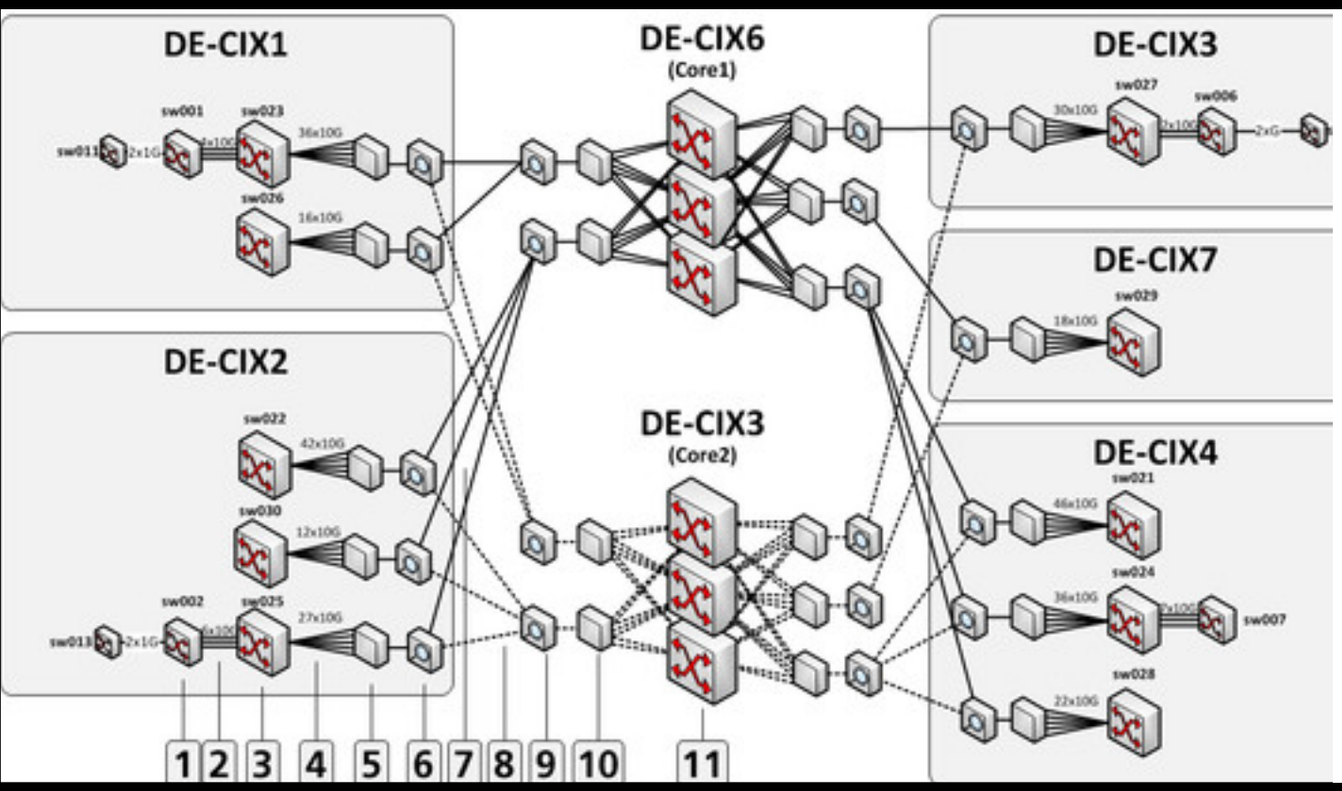
6/6/12

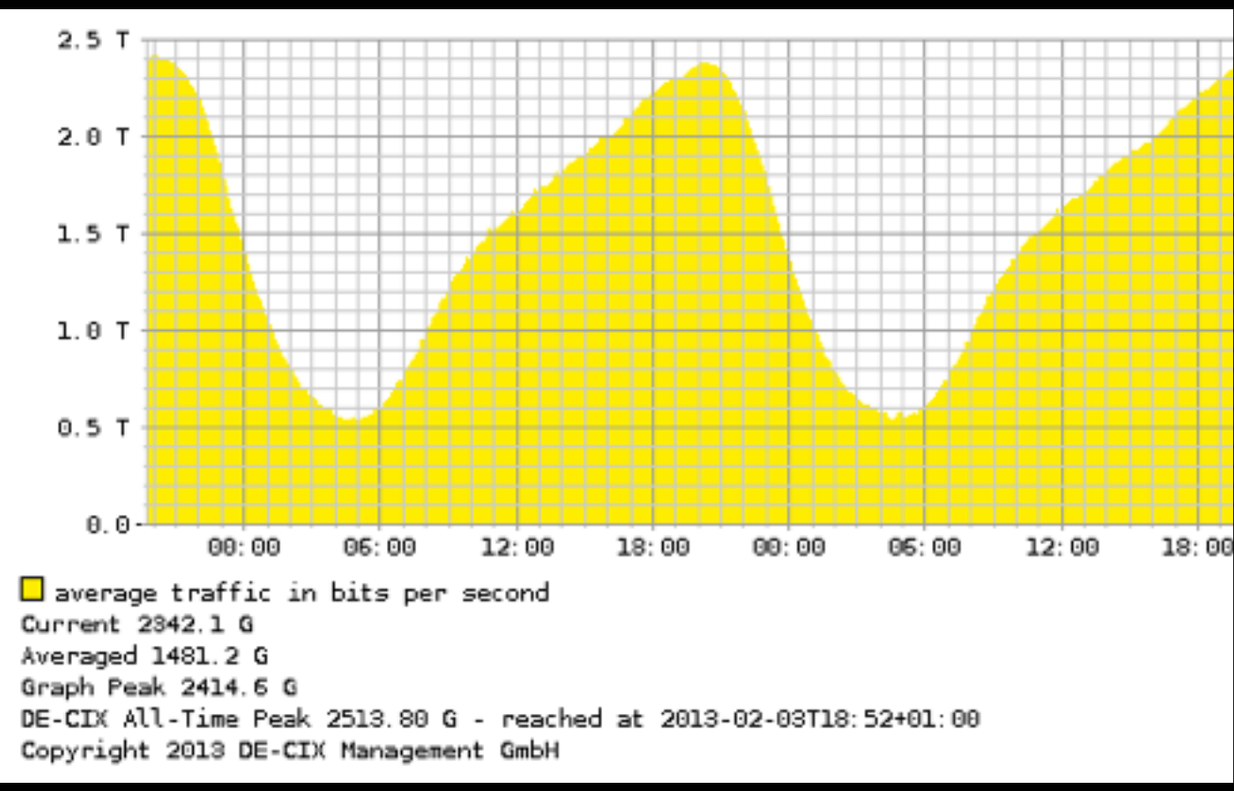
Synthetic monitoring

- Servers in DataCenters
- Strategically located
- Very well peered and monitored

Synthetic nodes that were available to us

Synthetic nodes are usually connected to main Internet Exchange Centers



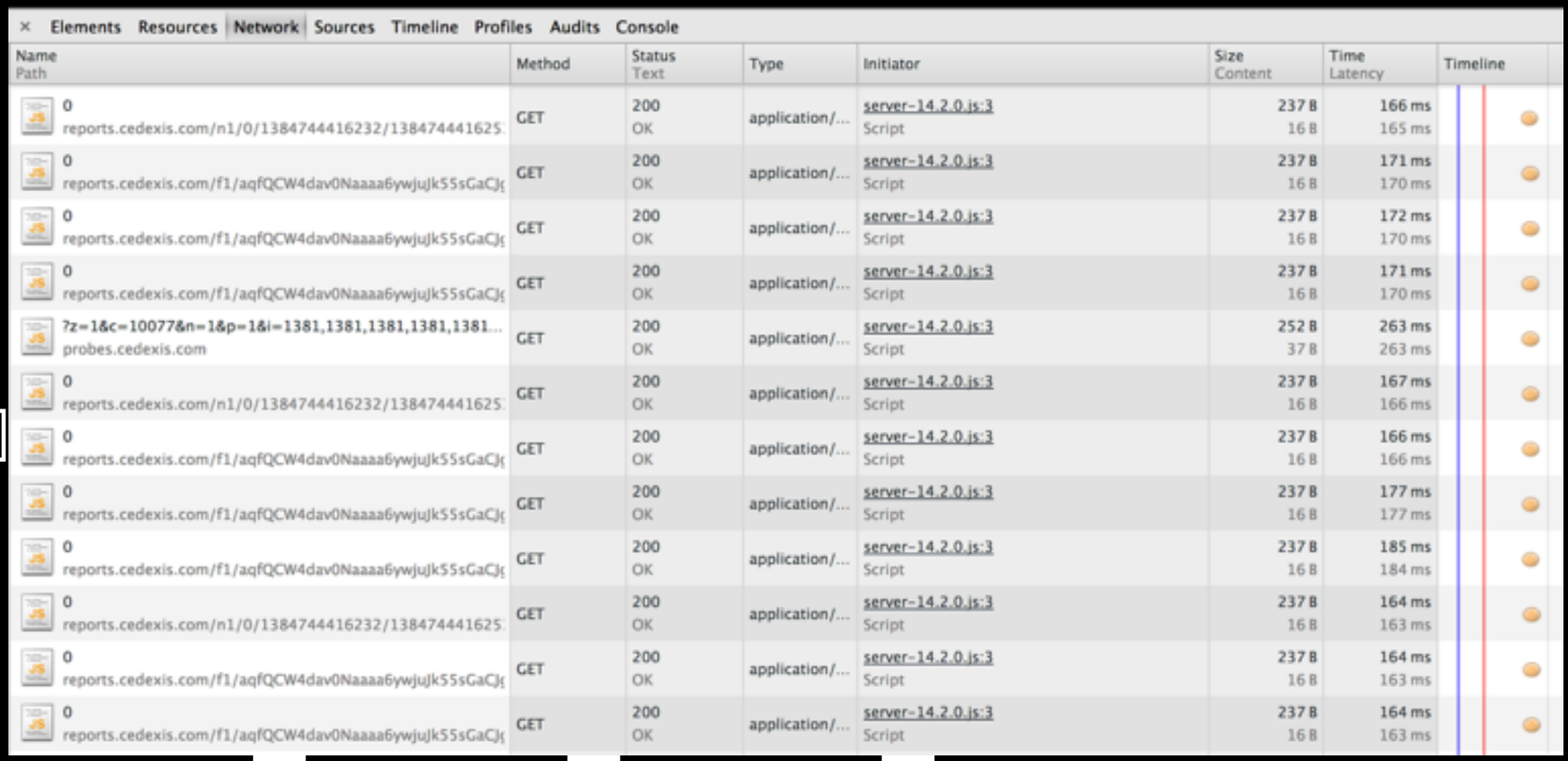


One Synthetic Failure is Too Many !

Prior to world IPv6 launch date, none of Synthetic nodes that we worked with had reliable v6 connectivity

RUM (Real User Monitoring)

How does RUM work



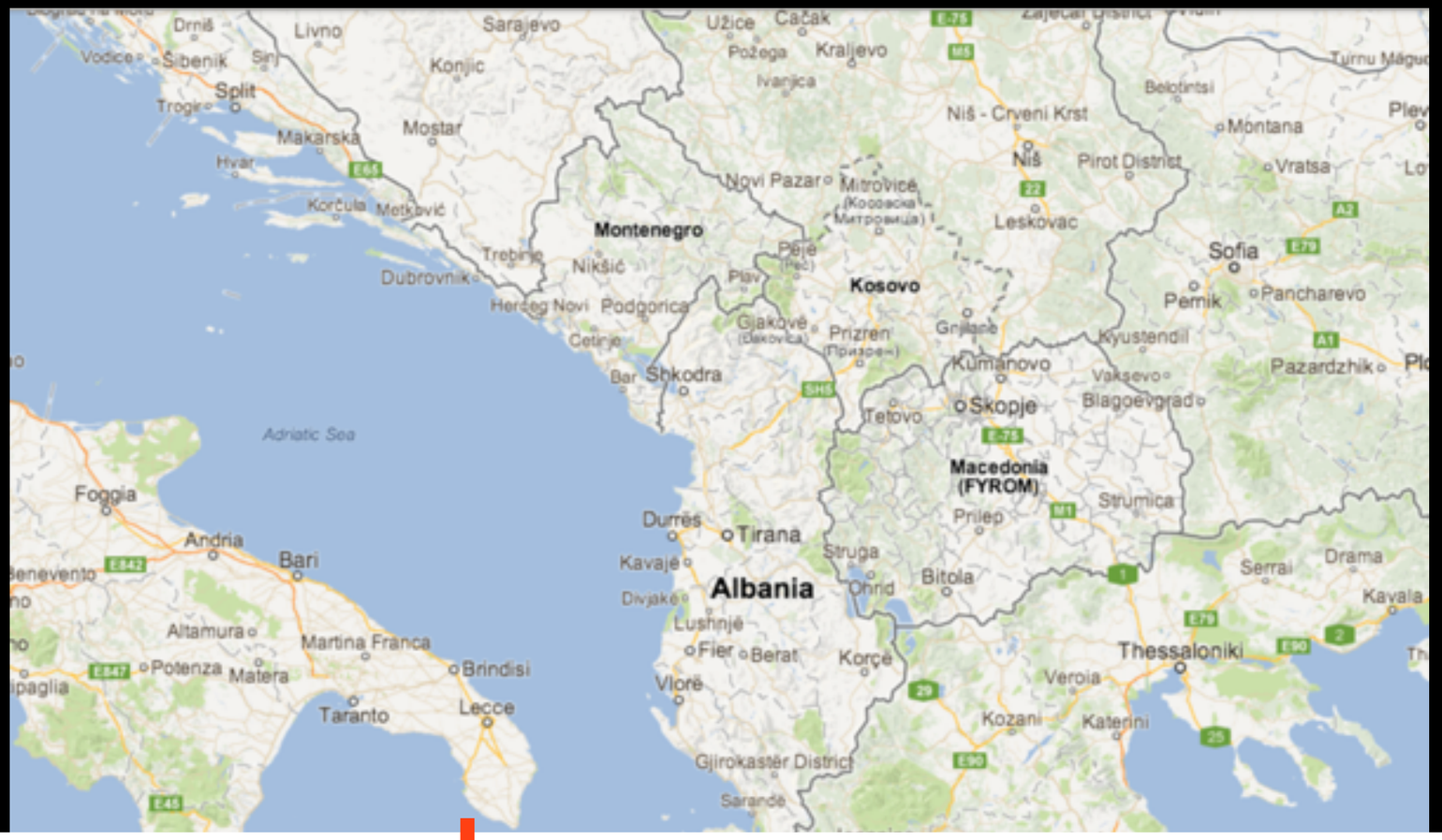
Reporting Server

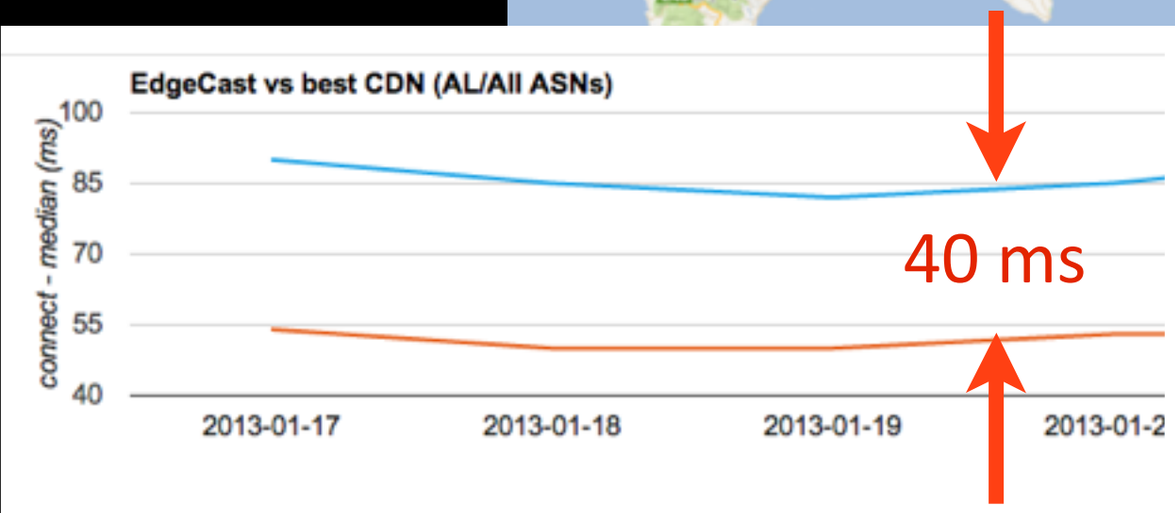
CDN1

CDN2

CDN3

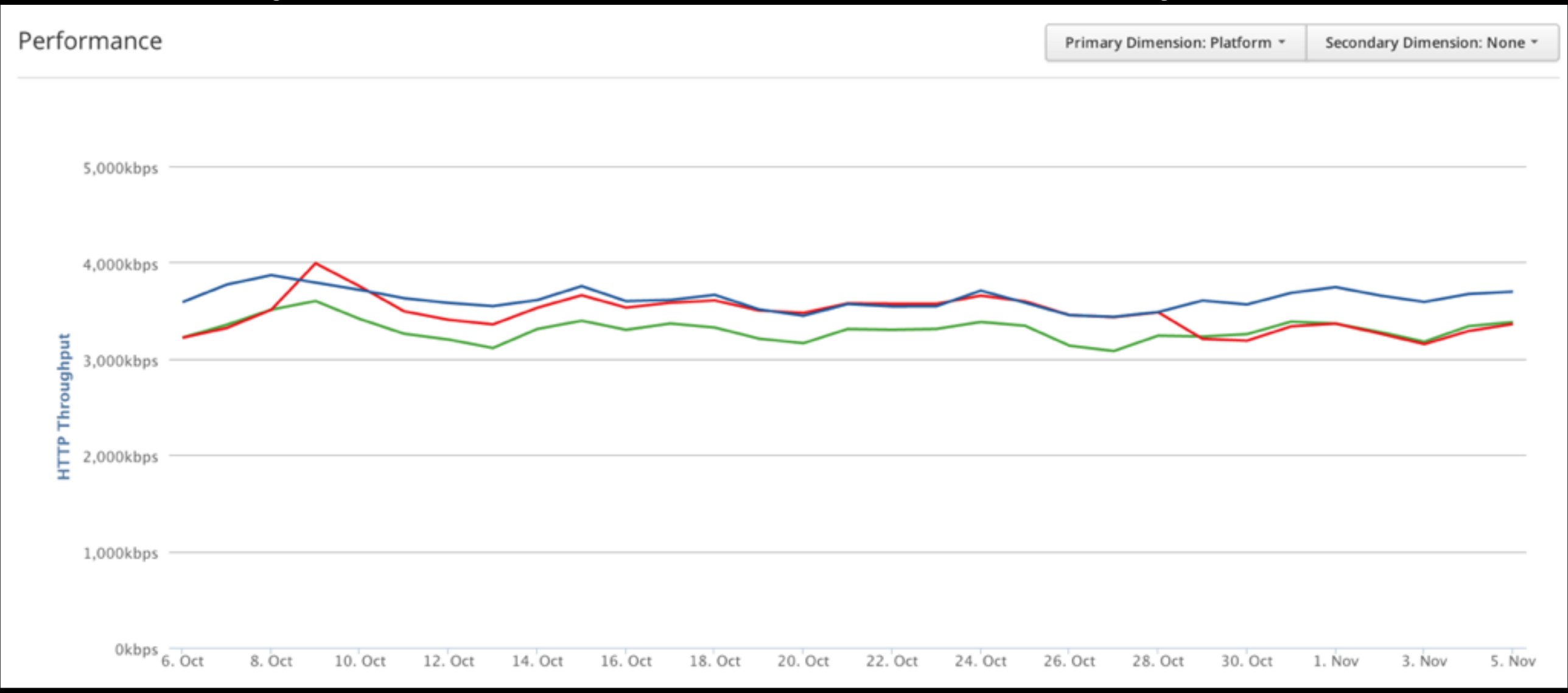
More visibility using RUM





RUM helped us detect a problem causing Albania traffic to go to US

RUM data is too noisy. Look for trends and patterns, not individual data points



Main RUM platforms were not offering IPv6 prior to v6 day

RIPE Atlas


RIPE
NCC


Map
Satellite
Greenland
Iceland
Canada
Russia
Kazakhstan
Mongolia
Afghanistan
Algeria
Libya
Egypt
Saudi Arabia
Niger
Chad
Ethiopia
Tanzania
Angola
Madagascar
Indonesia
Venezuela
Colombia
Brazil
Peru
Bolivia
Argentina
North Pacific Ocean
North Atlantic Ocean
South Pacific Ocean
South Atlantic Ocean
Indian Ocean
Google
Map data ©2013 MapLink
Terms of Use

This talk is not about Atlas but Lets see how Atlas helped us to fine tune our network for IPv6 launch

API

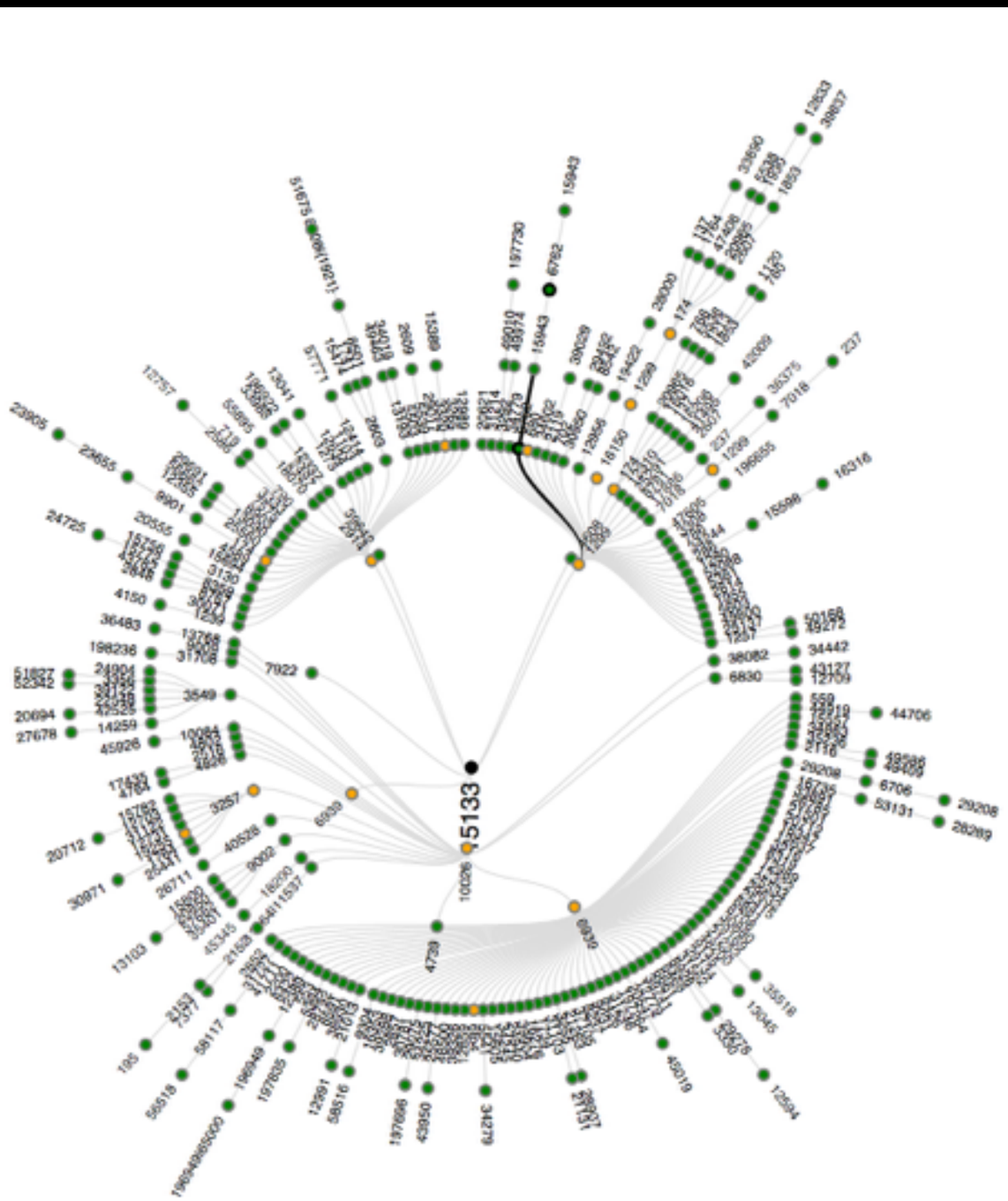

```
[
  - {
      af: 4,
      dst_addr: "192.229.145.163",
      dst_name: "192.229.145.163",
      endtime: 1382111640,
      from: "124.212.215.50",
      fw: 4560,
      msm_id: 1033548,
      msm_name: "Traceroute",
      paris_id: 1,
      prb_id: 2891,
      proto: "ICMP",
    - result: [
        - {
            hop: 1,
          - result: [
            - {
                from: "192.168.11.250",
                rtt: 2.928,
                size: 56,
                ttl: 255
              },
            - {
                from: "192.168.11.250",
                rtt: 2.721,
                size: 56,
                ttl: 255
              },
            - {
                from: "192.168.11.250",
                rtt: 2.862,
                size: 56,
                ttl: 255
              }
            ]
        },
      - {
          hop: 2,
```


Ability to run Multiple Tests

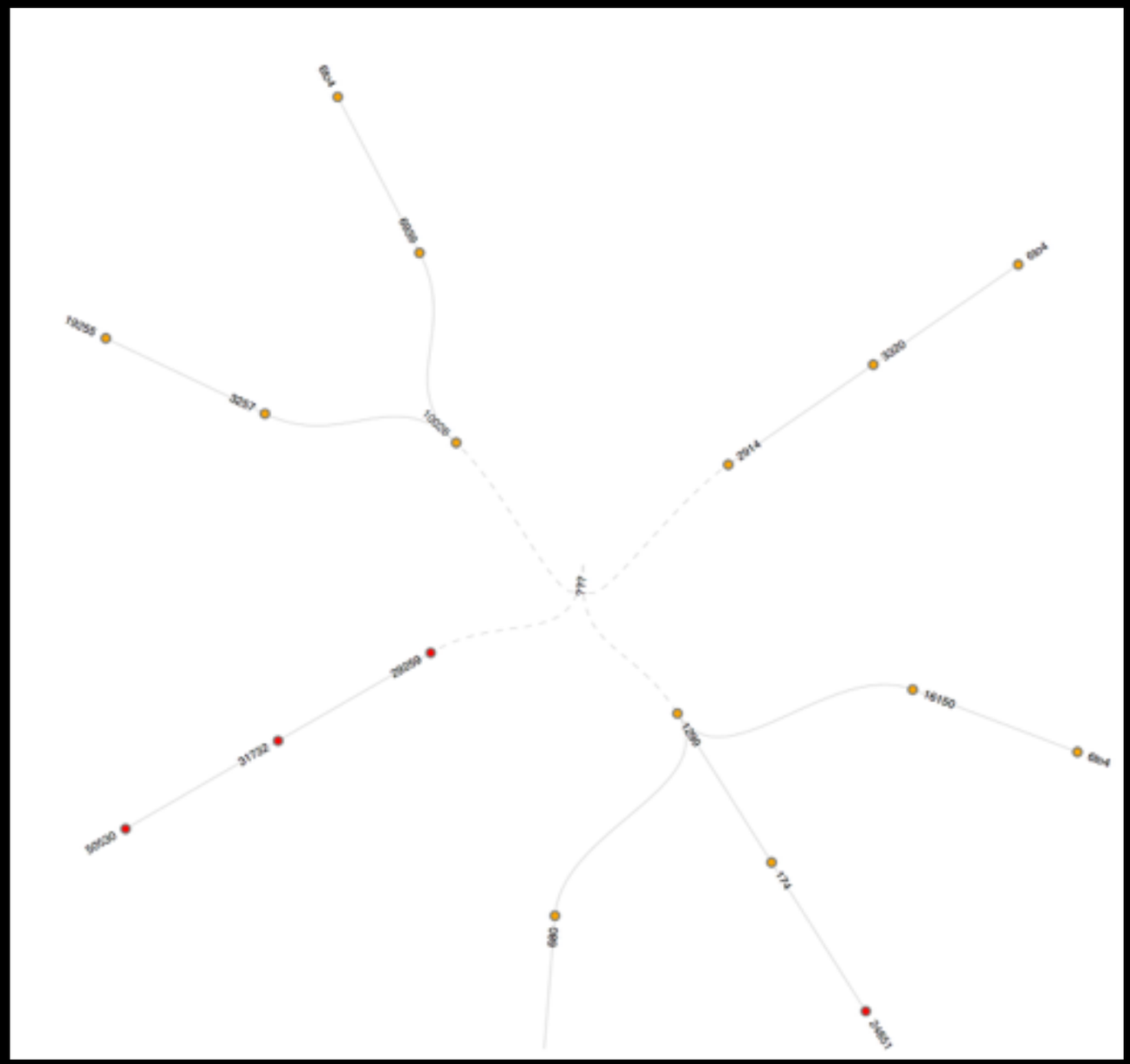
- Ping
- Ping6
- Traceroute
- Traceroute6
- DNS
- DNS6
- SSLCert
- SSLCert6

HTTP tests are in Beta

IPv6 reachability analysis by Atlas

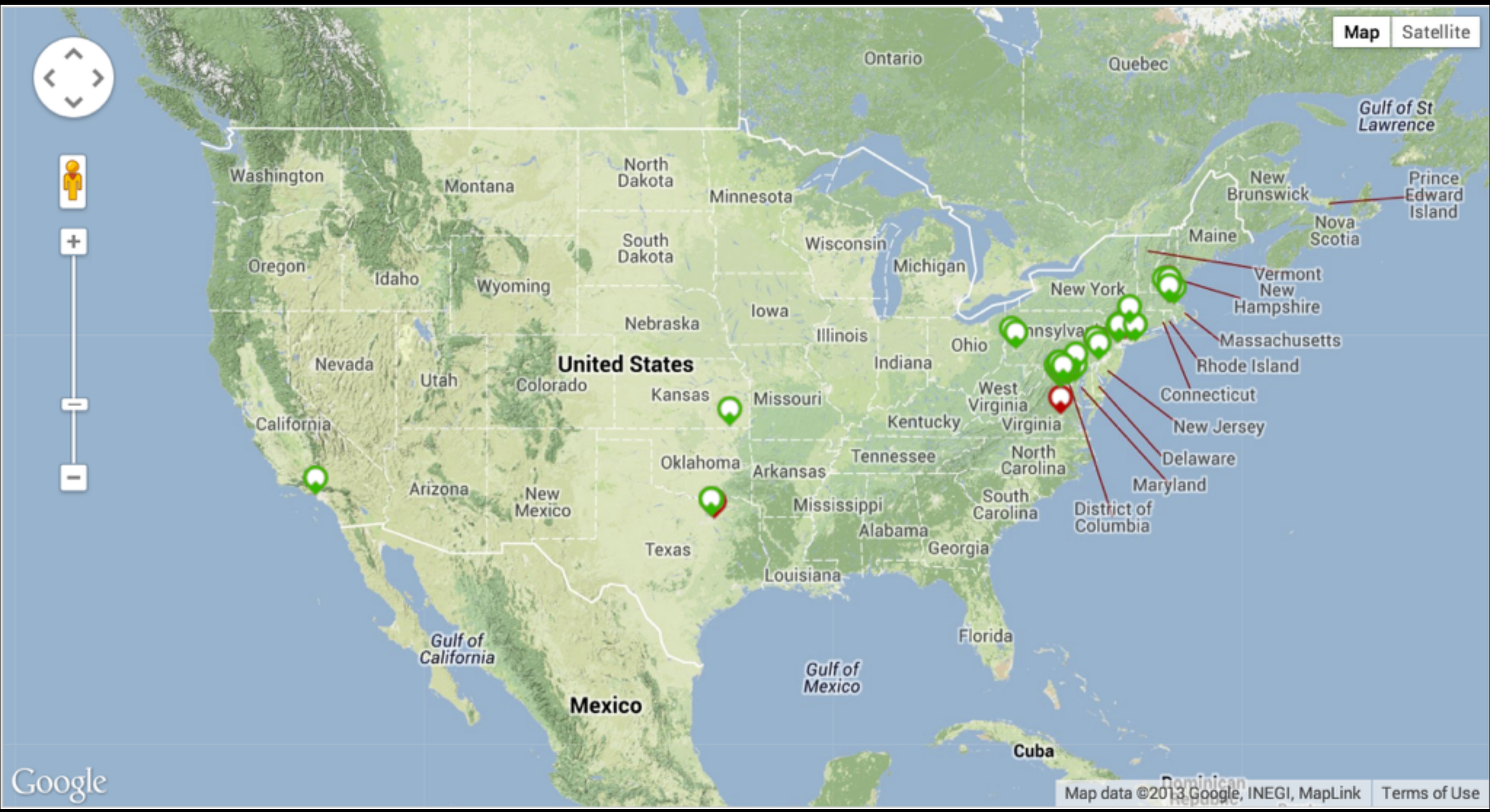


AS-path analysis



AS-path of failed traces

We need more Atlas probes in US !



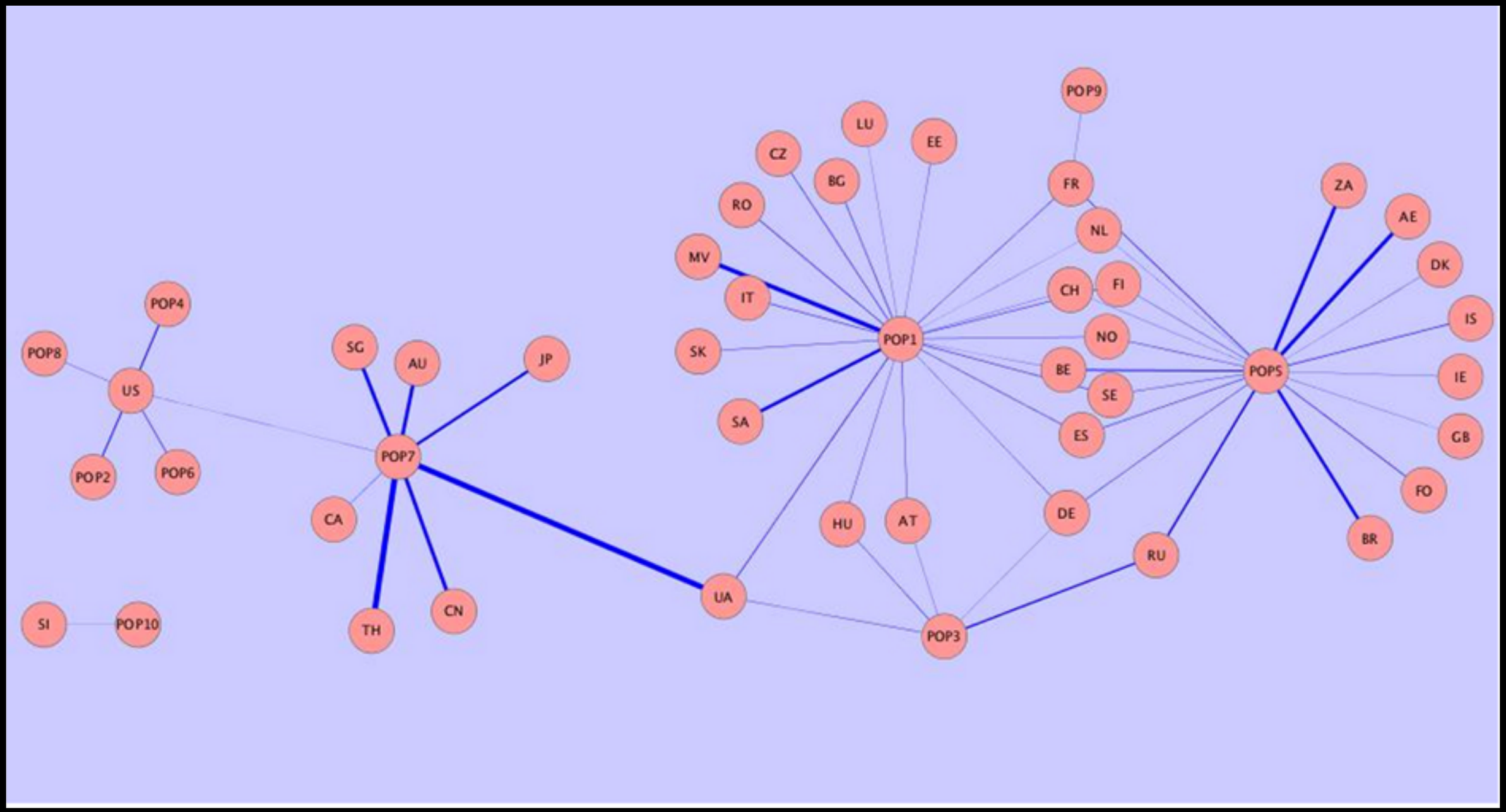
Current active probes in a major US mobile network

If you are thinking about launching a looking glass, consider hosting an Atlas probe instead. It will help the community way better than traditional looking glasses

and yes! one thing that atlas can not provide is BGP data

What did we learn from Atlas about our IPv6 Performance and availability?

Visualization of “which POP is handling traffic coming from a country?” and latency



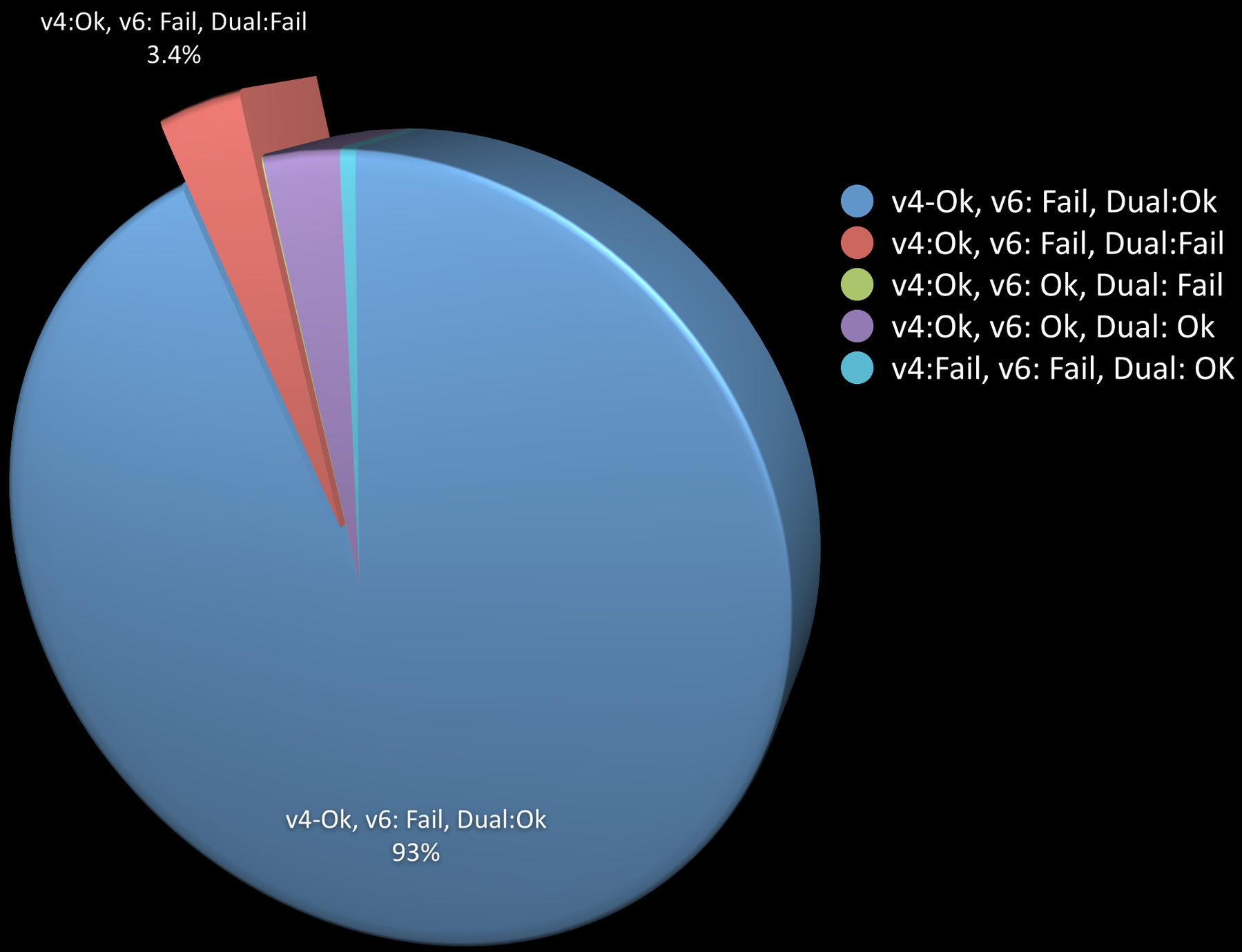
Ukraine goes to POP1 and POP7. POP7 has much higher latency

We launched a beacon dedicated to IPv6 measurements that tested the following:

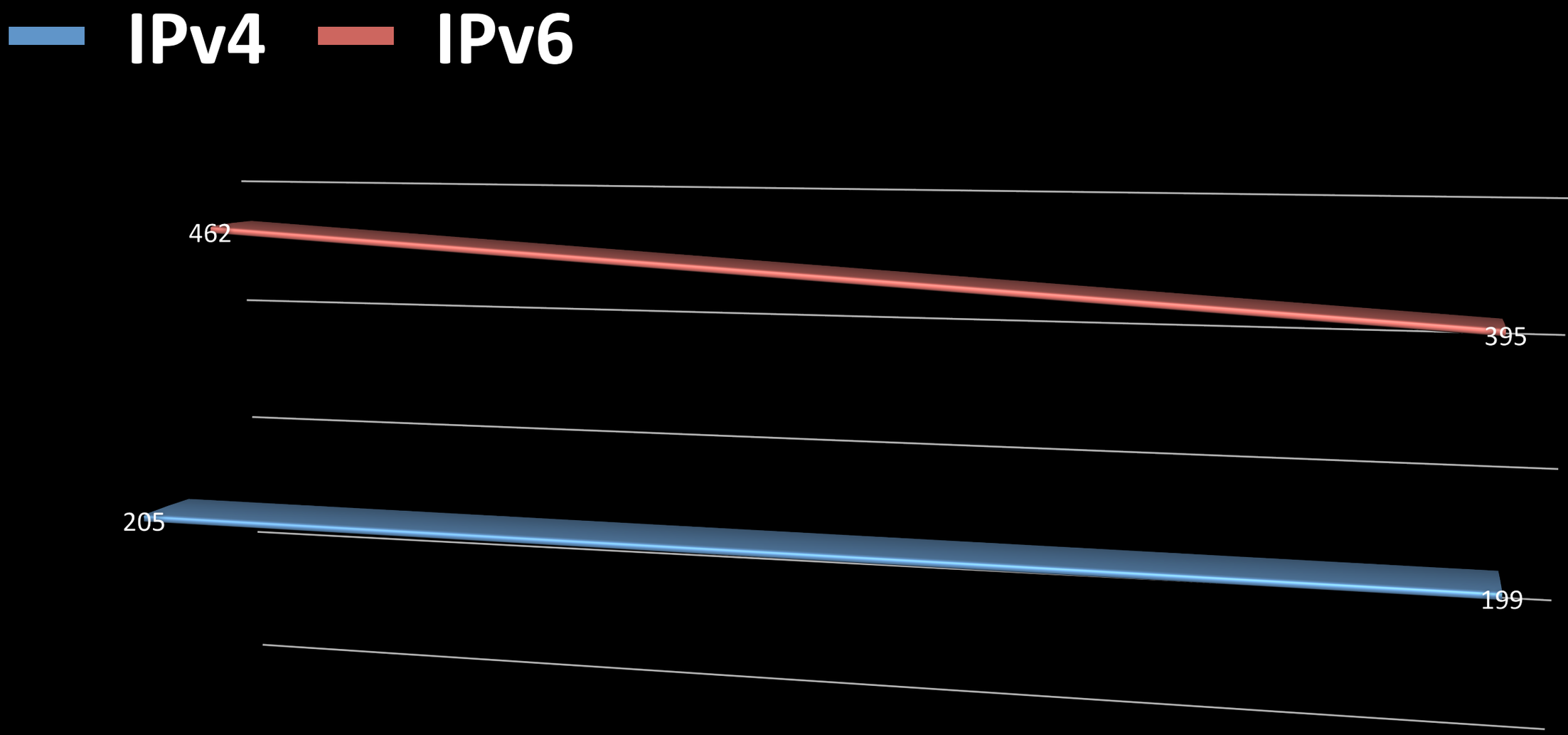
- IPv4 only
- IPv6 only
- Dual v4 and v6

In order to reduce the failures to a more actionable set, the beacon also checked connectivity to ipv6.google.com. We first focused on cases where AS numbers fail to reach us over v6 but can browse ipv6.google.com

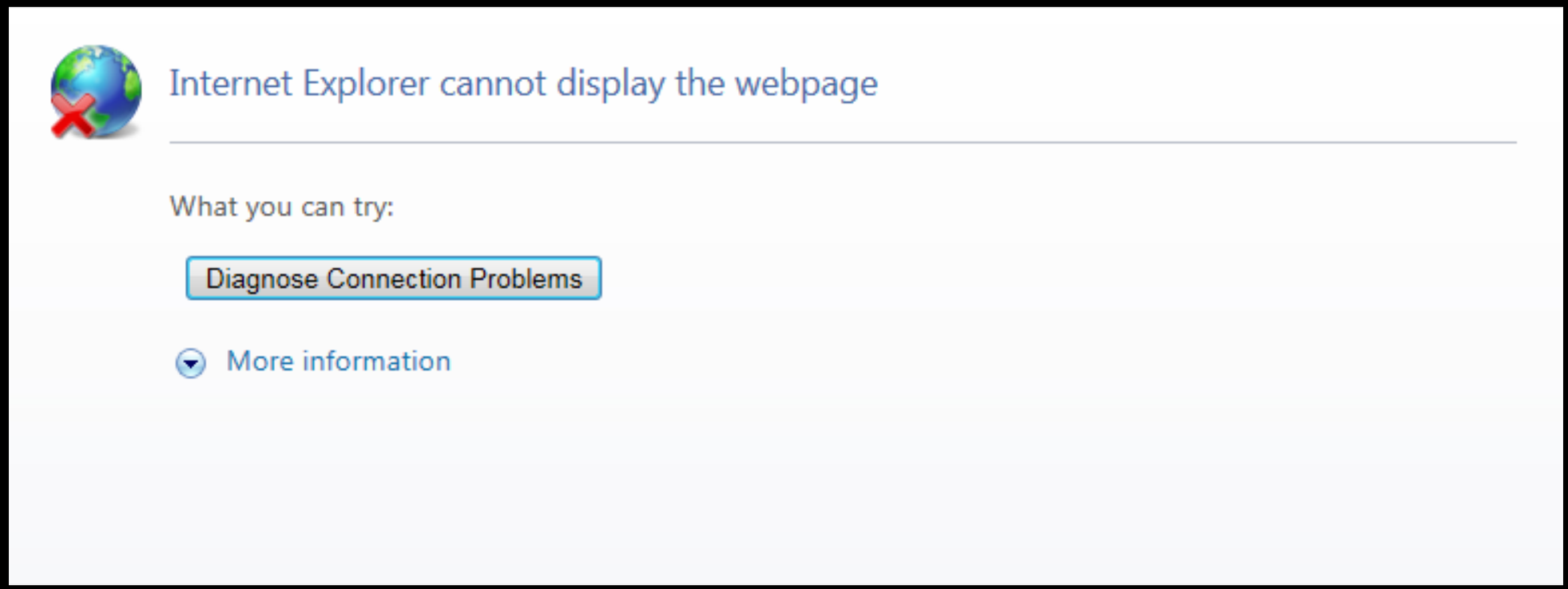
Availability predictions before launch



Latency calculations prior to launch and their progress during our pre-launch performance optimizations



In our http testing we noticed some clients can complete the trace to us, but fail to download http objects!!!



Here is what we saw in packet captures

16:21:13.051353 IP6 2606:2800:234:1df9:13d:1d4e:6b0:10cf.443 > 2001:778:627f:cf1::55.42947: Flags [.], seq 1:1441, ack 101, win 225, length 1440

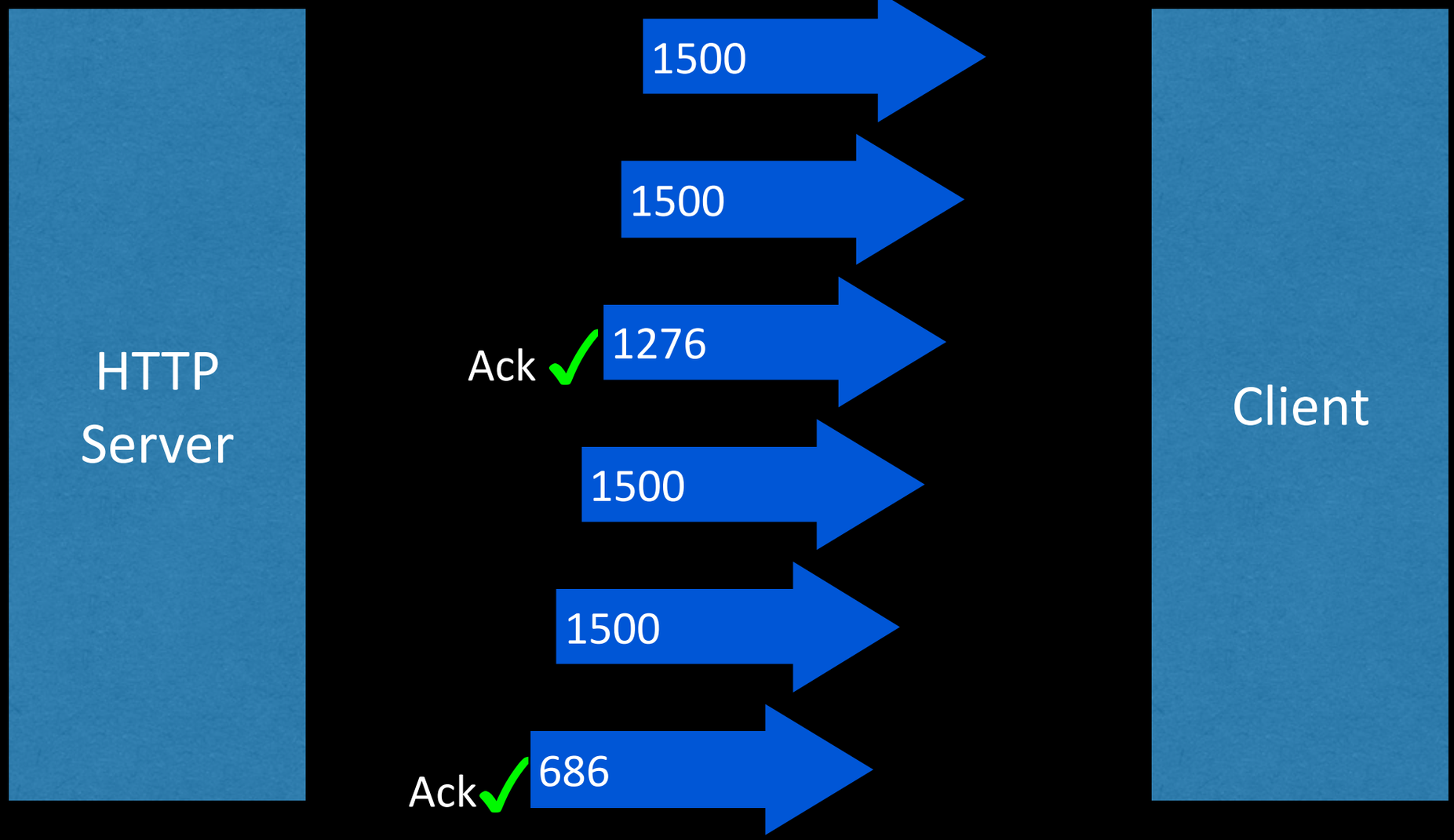
16:21:13.051367 IP6 2606:2800:234:1df9:13d:1d4e:6b0:10cf.443 > 2001:778:627f:cf1::55.42947: Flags [.], seq 1441:2881, ack 101, win 225, length 1440

16:21:13.051372 IP6 2606:2800:234:1df9:13d:1d4e:6b0:10cf.443 > 2001:778:627f:cf1::55.42947: Flags [P.], seq 2881:4097, ack 101, win 225, length 1216

16:21:13.051421 IP6 2606:2800:234:1df9:13d:1d4e:6b0:10cf.443 > 2001:778:627f:cf1::55.42947: Flags [.], seq 4097:5537, ack 101, win 225, length 1440

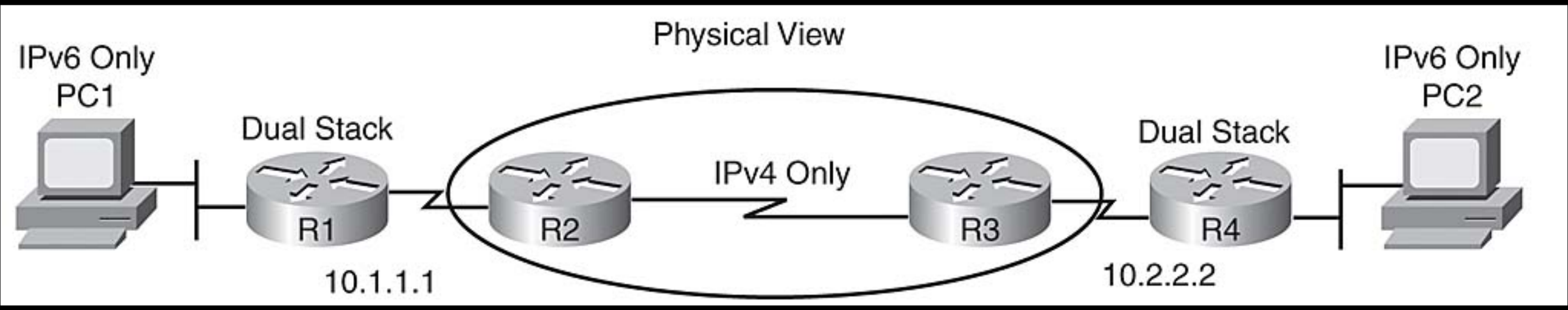
16:21:13.051427 IP6 2606:2800:234:1df9:13d:1d4e:6b0:10cf.443 > 2001:778:627f:cf1::55.42947: Flags [.], seq 5537:6977, ack 101, win 225, length 1440

16:21:13.051431 IP6 2606:2800:234:1df9:13d:1d4e:6b0:10cf.443 > 2001:778:627f:cf1::55.42947: Flags [P.], seq 6977:7603, ack 101, win 225, length 626



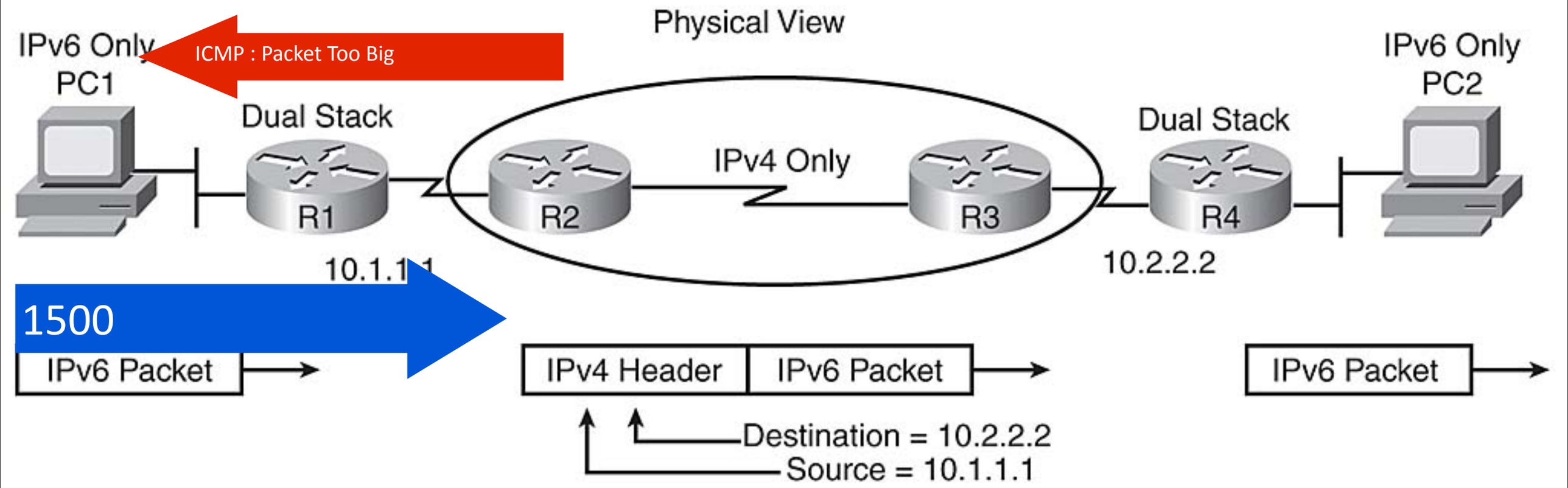
This is a clear sign of a path MTU problem

Lets try to explain what is happening here



http://www.networkworld.com/subnets/cisco/082307-ch7-ccna-icnd2.html

But there is already a mechanism to prevent this from happening



So Why the server did not adjust MSS based on "ICMP packet too big" message?

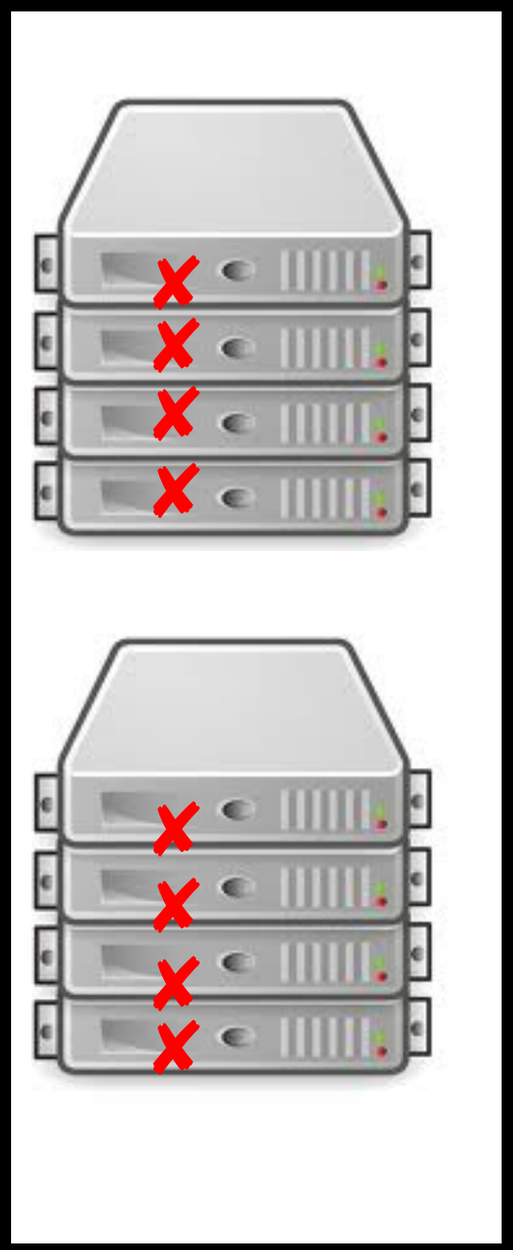
http://www.networkworld.com/subnets/cisco/082307-ch7-ccna-icnd2.html

Did the server ever received the “ICMP packet too big” message?

Server handling the flow in

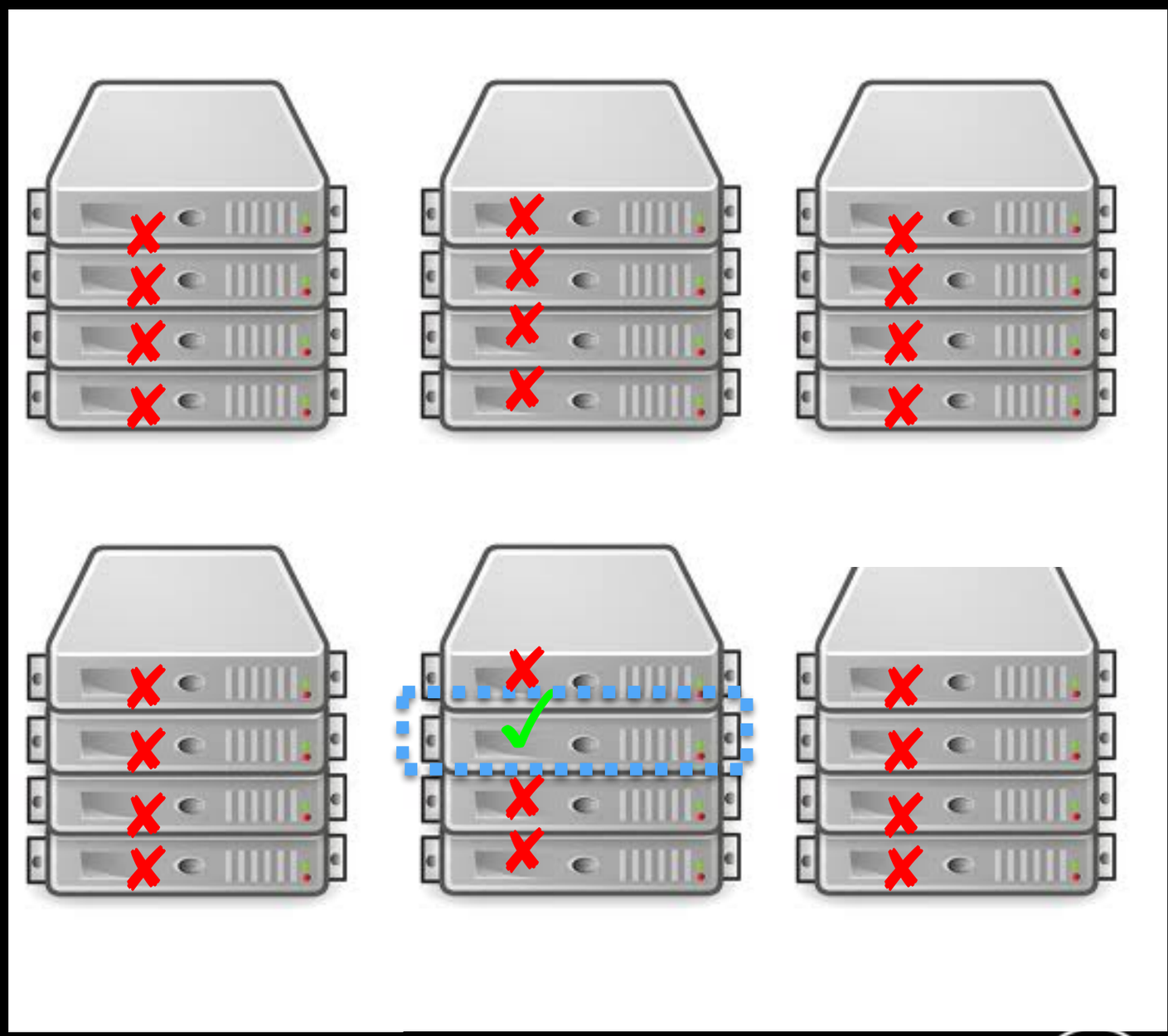
No!

All other servers in



No!

How about all the servers in the world?



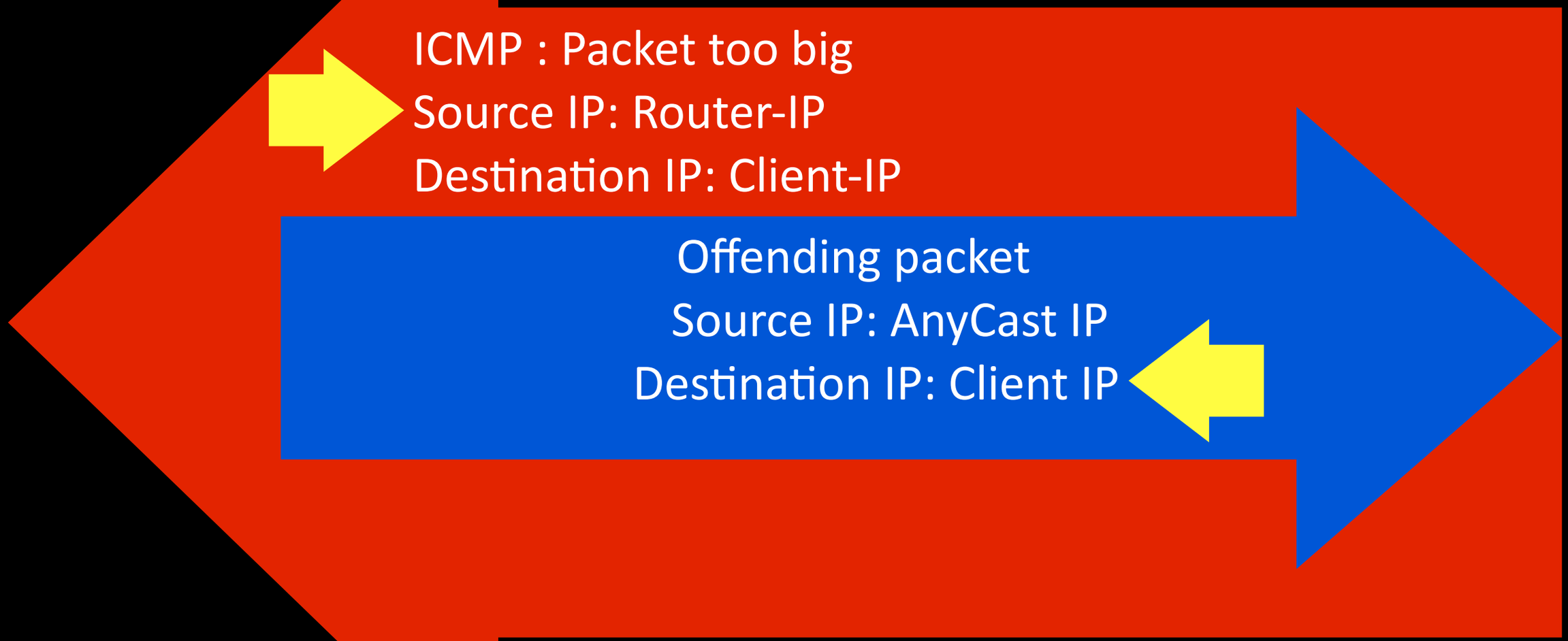
Found it in Paris!

We searched our entire platform for that ICMP message

…and found the ICMP packet in Paris!

Why that packet was received in paris?

The answer is inside the “ICMP packet too big message”



Actual packet

Server IP: 2606:2800:234:124e:17ca:871:eb2:2067

Client IP: 2002:5ee3:1e6b:0:705d:6734:7497:37e8

ICMP Source IP: 2001:470:0:24f::2

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	2001:470:0:24f::2	2606:2800:234:124e:17ca:871:eb2:2067	ICMPv6	1294	Packet Too Big
2	0.009201	2001:470:0:24f::2	2606:2800:234:124e:17ca:871:eb2:2067	ICMPv6	1294	Packet Too Big
3	0.010257	2001:470:0:24f::2	2606:2800:234:124e:17ca:871:eb2:2067	ICMPv6	1294	Packet Too Big
4	3.004244	2001:470:0:24f::2	2606:2800:234:124e:17ca:871:eb2:2067	ICMPv6	1294	Packet Too Big
5	3.009929	2001:470:0:24f::2	2606:2800:234:124e:17ca:871:eb2:2067	ICMPv6	1294	Packet Too Big
6	3.013281	2001:470:0:24f::2	2606:2800:234:124e:17ca:871:eb2:2067	ICMPv6	1294	Packet Too Big
7	9.014806	2001:470:0:24f::2	2606:2800:234:124e:17ca:871:eb2:2067	ICMPv6	1294	Packet Too Big
8	9.027588	2001:470:0:24f::2	2606:2800:234:124e:17ca:871:eb2:2067	ICMPv6	1294	Packet Too Big
9	9.033525	2001:470:0:24f::2	2606:2800:234:124e:17ca:871:eb2:2067	ICMPv6	1294	Packet Too Big

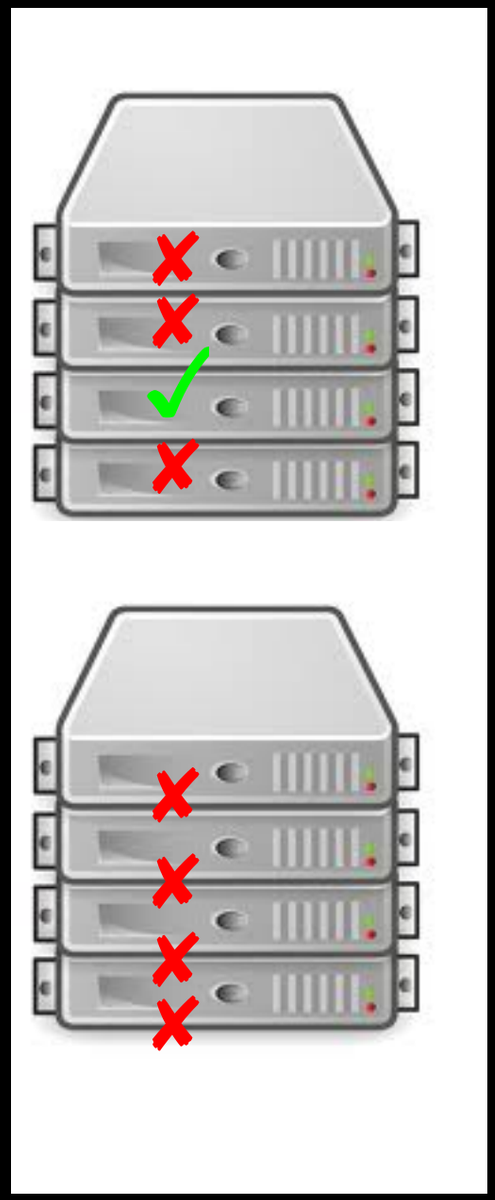
```
▷ Internet Protocol Version 6, Src: 2001:470:0:24f::2 (2001:470:0:24f::2), Dst: 2606:2800:234:124e:17ca:871:eb2:2067 (2606:2800:234:124e:17ca:871:eb2:2067)
▽ Internet Control Message Protocol v6
    Type: Packet Too Big (2)
    Code: 0
    Checksum: 0xd37b [correct]
    MTU: 1480
  ▷ Internet Protocol Version 6, Src: 2606:2800:234:124e:17ca:871:eb2:2067 (2606:2800:234:124e:17ca:871:eb2:2067), Dst: 2002:5ee3:1e6b:0:705d:6734:7497:37e8
  ▷ Transmission Control Protocol, Src Port: http (80), Dst Port: 57125 (57125), Seq: 257948825, Ack: 2246093004
  ▷ Hypertext Transfer Protocol
```

Since we had different peering arrangements with the AS number which was sending ICMP packets to us, the packets that were originated from that AS were going to a different pop

so we fixed the peering with offending router. What happened next?

Server Handling the flow in Frankfurt

No!

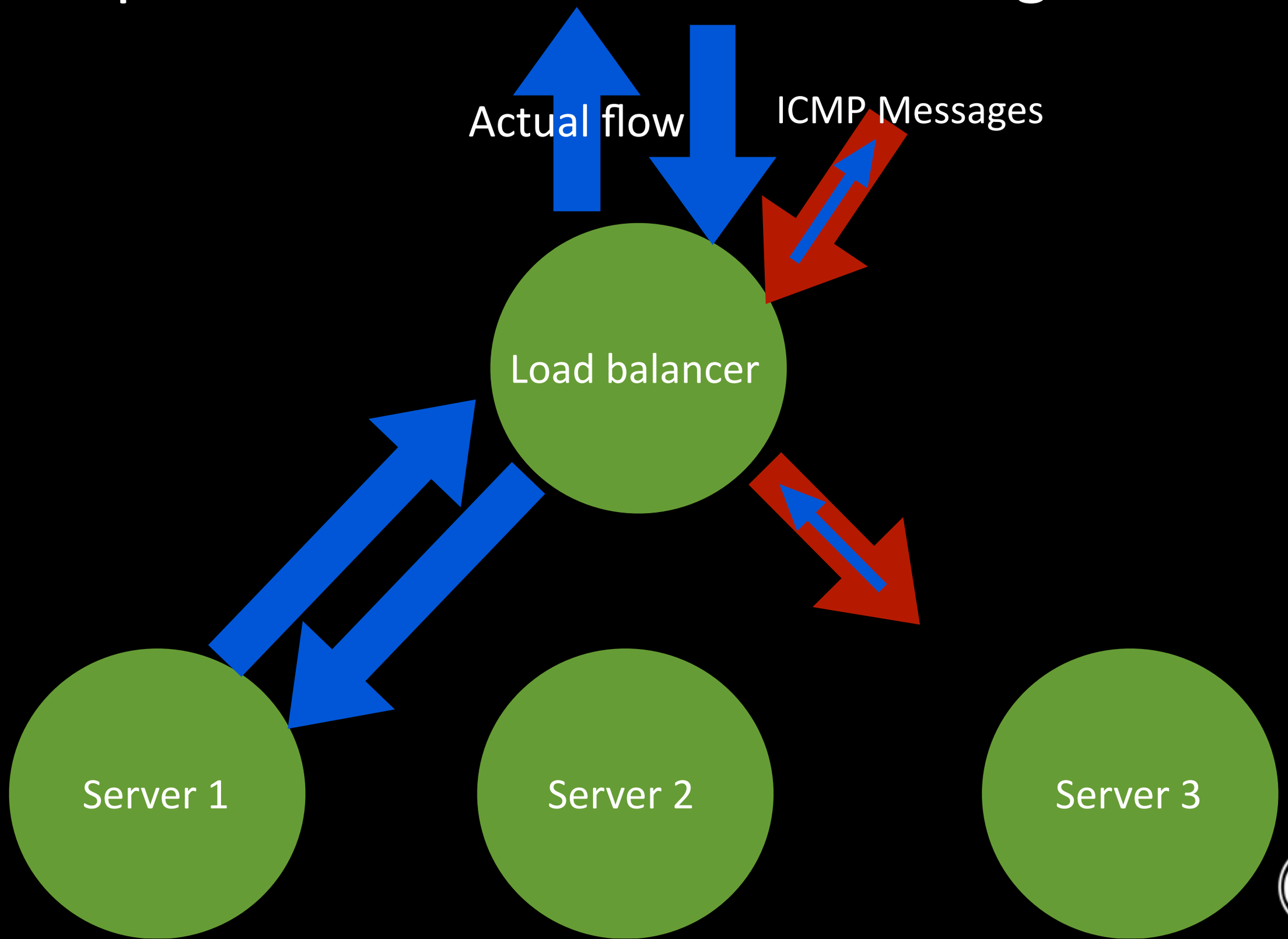


All other servers in Frankfurt

Now we receive the packet in the right pop, but by the wrong server

We have multiple load balancing layers. One of them is based on Equal Cost Multi path (ECMP)

ECMP Load balancers do not check offending ICMP packets to make the forwarding decision



Solution?

The simples solution is in the RFC 2460:

5. Packet Size Issues

IPv6 requires that every link in the internet have an MTU of **1280** octets or greater. On any link that cannot convey a 1280-octet packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6.

The Bigger Problem:

This is happening in IPv4 as well, and if you have an Anycast network, your availability could be impacted by this problem as well!

EdgeCast took a different approach to make sure IPv4 ICMP messages are received by the right server. But since it is only specific to our architecture, we will skip talking about it.

Suggestions:

- To measure the impact of this problem, we recommend monitoring orphaned ICMP messages in Anycast networks.
- You can also setup last mile tests and compare availability of Anycast and Unicast services.

Thank you

Hossein Lotfi

hlotfi@edgecast.com

EdgeCast Networks Inc.