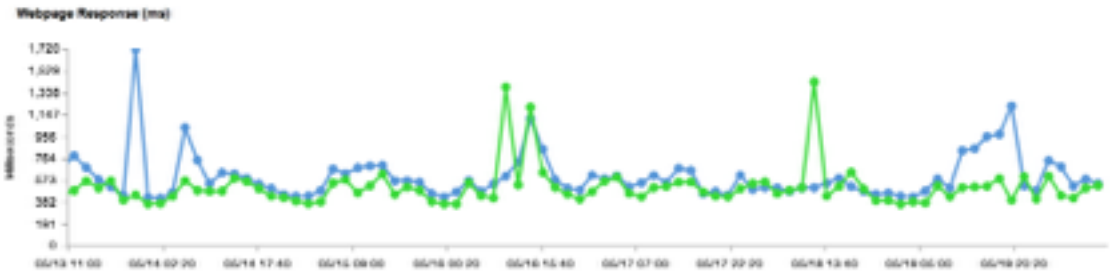
IPv6 and PathMTU Problems in Anycast networks

Hossein Lotfi

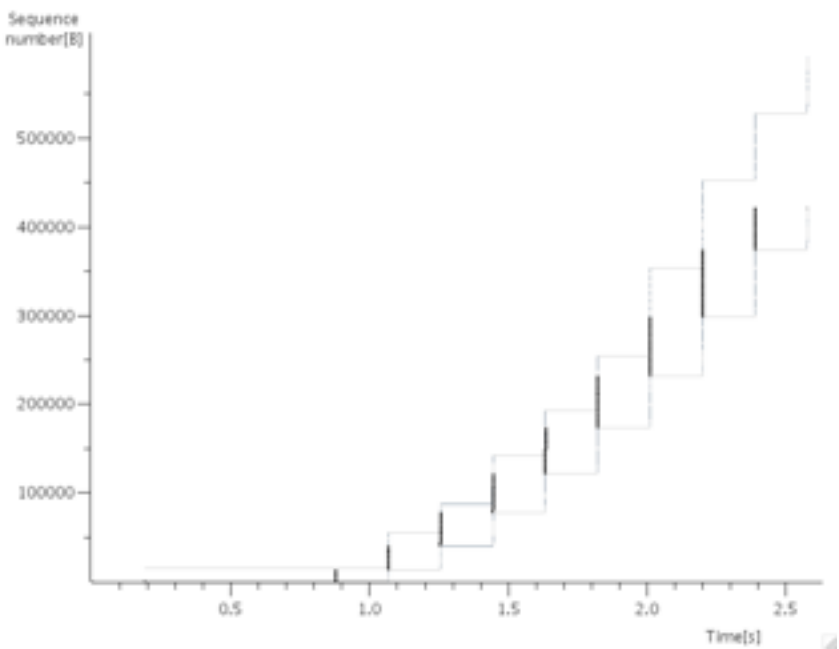
About Verizon EdgeCast

- 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

About Performance Team at EdgeCast



We Analyze mountains of Performance Data and work on ideas to make the CDN faster

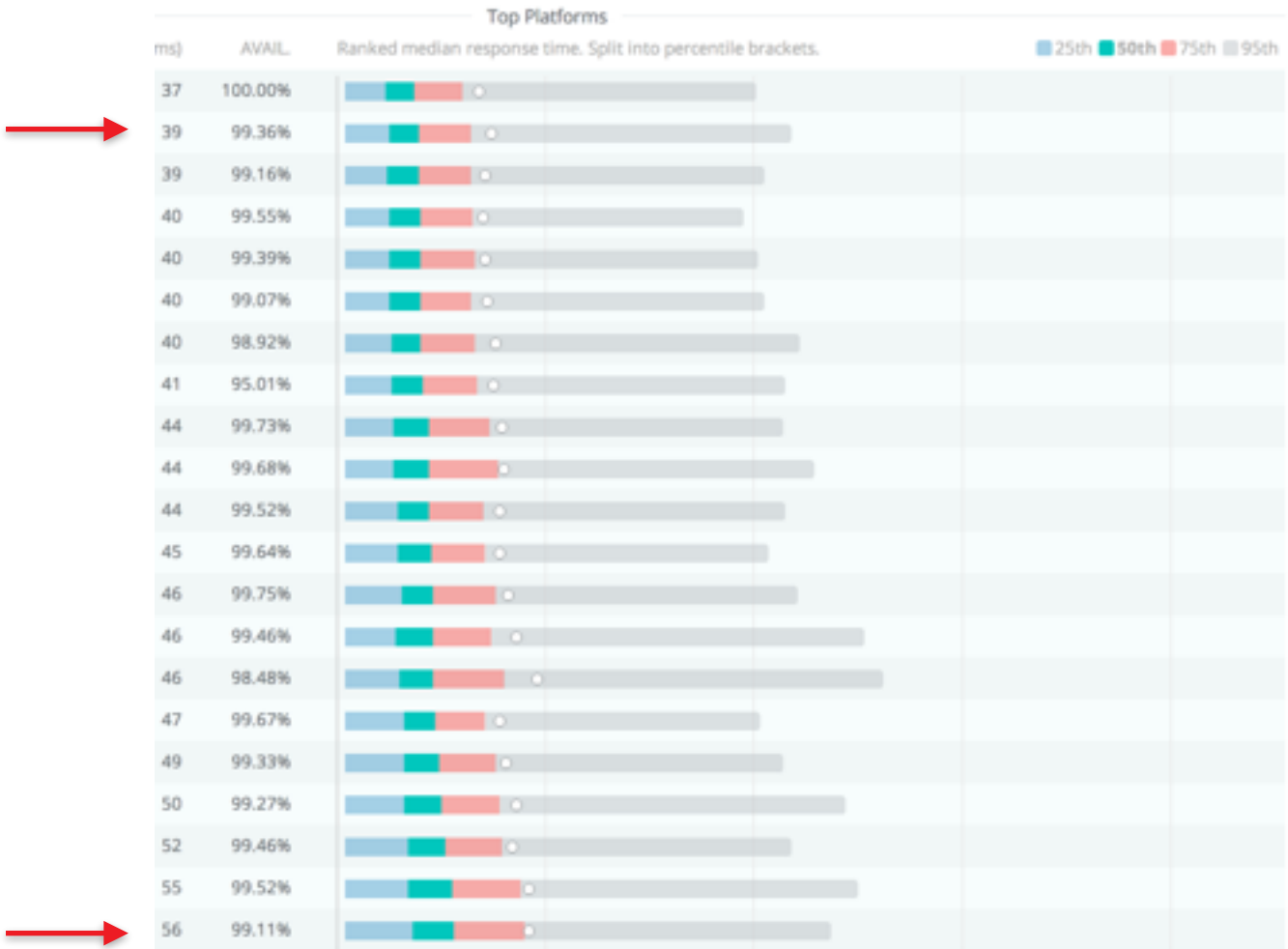


Lots of TCP Optimization!

Perf. Team is formed by full-stack engineers, our projects require us to study every component of a large scale computer network. From physical cables connecting servers to efficiency of our any cast routing and peering session, all the way up to Kernel and Application Layers.

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

CDN is a Very Competitive Industry



The Difference between the best and worse CDN in US is only 17 ms !
(median response time of a 40B object measured 2.8B times / month per CDN)

Today we will talk about:

- How did we test our network prior to IPv6 Launch ?
- PathMTU Problems
- Possible solutions

How did EdgeCast test its network for IPv6 Launch

6/6/12

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

Testing Method 1:

Synthetic Monitoring

Synthetic Monitoring

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

Synthetic nodes that were available to us

Testing Method 1:

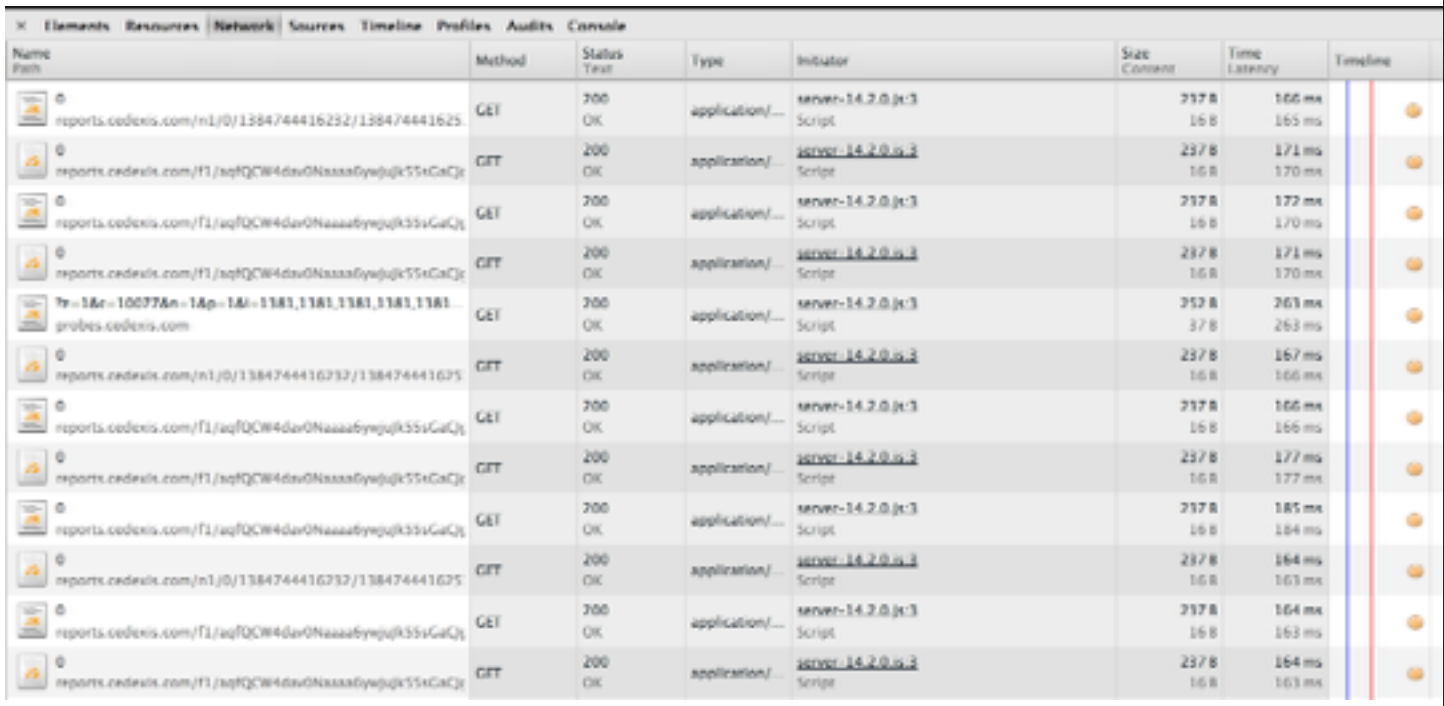
Unfortunately none of them were v6 ready prior to IPv6 Launch

~~Synthetic Monitoring~~

Testing Method 2:

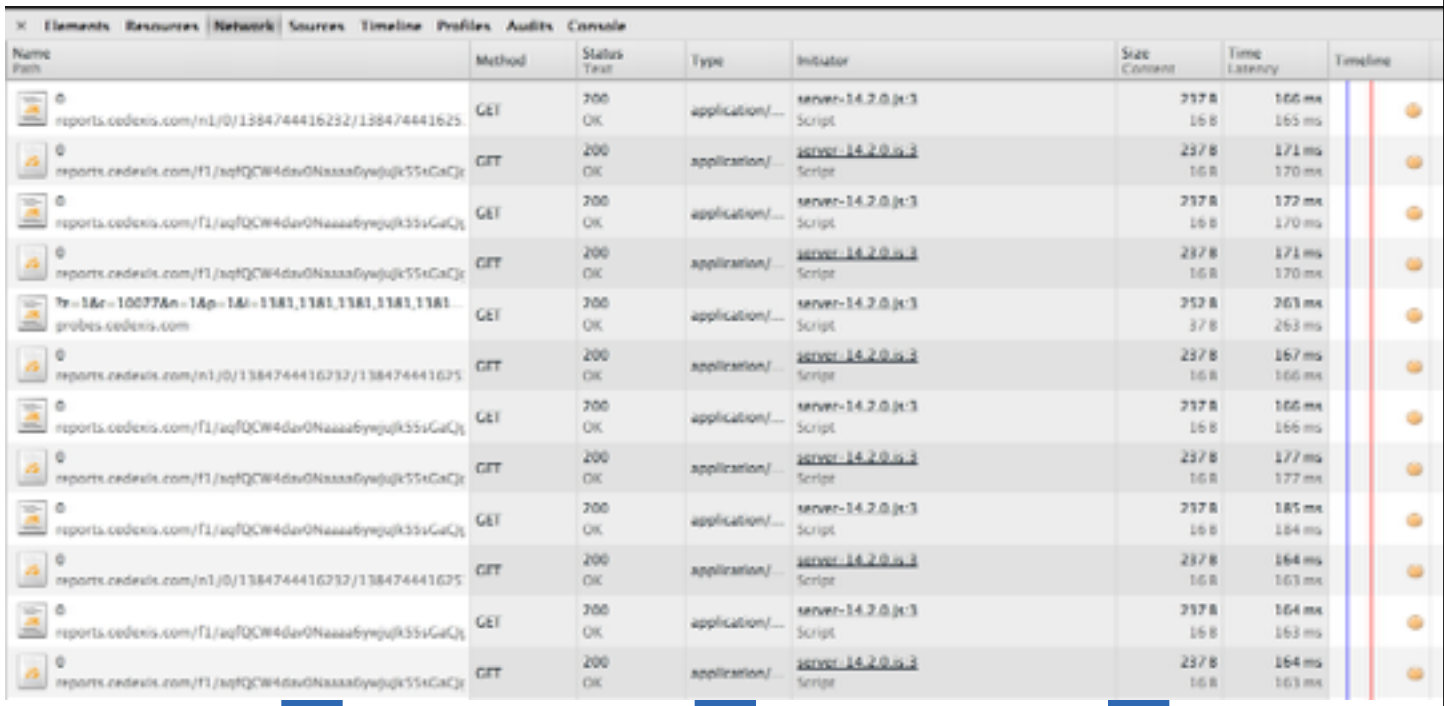
Real User Monitoring (RUM)

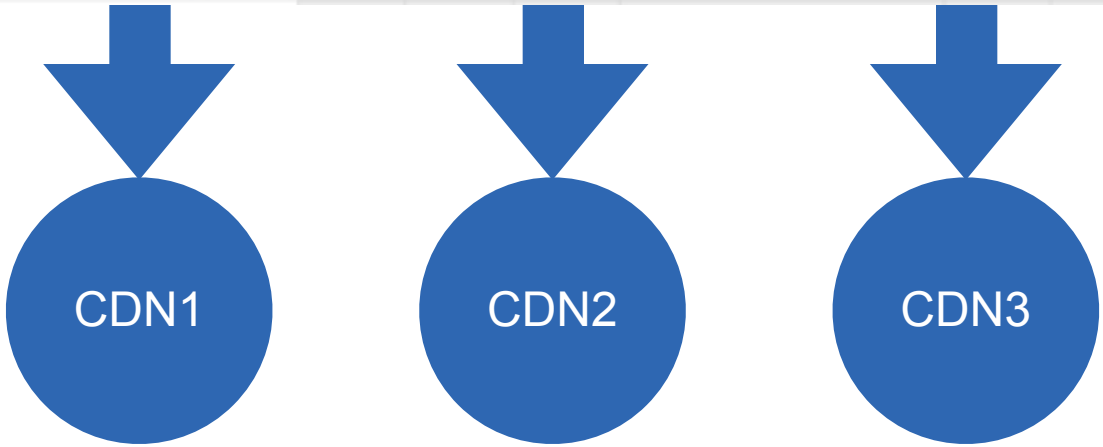
How does RUM work



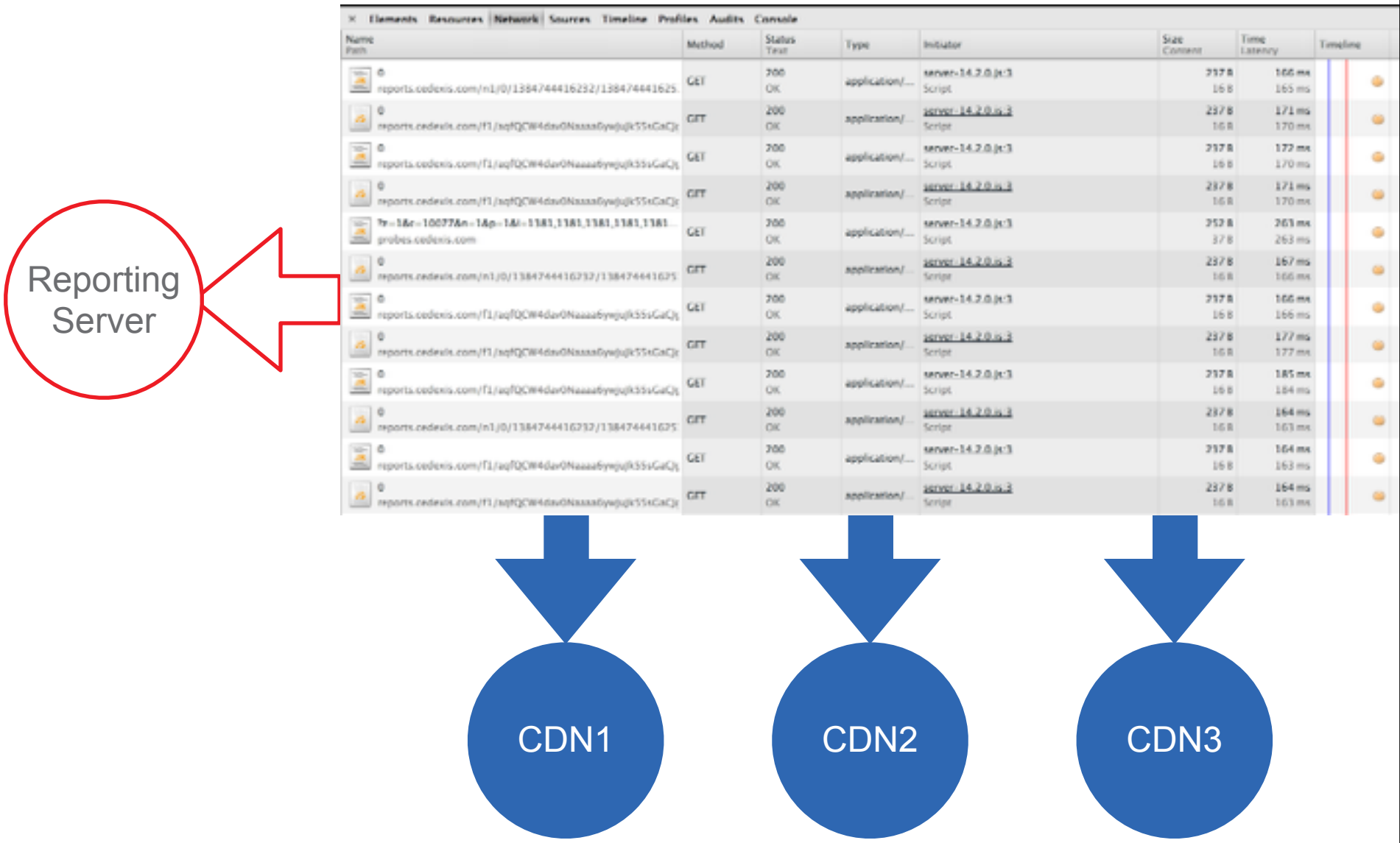
Name / Path	Method	Status / Text	Type	Initiator	Size / Content	Time / Latency	Timeline
0 / reports.cedexis.com/n1/0/1384744416232/1384744416251	GET	200 OK	application/...	server-14.2.0.js:3 Script	217 B 16 B	166 ms 165 ms	
0 / reports.cedexis.com/f1/aqfQCW4dav0Naaaa6ywqujk55xGaCJ	GET	200 OK	application/...	server-14.2.0.js:3 Script	237 B 16 B	171 ms 170 ms	
0 / reports.cedexis.com/f1/aqfQCW4dav0Naaaa6ywqujk55xGaCJ	GET	200 OK	application/...	server-14.2.0.js:3 Script	217 B 16 B	172 ms 170 ms	
0 / reports.cedexis.com/f1/aqfQCW4dav0Naaaa6ywqujk55xGaCJ	GET	200 OK	application/...	server-14.2.0.js:3 Script	237 B 16 B	171 ms 170 ms	
?r=1&r=10077&n=1&p=1&i=1181,1181,1181,1181,1181... / probes.cedexis.com	GET	200 OK	application/...	server-14.2.0.js:3 Script	252 B 37 B	263 ms 263 ms	
0 / reports.cedexis.com/n1/0/1384744416232/1384744416251	GET	200 OK	application/...	server-14.2.0.js:3 Script	237 B 16 B	167 ms 166 ms	
0 / reports.cedexis.com/f1/aqfQCW4dav0Naaaa6ywqujk55xGaCJ	GET	200 OK	application/...	server-14.2.0.js:3 Script	217 B 16 B	166 ms 166 ms	
0 / reports.cedexis.com/f1/aqfQCW4dav0Naaaa6ywqujk55xGaCJ	GET	200 OK	application/...	server-14.2.0.js:3 Script	237 B 16 B	177 ms 177 ms	
0 / reports.cedexis.com/f1/aqfQCW4dav0Naaaa6ywqujk55xGaCJ	GET	200 OK	application/...	server-14.2.0.js:3 Script	217 B 16 B	185 ms 184 ms	
0 / reports.cedexis.com/n1/0/1384744416232/1384744416251	GET	200 OK	application/...	server-14.2.0.js:3 Script	237 B 16 B	164 ms 163 ms	
0 / reports.cedexis.com/f1/aqfQCW4dav0Naaaa6ywqujk55xGaCJ	GET	200 OK	application/...	server-14.2.0.js:3 Script	217 B 16 B	164 ms 163 ms	
0 / reports.cedexis.com/f1/aqfQCW4dav0Naaaa6ywqujk55xGaCJ	GET	200 OK	application/...	server-14.2.0.js:3 Script	237 B 16 B	164 ms 163 ms	

How does RUM work

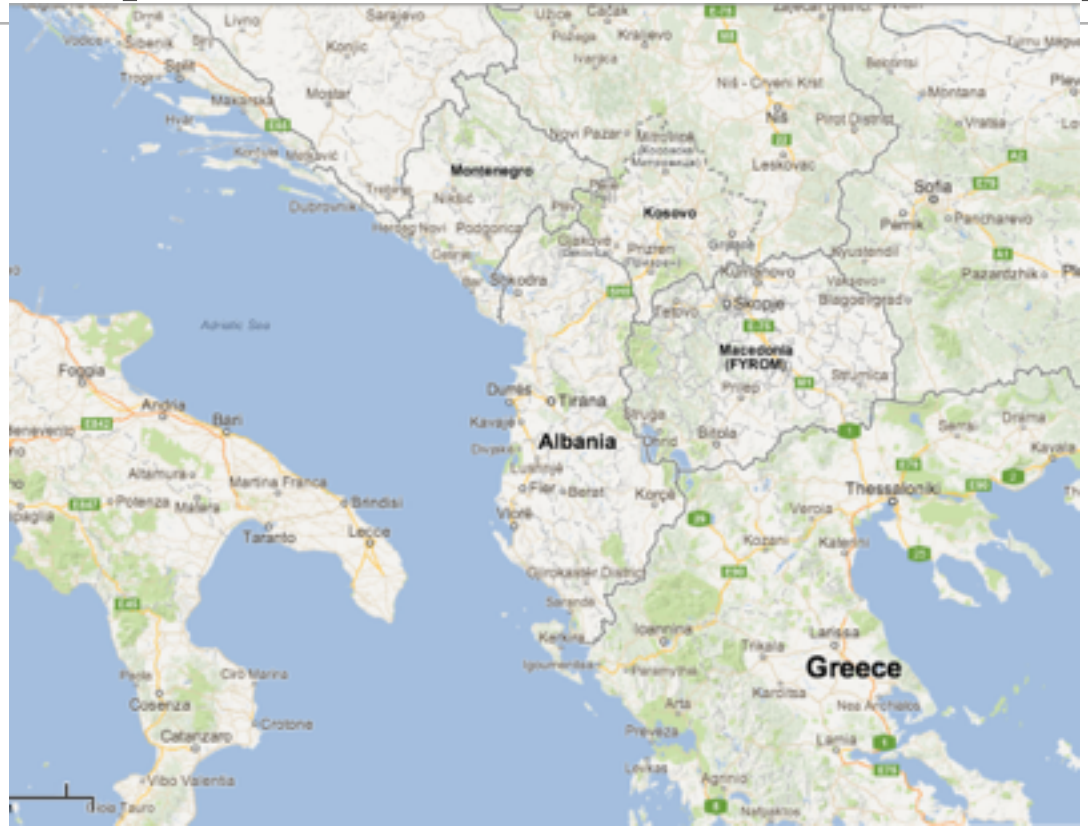




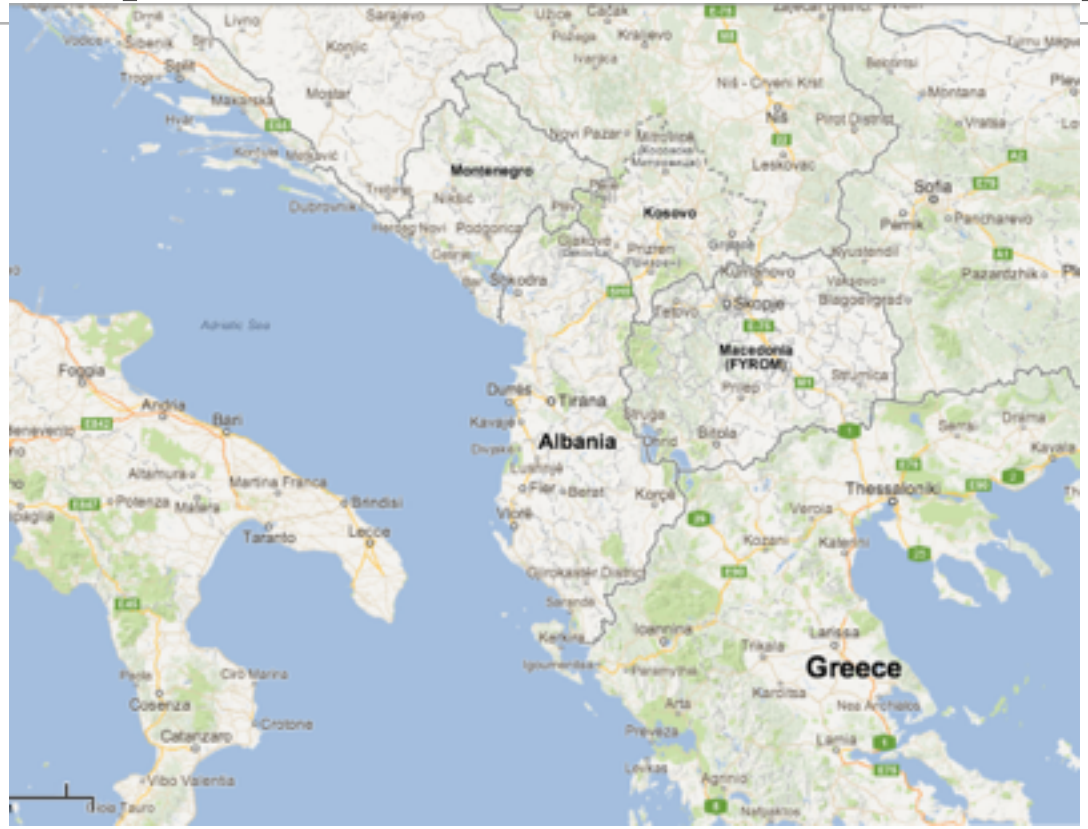
How does RUM work

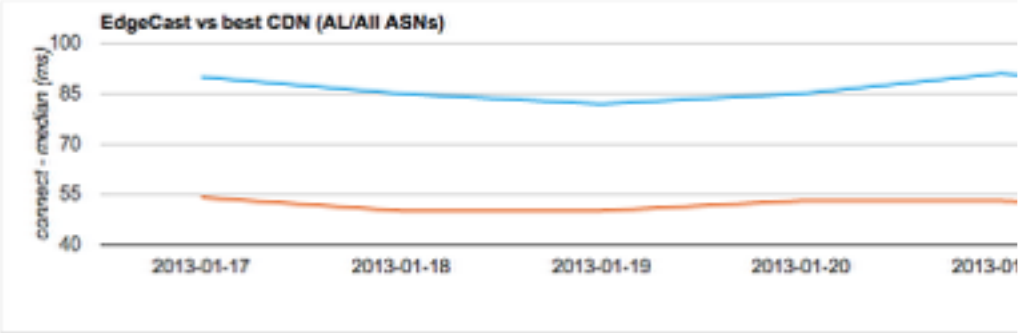


Example of how RUM helps us

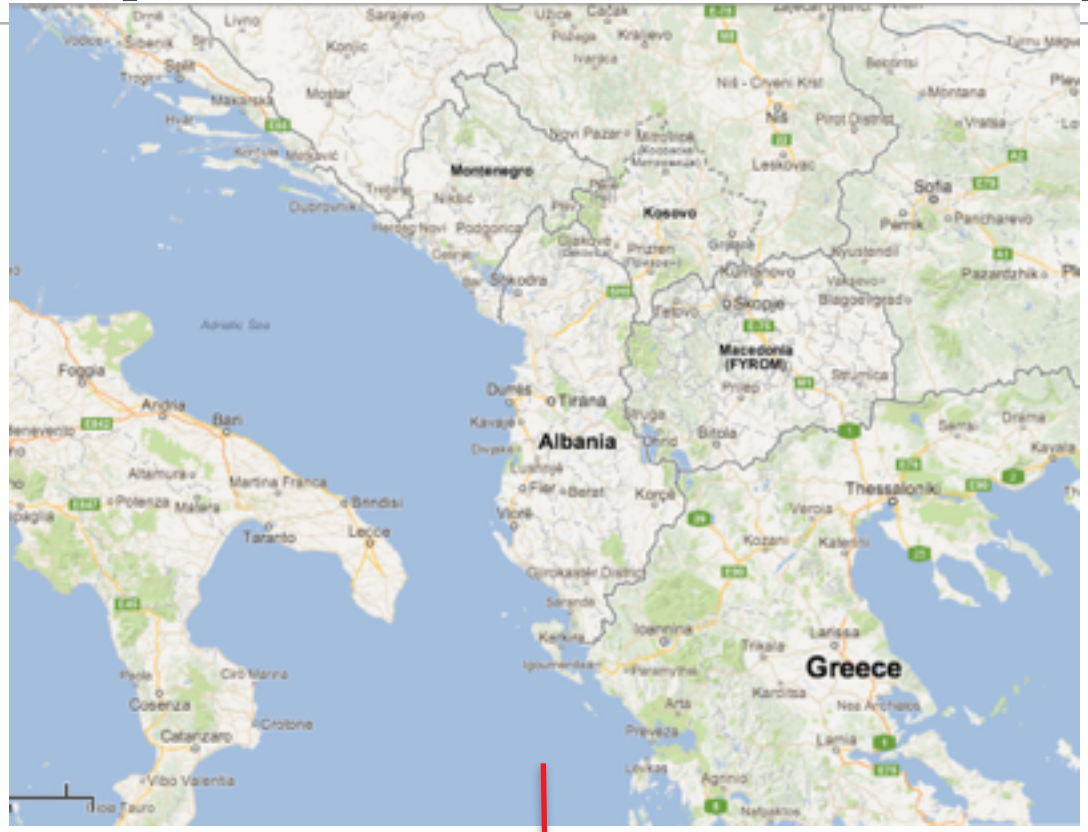


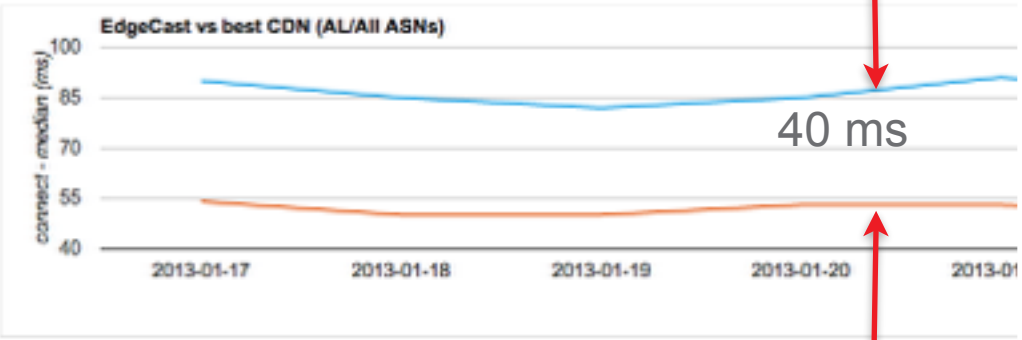
Example of how RUM helps us



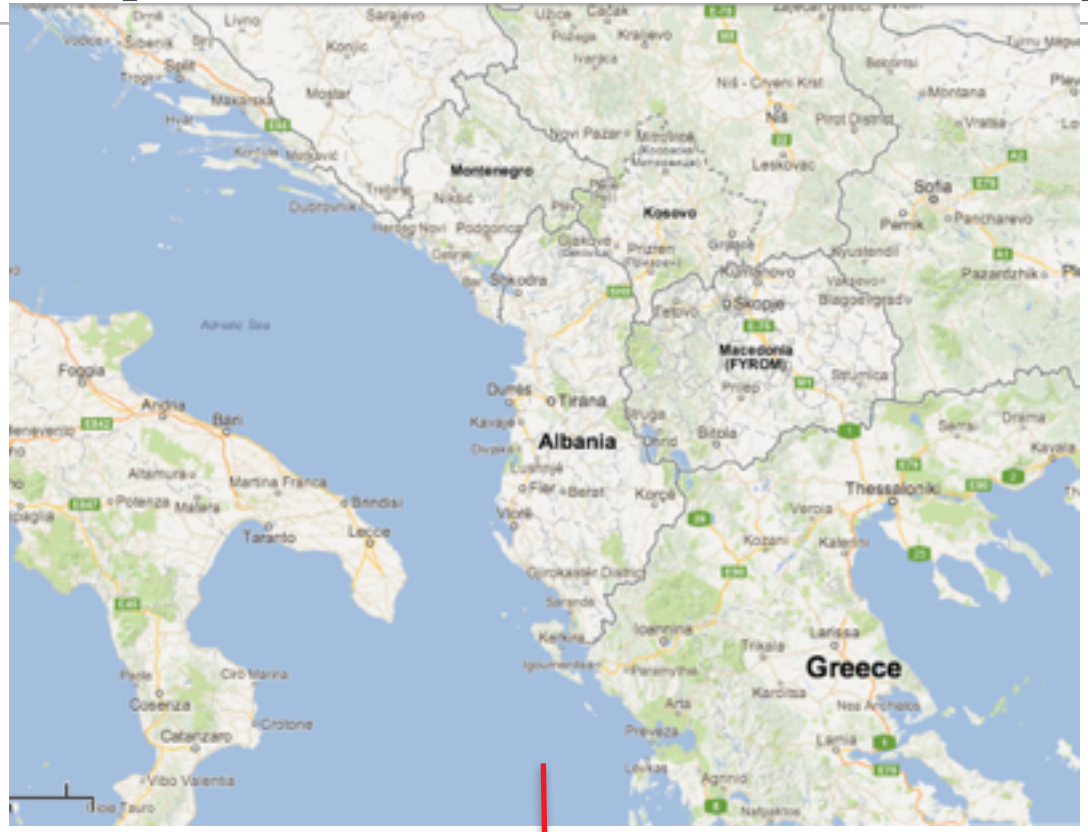


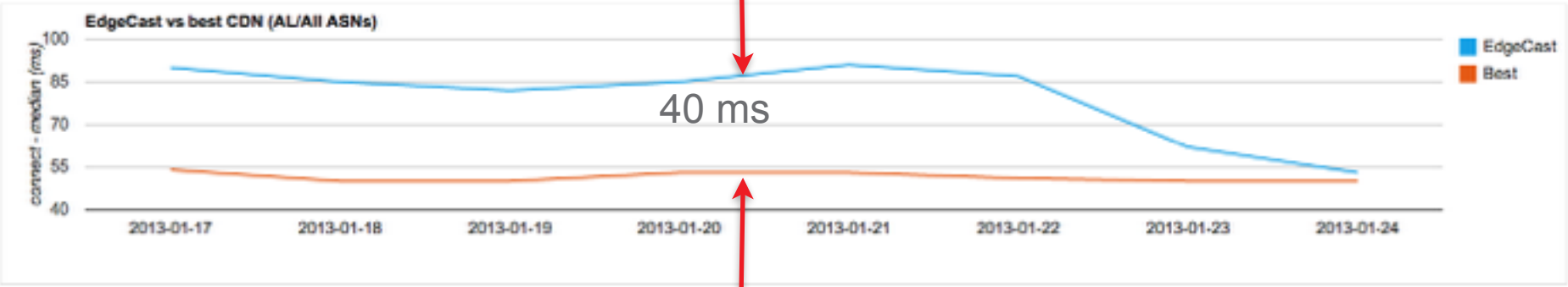
Example of how RUM helps us



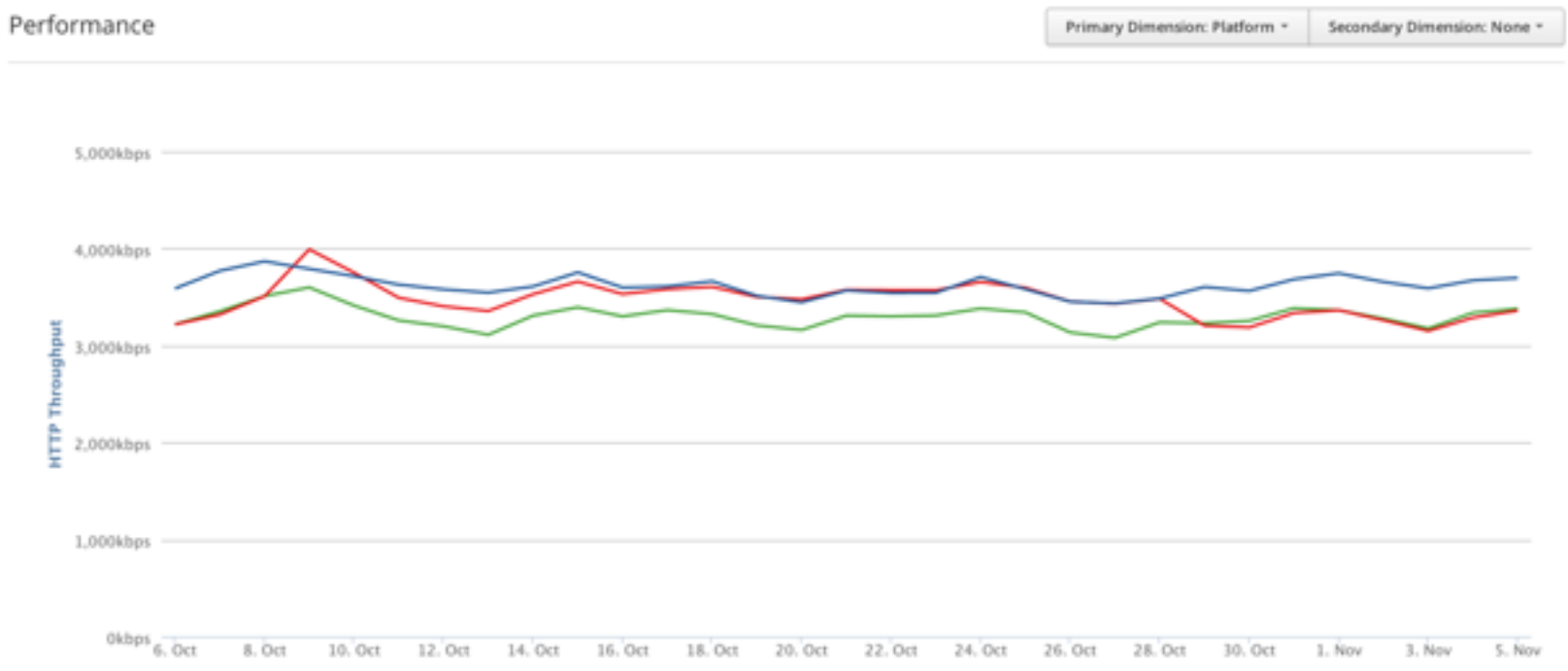


Example of how RUM helps us





RUM = Noise



Look for trends and patterns, not individual data points

Testing Method 2:

Main RUM platforms were not offering IPv6 prior to v6 day

~~Real User Monitoring (RUM)~~

Test 3: RIPE Atlas

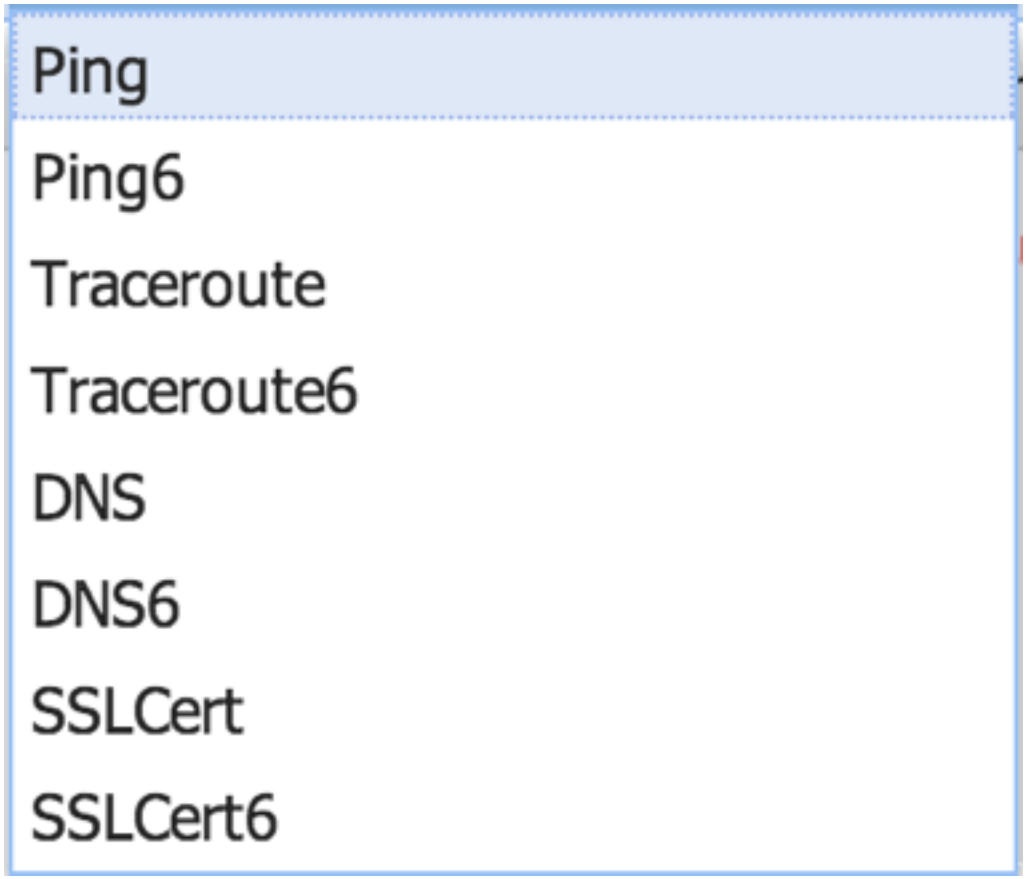
Test 3: RIPE Atlas

Atlas Features

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

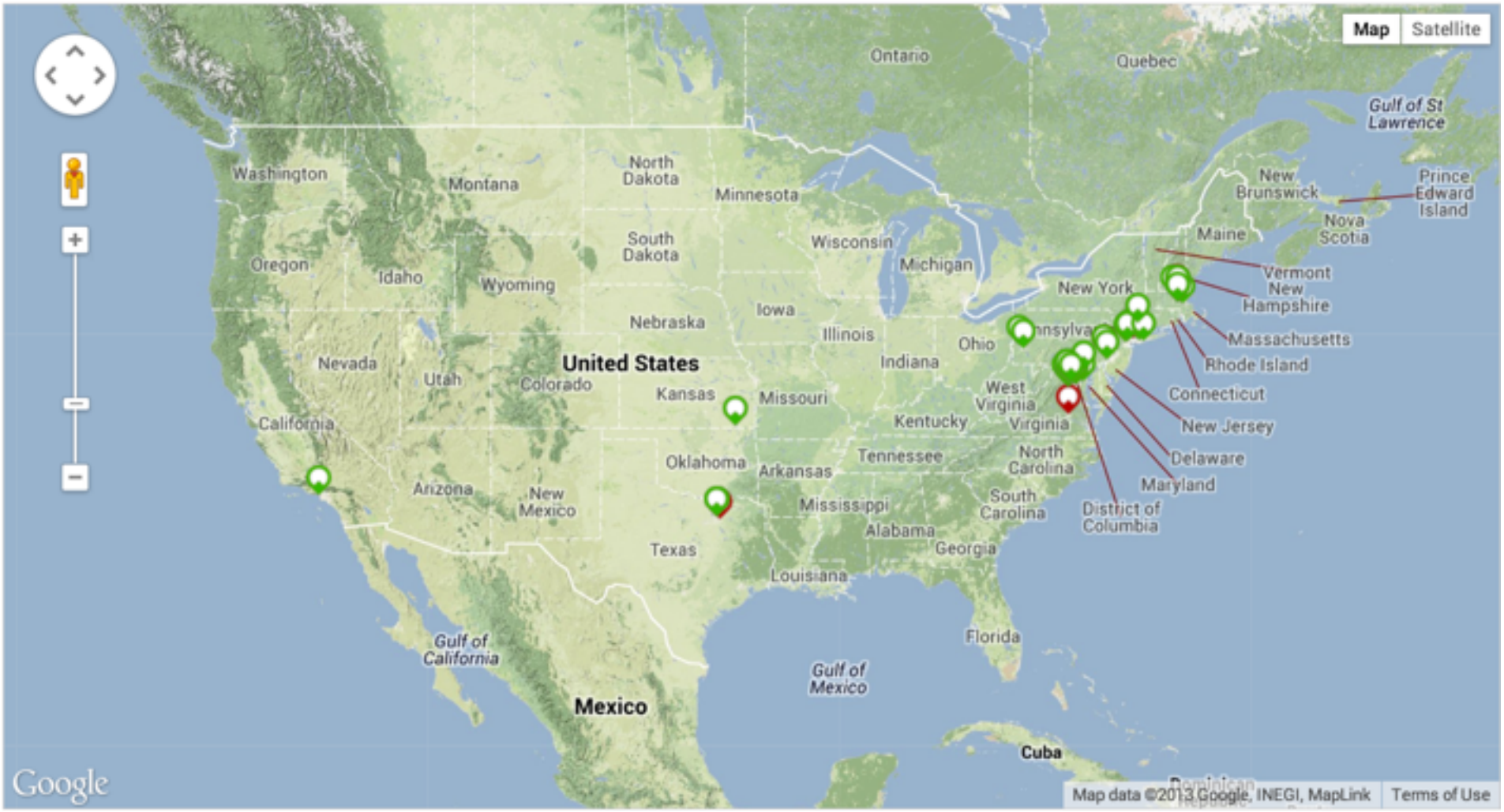


HTTP tests are in Beta

IPv6 reachability analysis by Atlas

Cool things that Beta testers had access to !

We need more Atlas probes in US



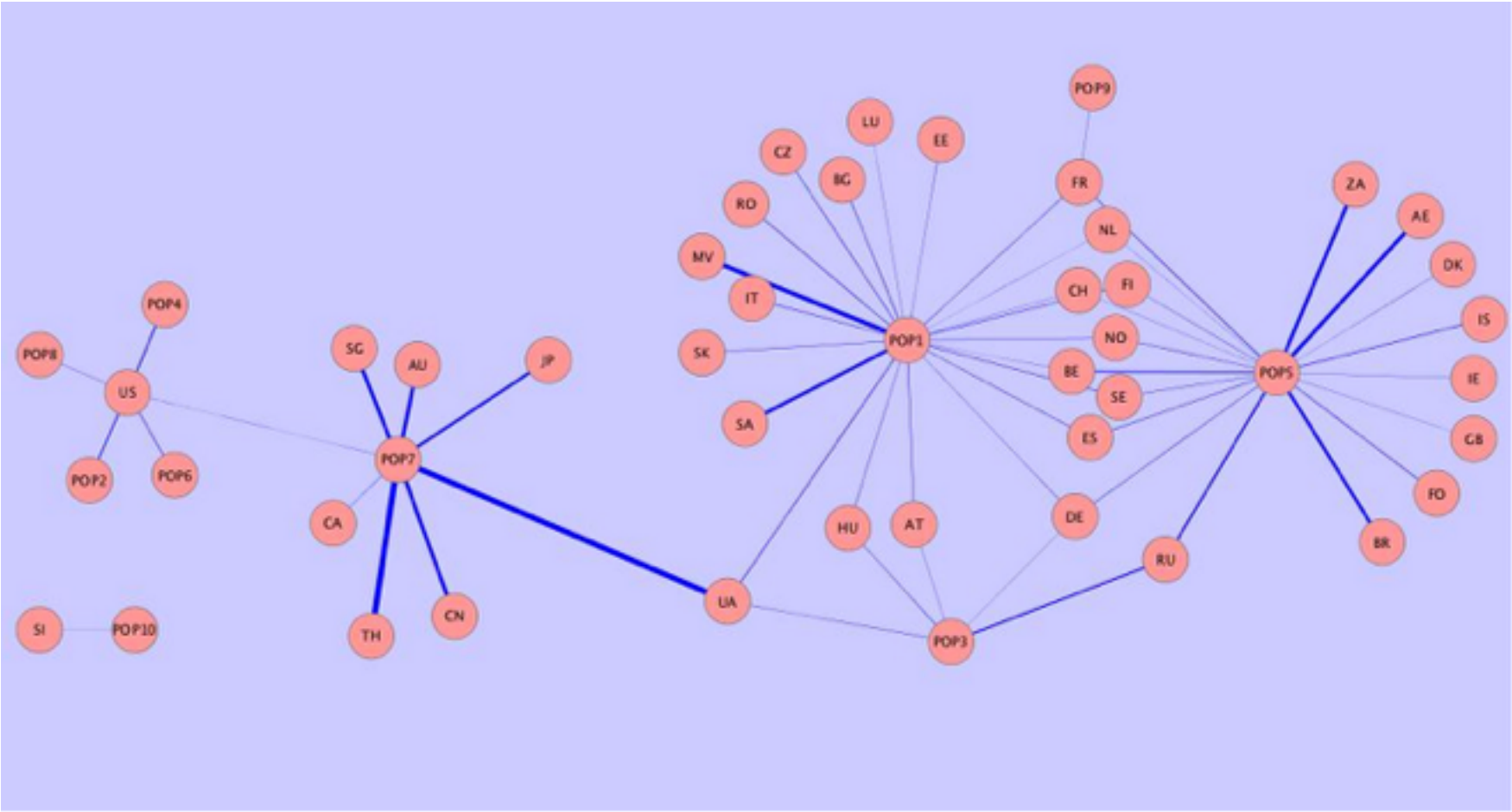
Looking Glass?

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

It can provide all the features of a looking glass expect BGP Lookups

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

Visualization of POP-Country link and latency



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

This report was published by RIPE

Test 3: Internal RUM

IPv4 only

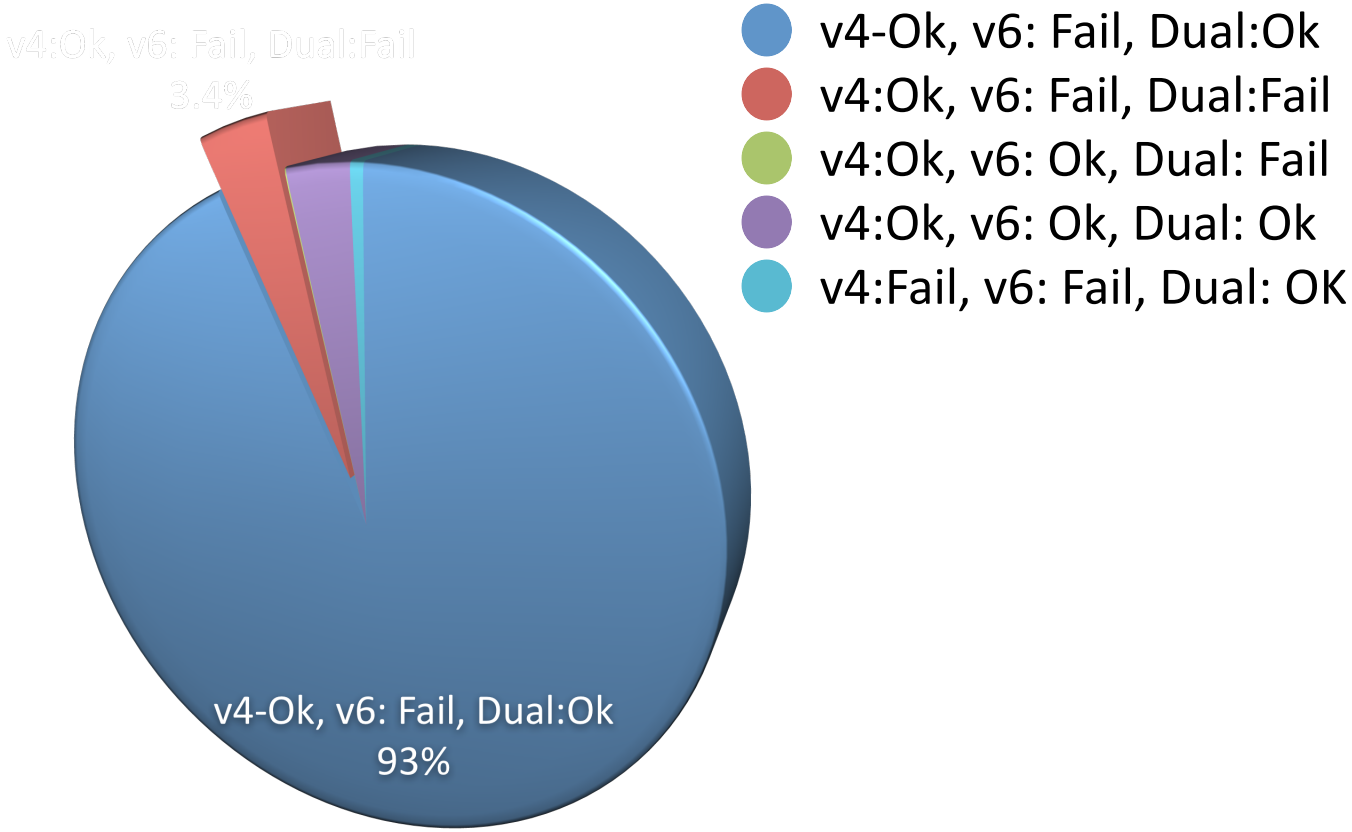
IPv6 only

Dual v4 and v6

Availability predictions before ipv6 launch

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

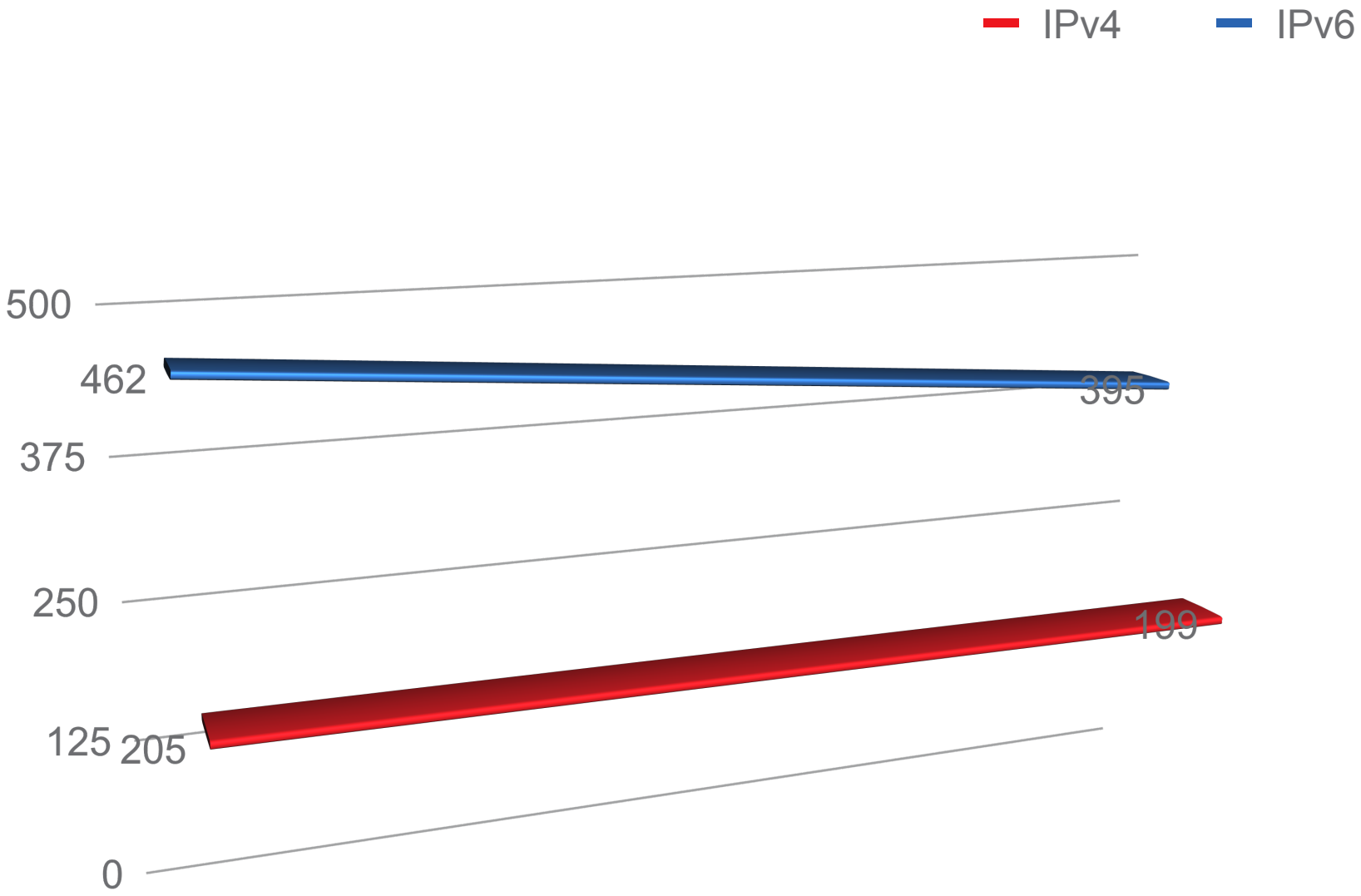
Availability predictions before ipv6 launch



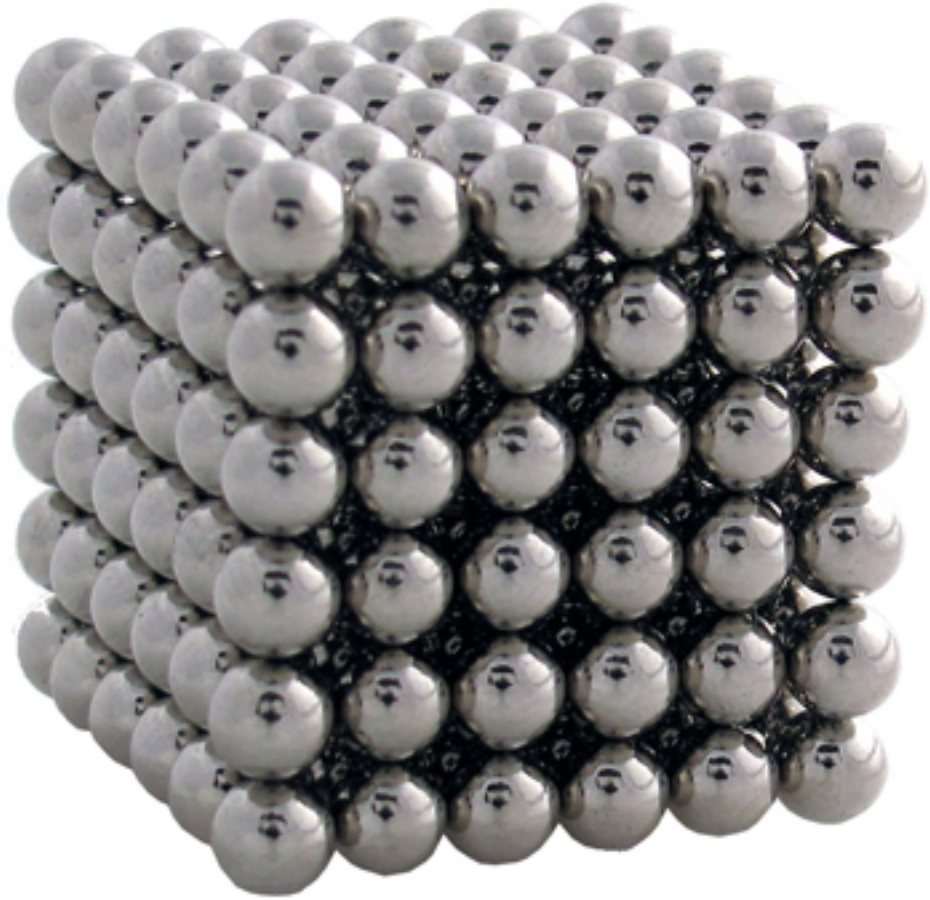
In order to reduce the failures to a more actionable set, the beacon also checked connectivity to ipv6.google.com. We first focused on cases where AS numbers fail to reach us over v6 but can browse ipv6.google.com

Latency Calculations

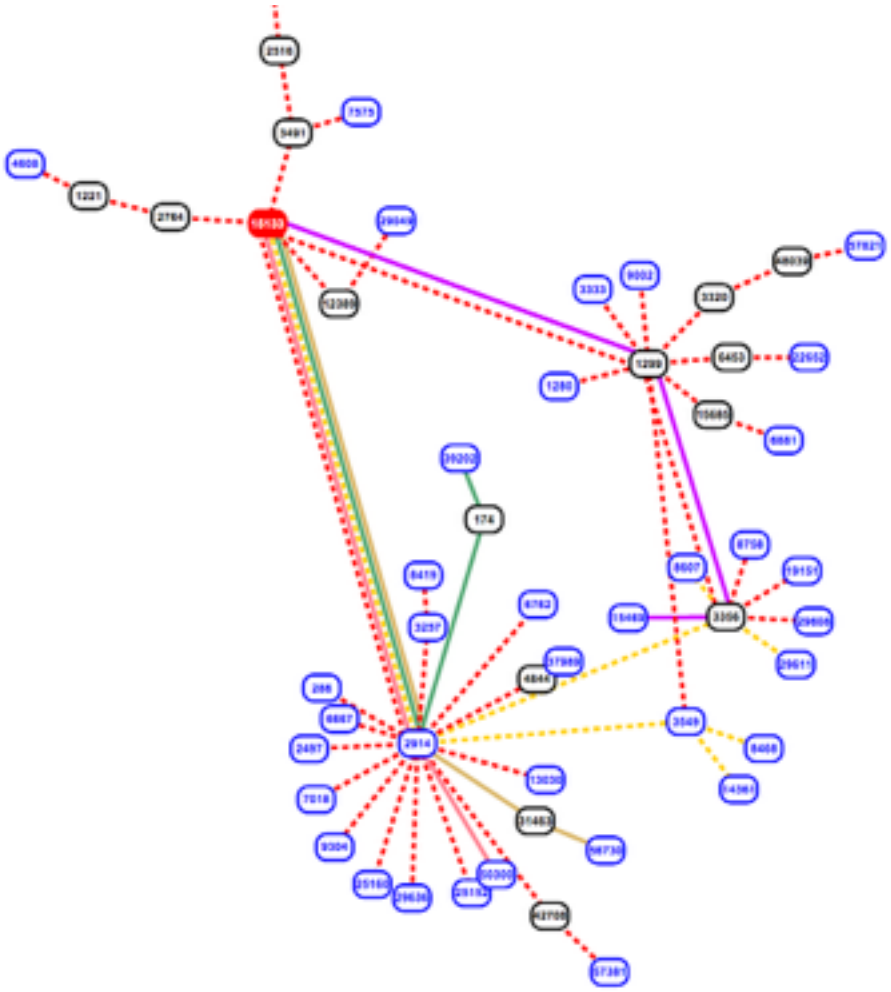
Latency Calculations



This is Anycast!

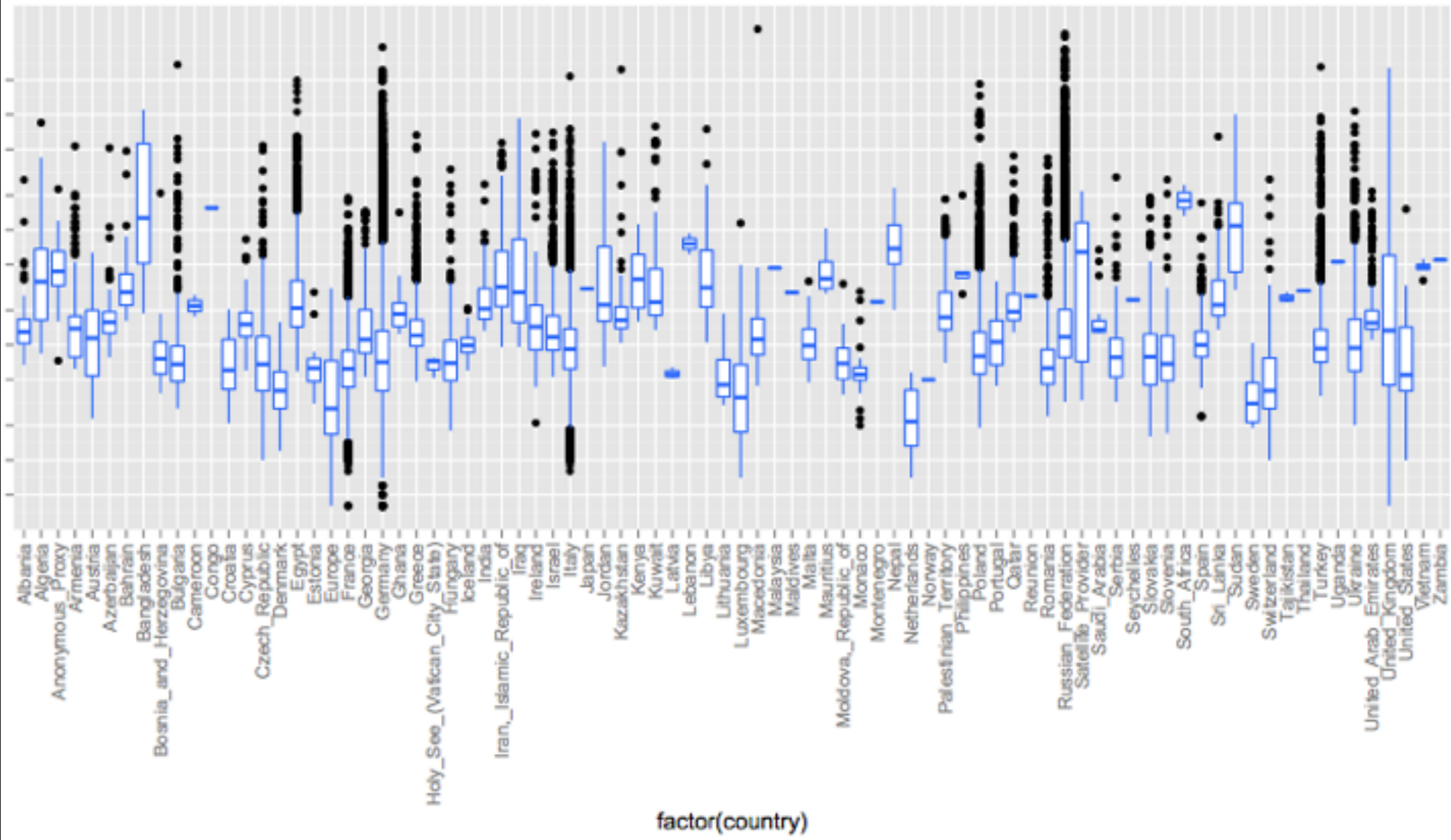


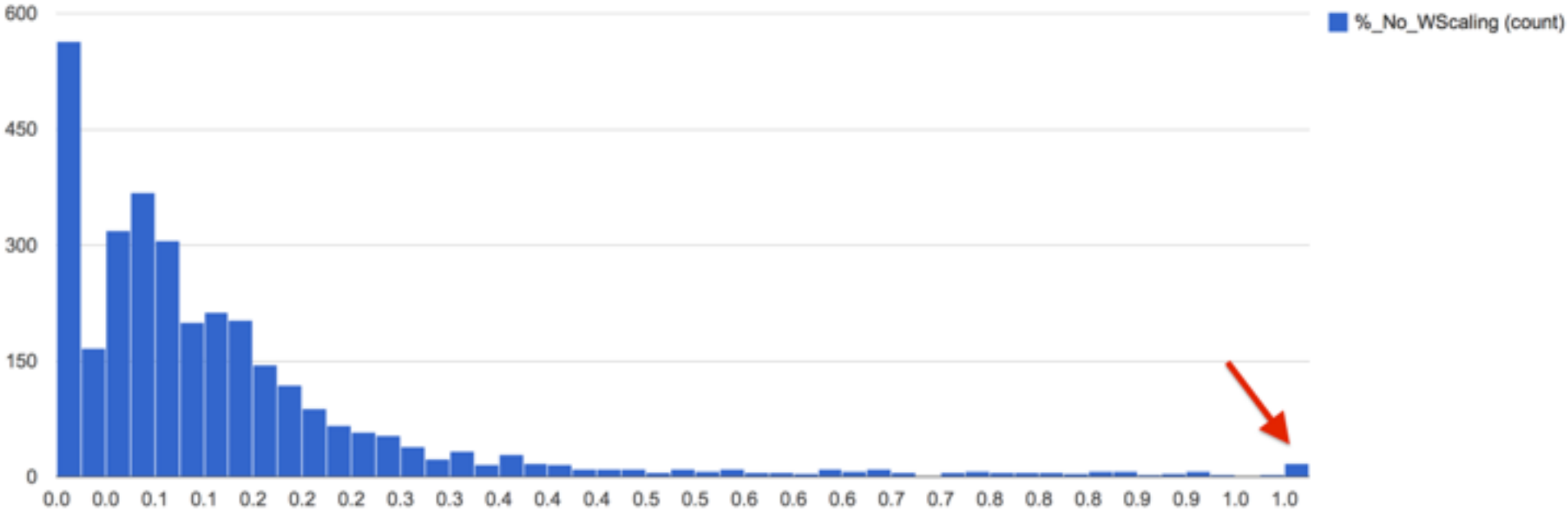
Biggest Enemy of Anycast -> Local Prefs



The AWESOME TCP_INFO


```
  tcp-info={"socket":28,"state":1,"ca_state":
0,"retransmits":0,"probes":0,"backoff":
0,"options":6,"snd_wscale":6,"rcv_wscale":
6,"rto":240000,"ato":40000,"snd_mss":
1460,"rcv_mss":536,"unacked":44,"sacked":
0,"lost":0,"retrans":0,"fackets":
0,"last_data_sent":0,"last_ack_sent":
0,"last_data_recv":40,"last_ack_recv":0,"pmtu":
1500,"rcv_ssthresh":16744,"rtt":40000,"rttvar":
7500,"snd_ssthresh":2147483647,"snd_cwnd":
50,"advmss":1460,"reordering":3,"rcv_rtt":
0,"rcv_space":14600,"total_retrans":0}
```


Albania
Algeria
Anonymous_Proxy
Armenia
Austria
Azerbaijan
Bahrain
Bangladesh
Bosnia_and_Herzegovina
Bulgaria
Cameroon
Congo
Croatia
Cyprus
Czech_Republic
Denmark
Egypt
Estonia
Europe
France
Georgia
Germany
Ghana
Greece
Holy_See_(Vatican_City_State)
Hungary
Iceland
India
Iran,_Islamic_Republic_of
Iraq
Ireland
Israel
Italy
Japan
Jordan
Kazakhstan
Kenya
Kuwait
Latvia
Lebanon
Libya
Lithuania
Luxembourg
Macedonia
Malaysia
Maldives
Malta
Mauritius
Moldova,_Republic_of
Monaco
Montenegro
Nepal
Netherlands
Norway
Palestinian_Territory
Philippines
Poland
Portugal
Qatar
Reunion
Romania
Russian_Federation
Satellite_Provider
Saudi_Arabia
Serbia
Seychelles
Slovakia
Slovenia
South_Africa
Spain
Sri_Lanka
Sudan
Sweden
Switzerland
Tajikistan
Thailand
Turkey
Uganda
Ukraine
United_Arab_Emirates
United_Kingdom
United_States
Vietnam
Zambia
factor(country)


600
450
300
150
0
0.0 0.0 0.1 0.1 0.2 0.2 0.2 0.3 0.3 0.4 0.4 0.4 0.5 0.5 0.6 0.6 0.6 0.7 0.7 0.8 0.8 0.8 0.9 0.9 1.0 1.0
%_No_WScaling (count)

The Problem !

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

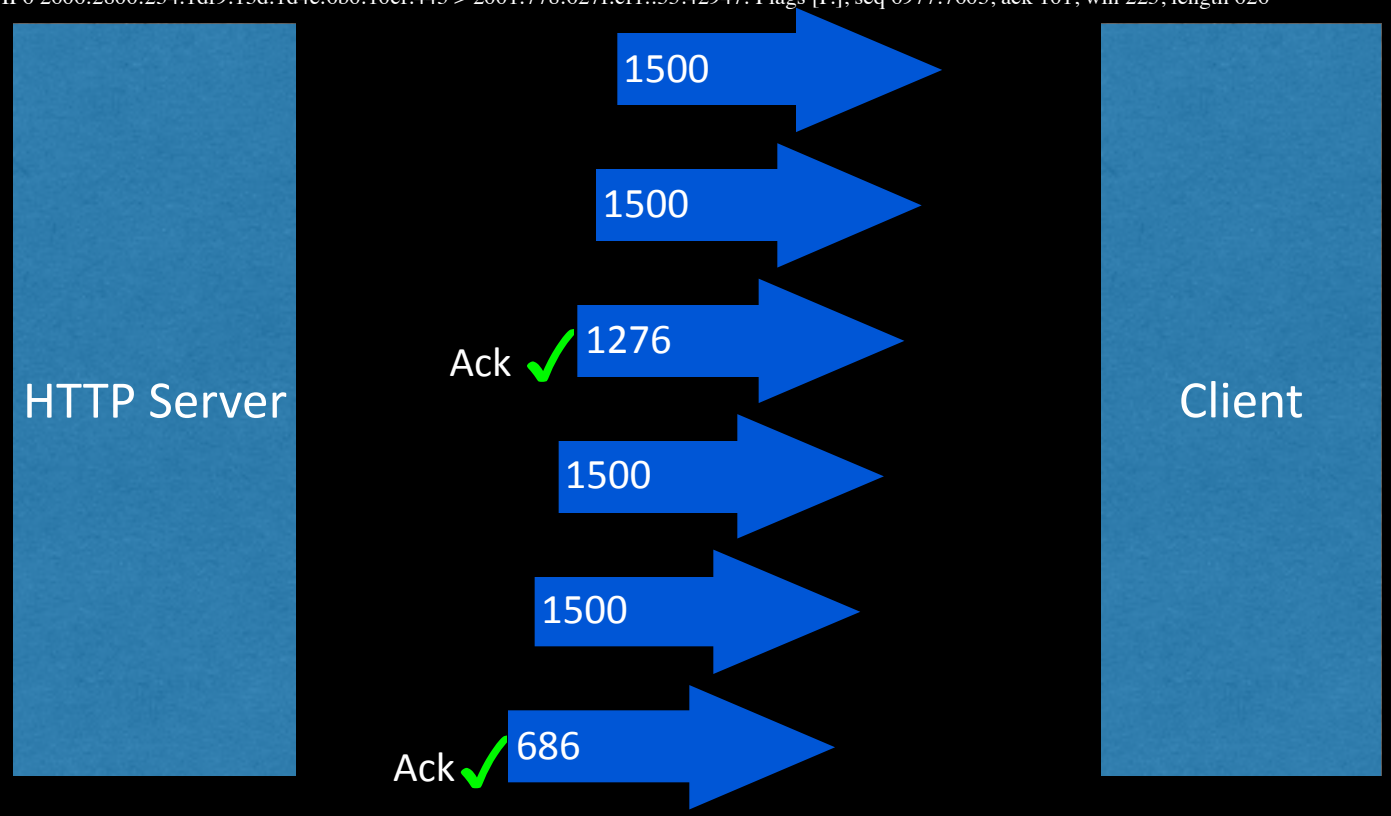
16:21:13.051367 IP6 2606:2800:234:1df9:13d:1d4e:6b0:10cf.443 > 2001:778:627f:cf1::55.42947: Flags [.], seq 1441:2881, ack 101, win 225, length 1440

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

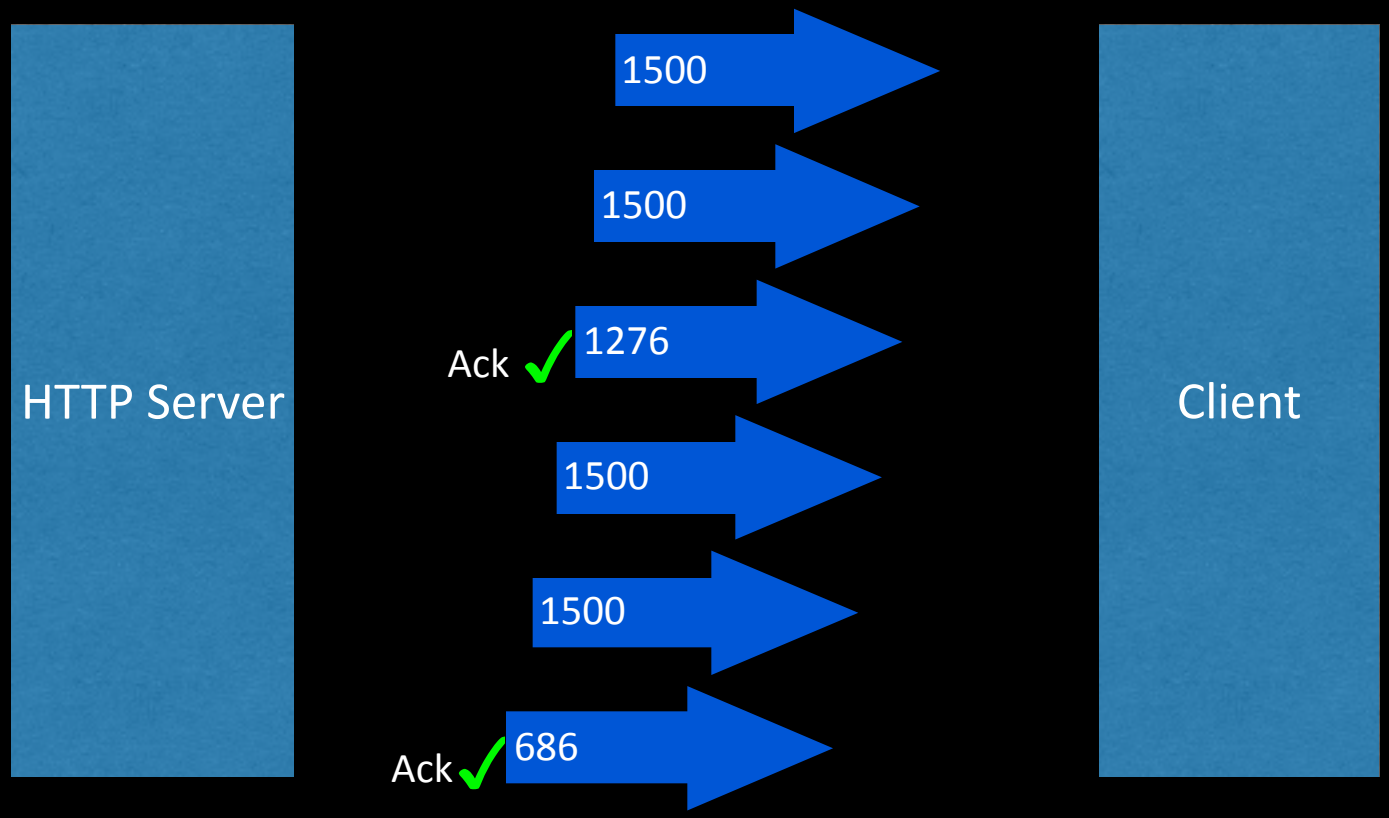
16:21:13.051427 IP6 2606:2800:234:1df9:13d:1d4e:6b0:10cf.443 > 2001:778:627f:cf1::55.42947: Flags [.], seq 5537:6977, ack 101, win 225, length 1440

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

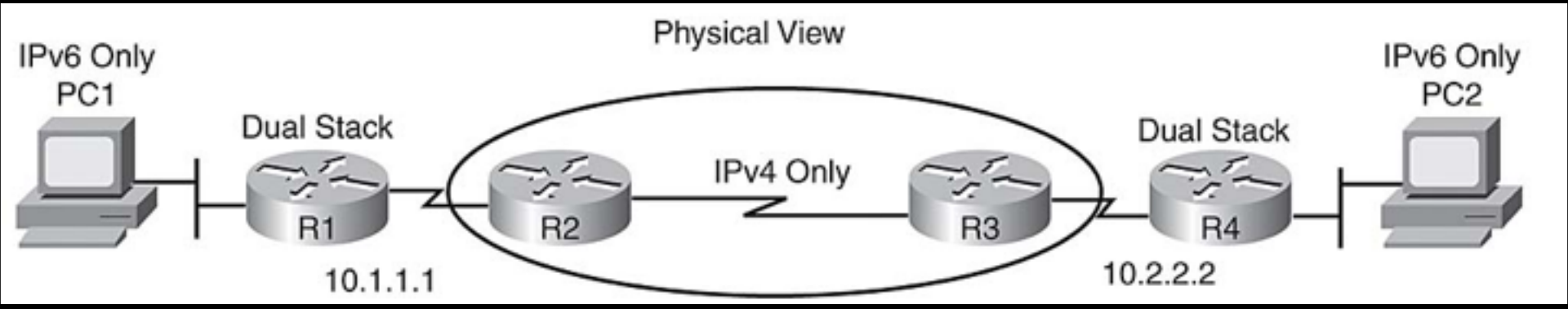


This is a clear sign of a path MTU problem

Lets look at it again

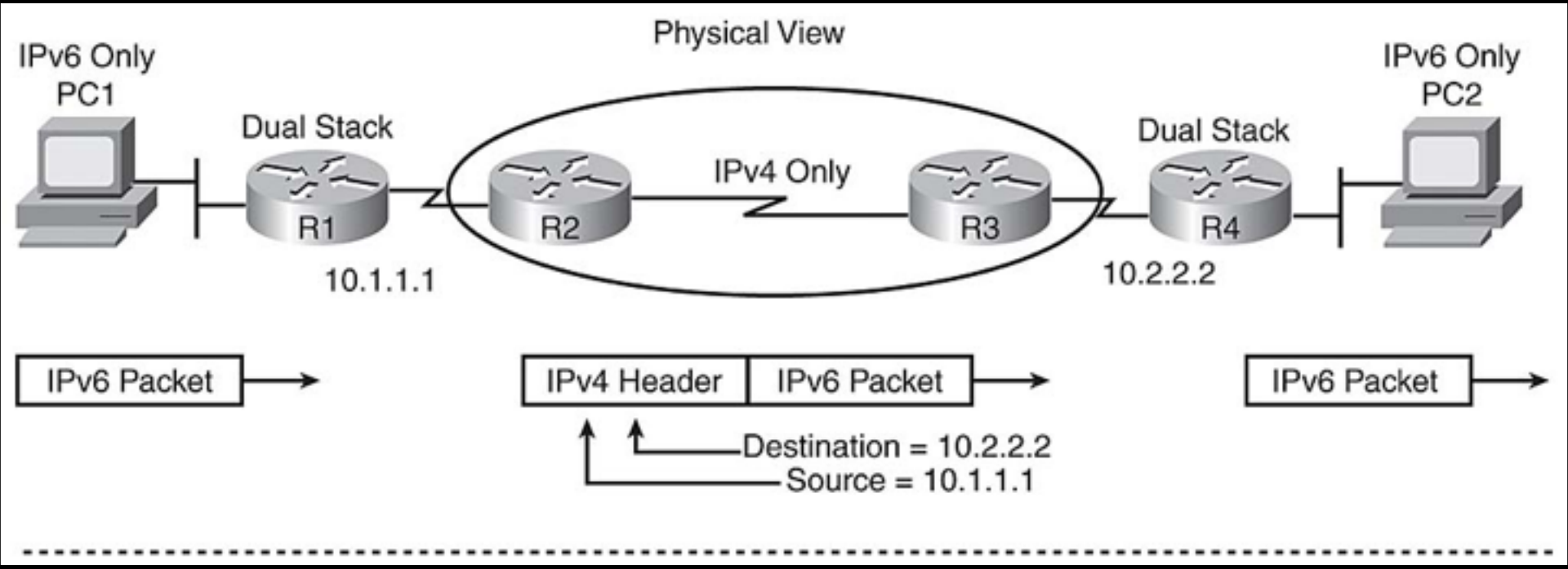


Lets try to explain what is happening here



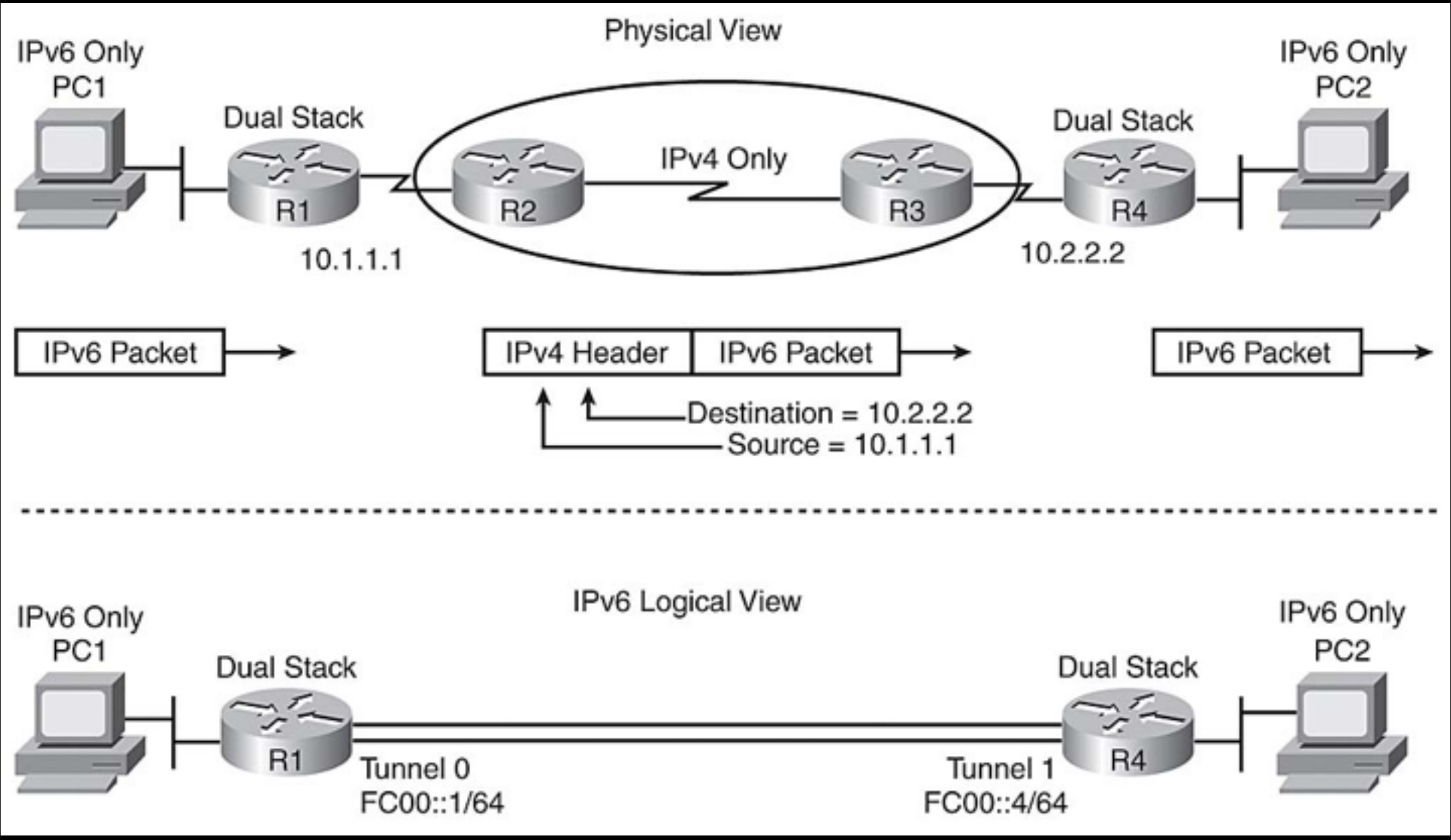
http://www.networkworld.com/subnets/cisco/082307-ch7-ccna-icnd2.html

Lets try to explain what is happening here



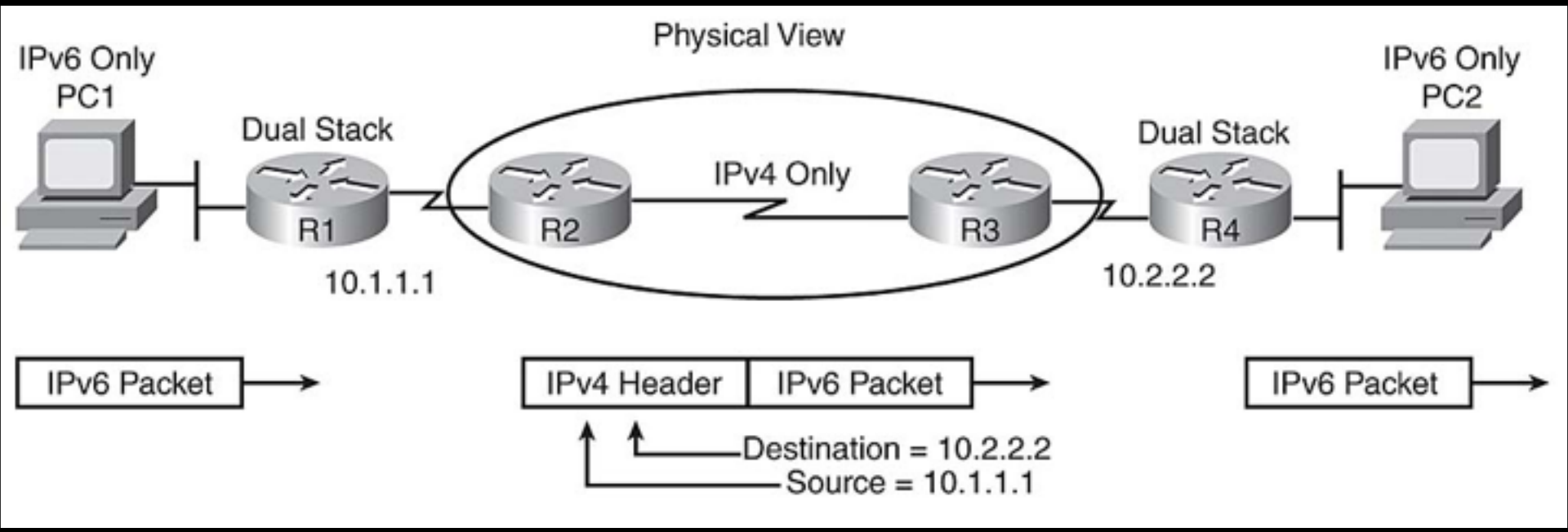
http://www.networkworld.com/subnets/cisco/082307-ch7-ccna-icnd2.html

Lets try to explain what is happening here



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

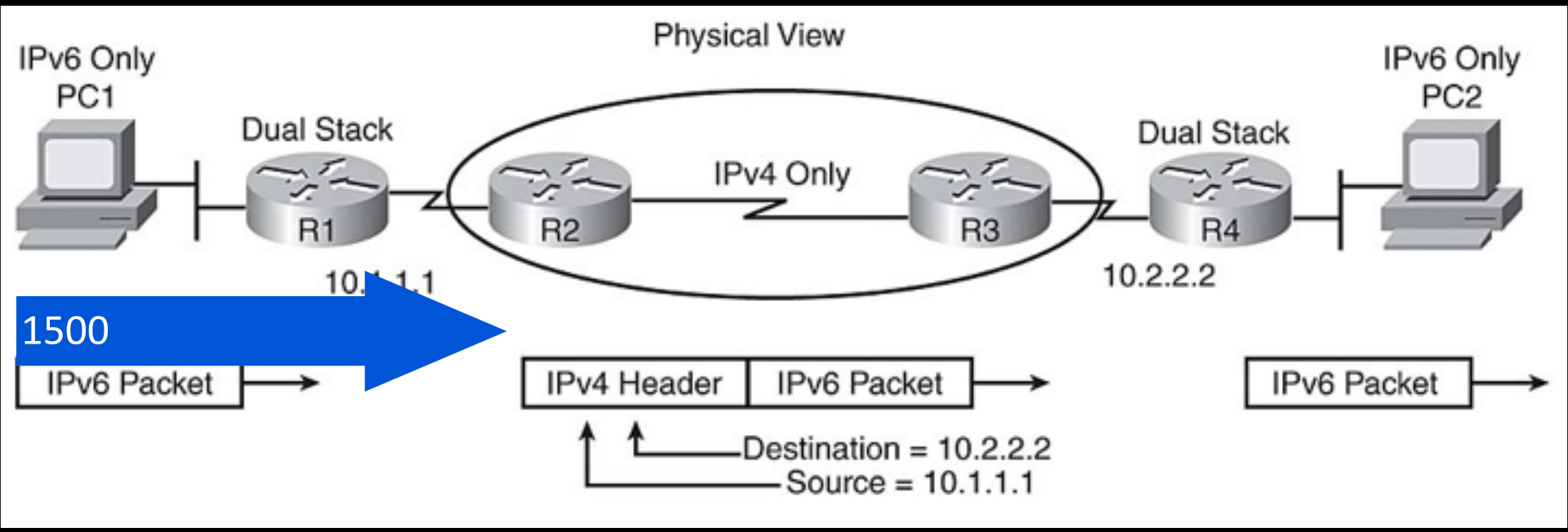
But there is already a mechanism to prevent this from happening



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

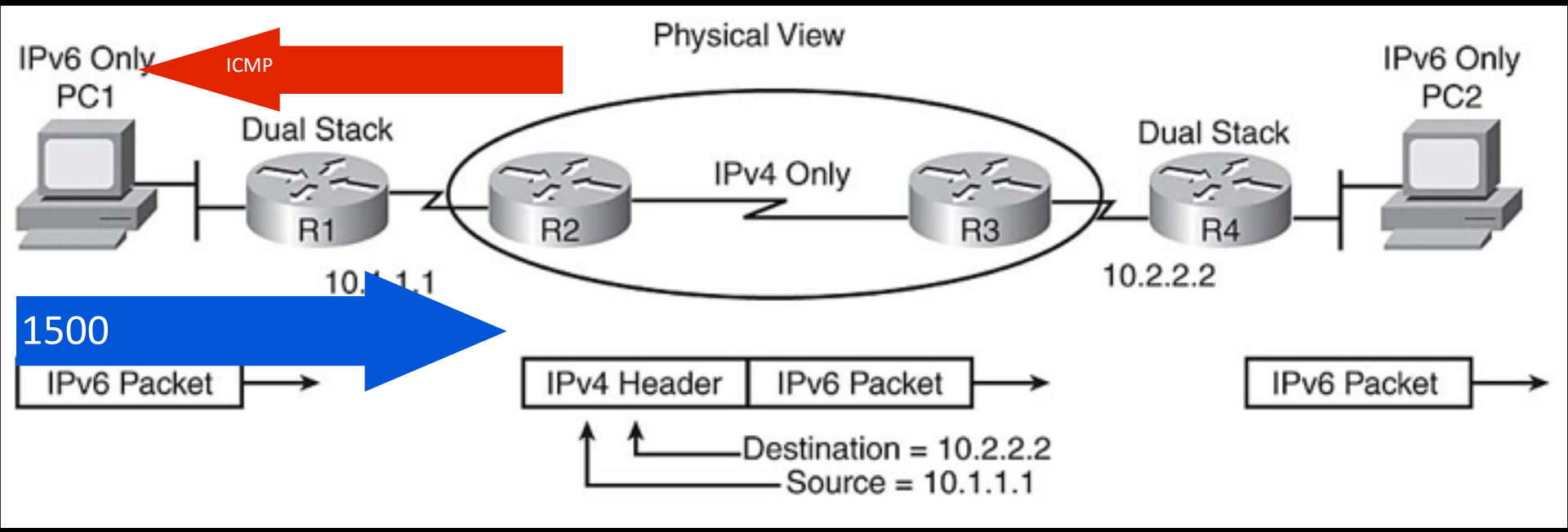
But there is already a mechanism to prevent this from happening



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

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 Frankfurt

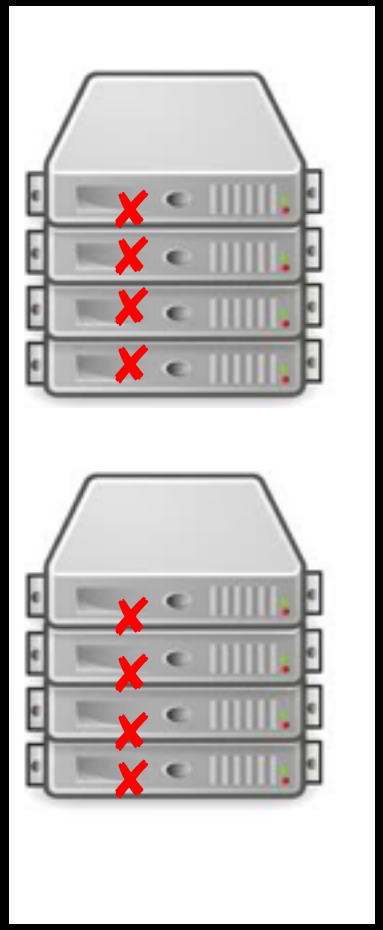
No!

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

Server handling the flow in Frankfurt

No!

All other servers in Frankfurt



No!

We searched our entire platform for that ICMP message

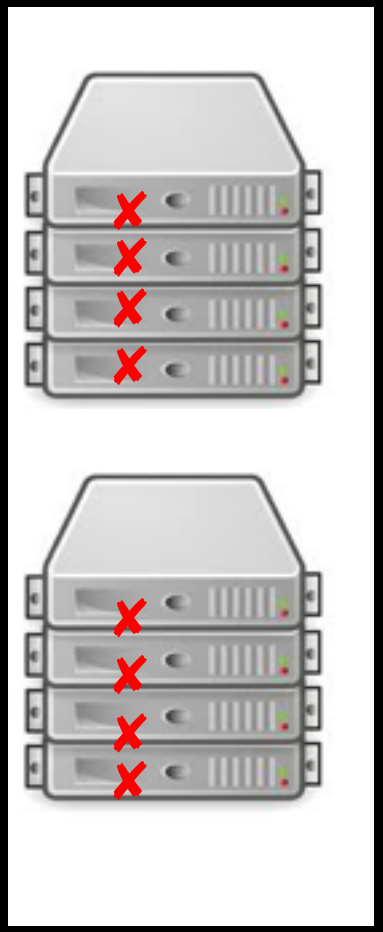
…and found the ICMP packet in Paris!

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

Server handling the flow in Frankfurt

No!

All other servers in Frankfurt



No!

How about all the servers in the world?

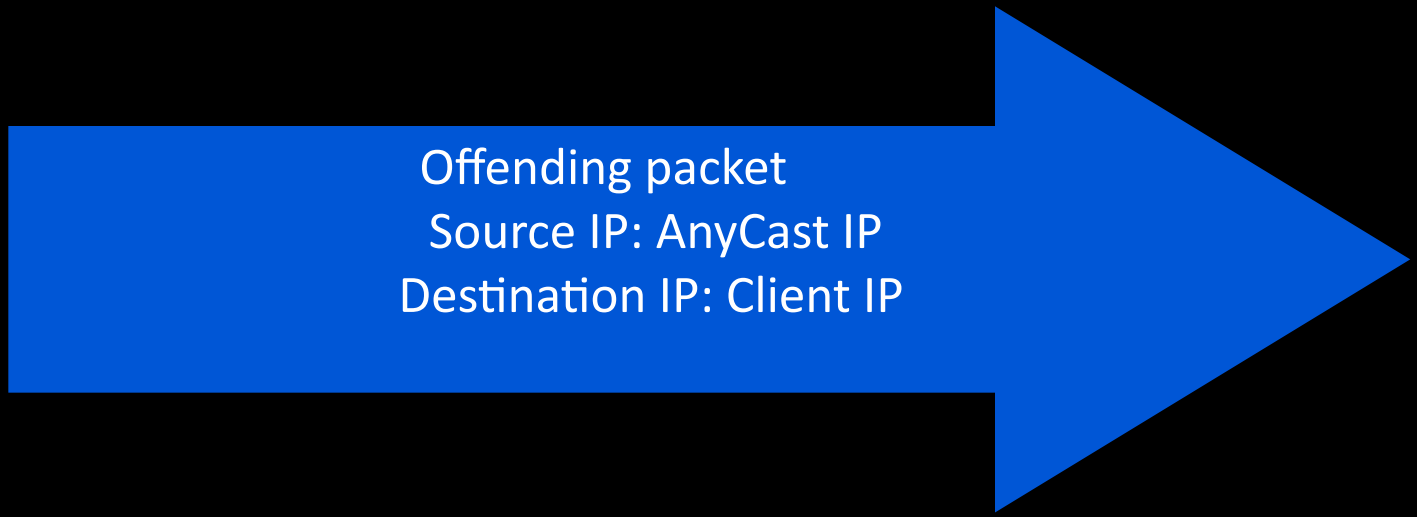
Found it in Paris!

Why that packet was received in paris?

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

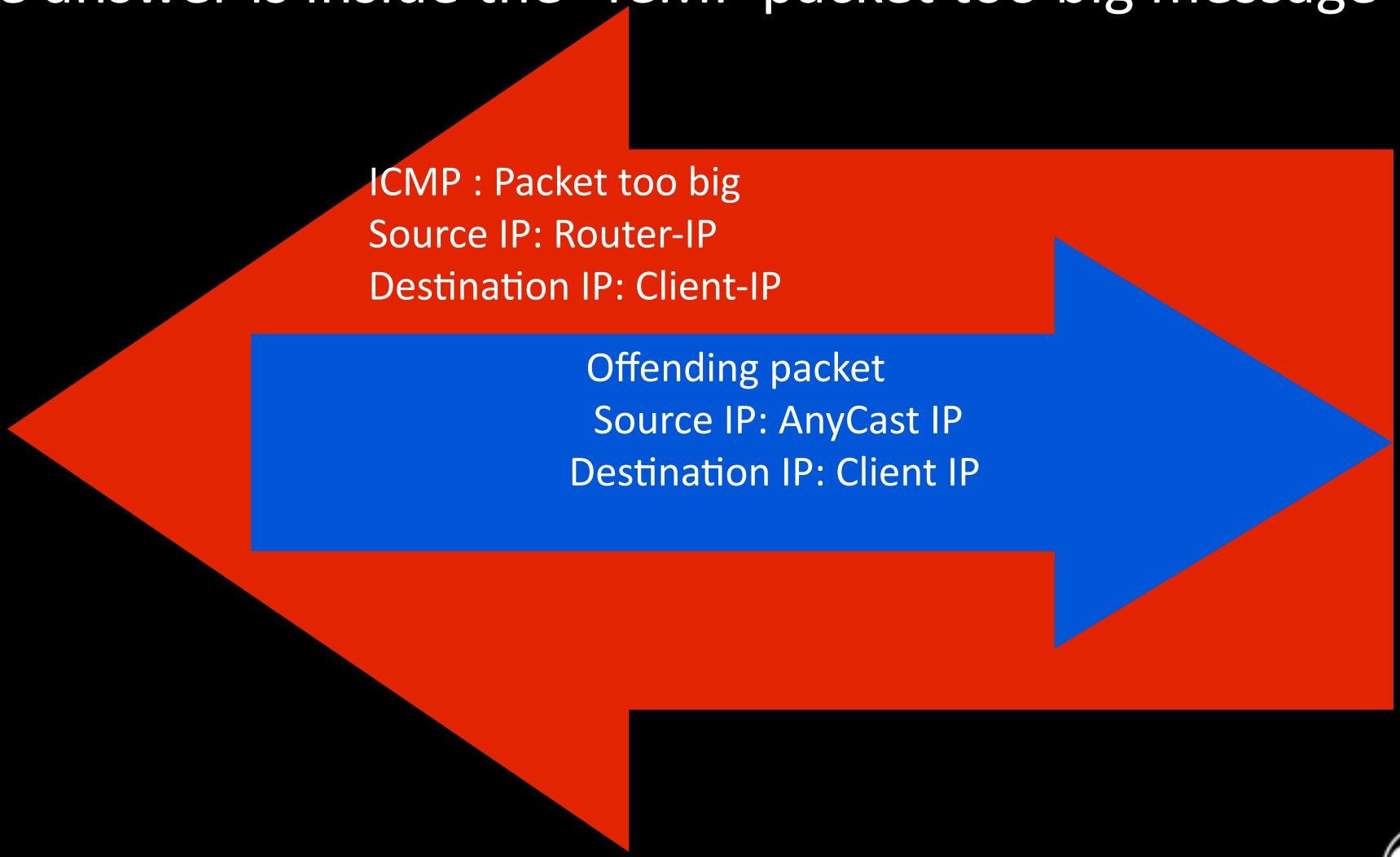
Why that packet was received in paris?

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



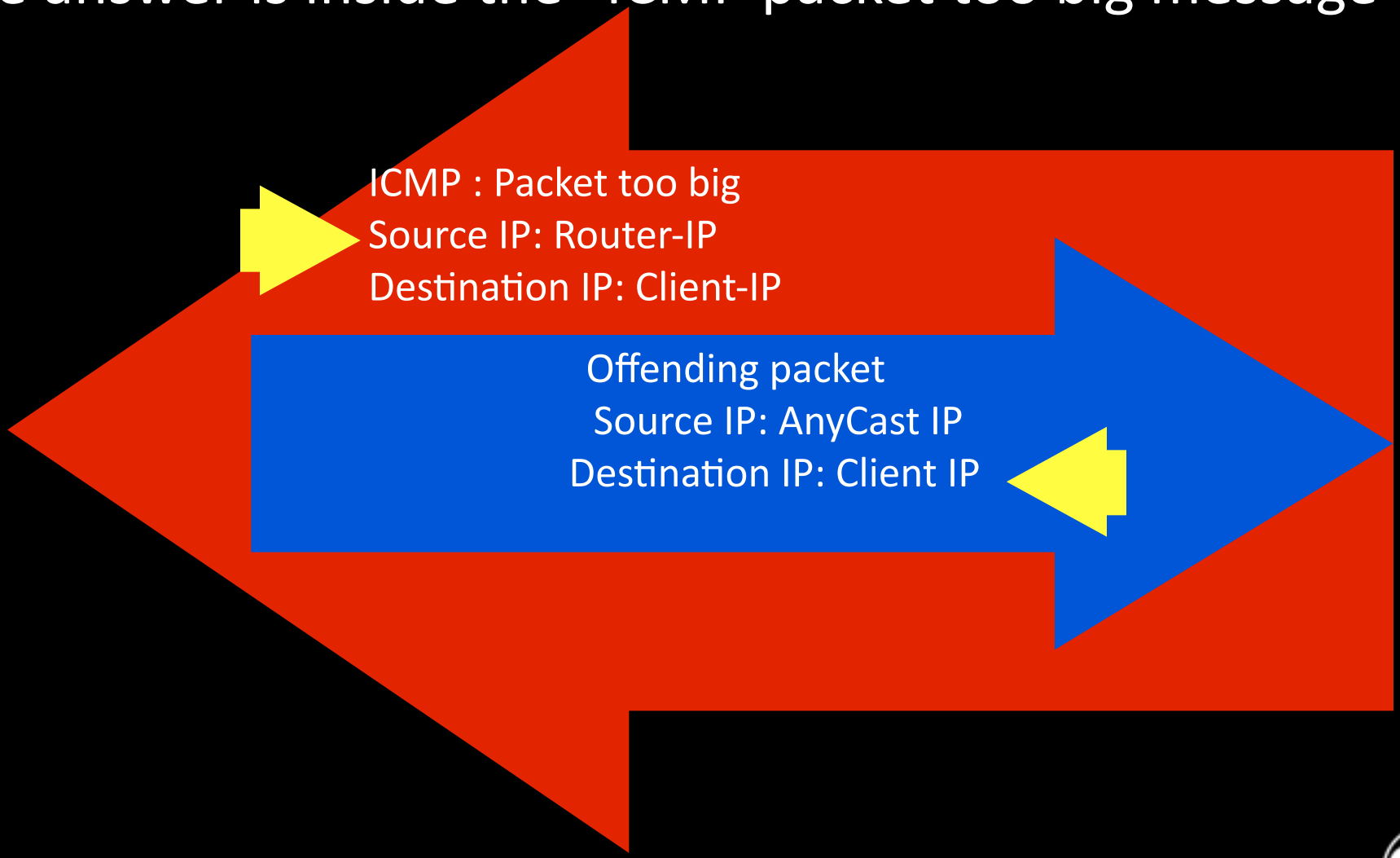
Why that packet was received in paris?

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



Why that packet was received in paris?

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



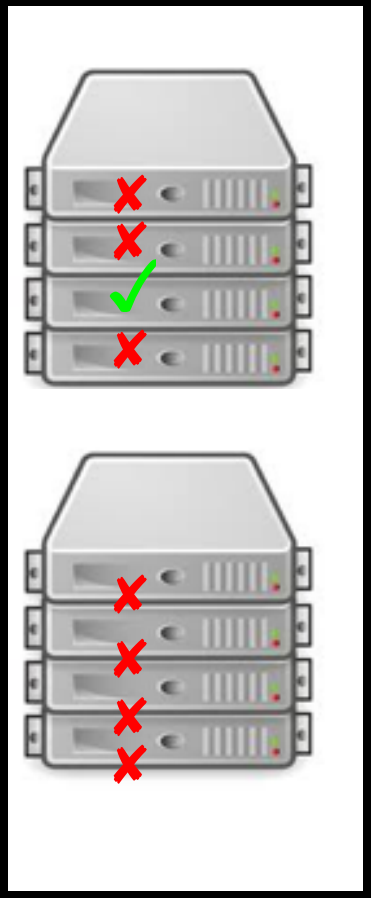
Client -> AnyCast -> Frankfurt

Router -> AnyCast -> Paris

so we fixed the peering with HE. What happened next?

Server Handling the flow in Frankfurt

No!

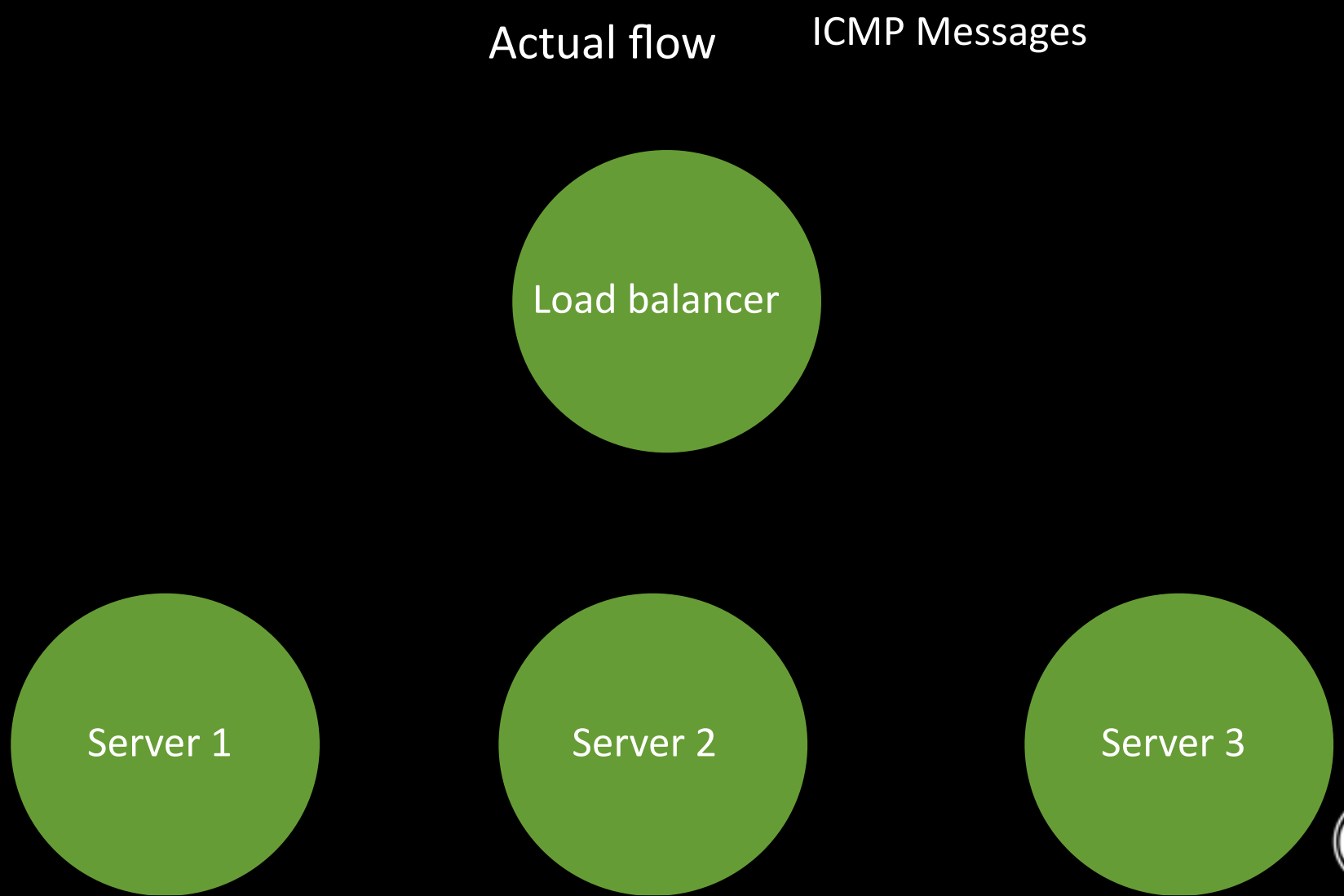


All other servers in Frankfurt

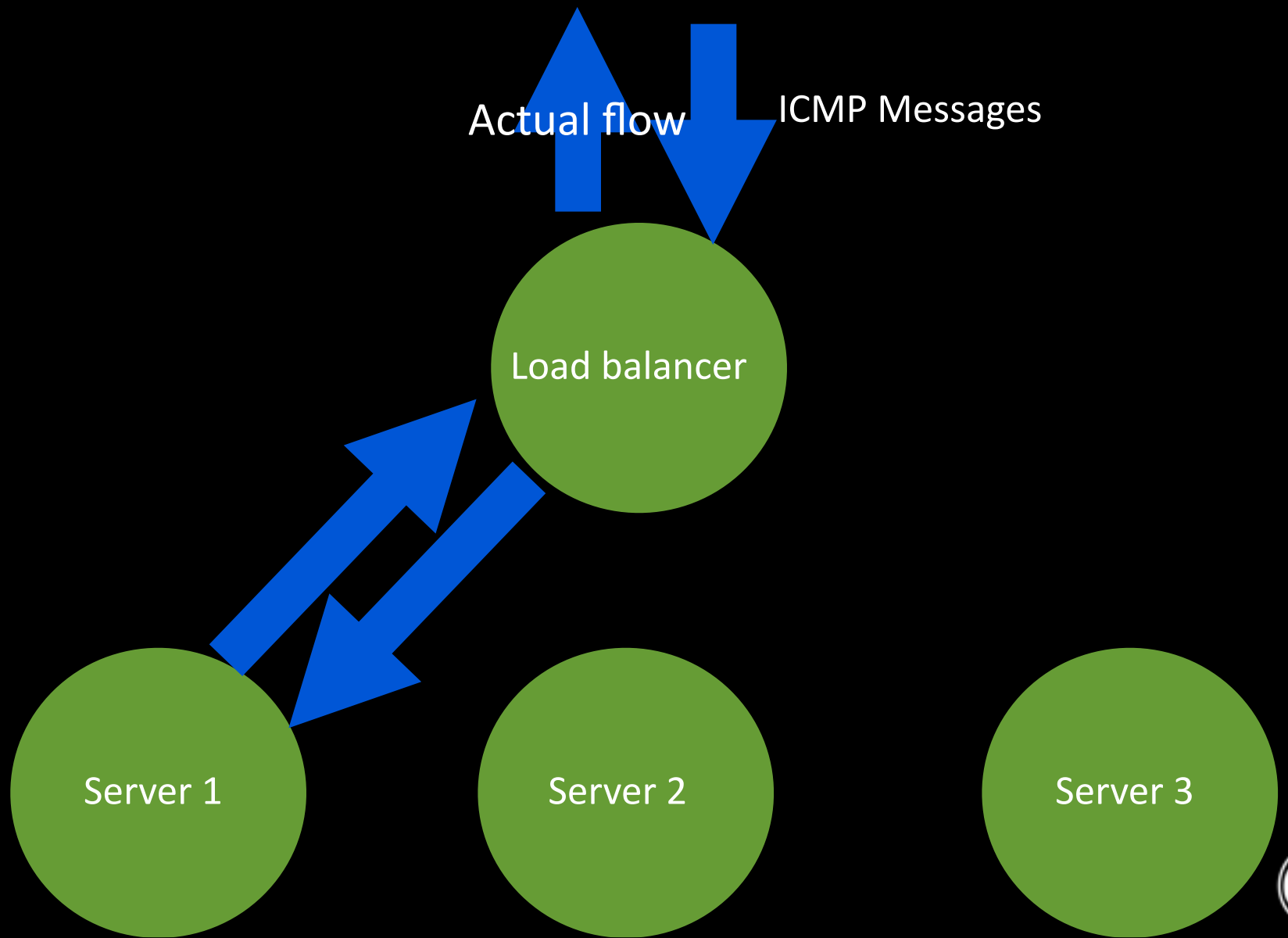
Now we receive the packet in the right pop, but by the wrong server

SRC-DST Based HASH

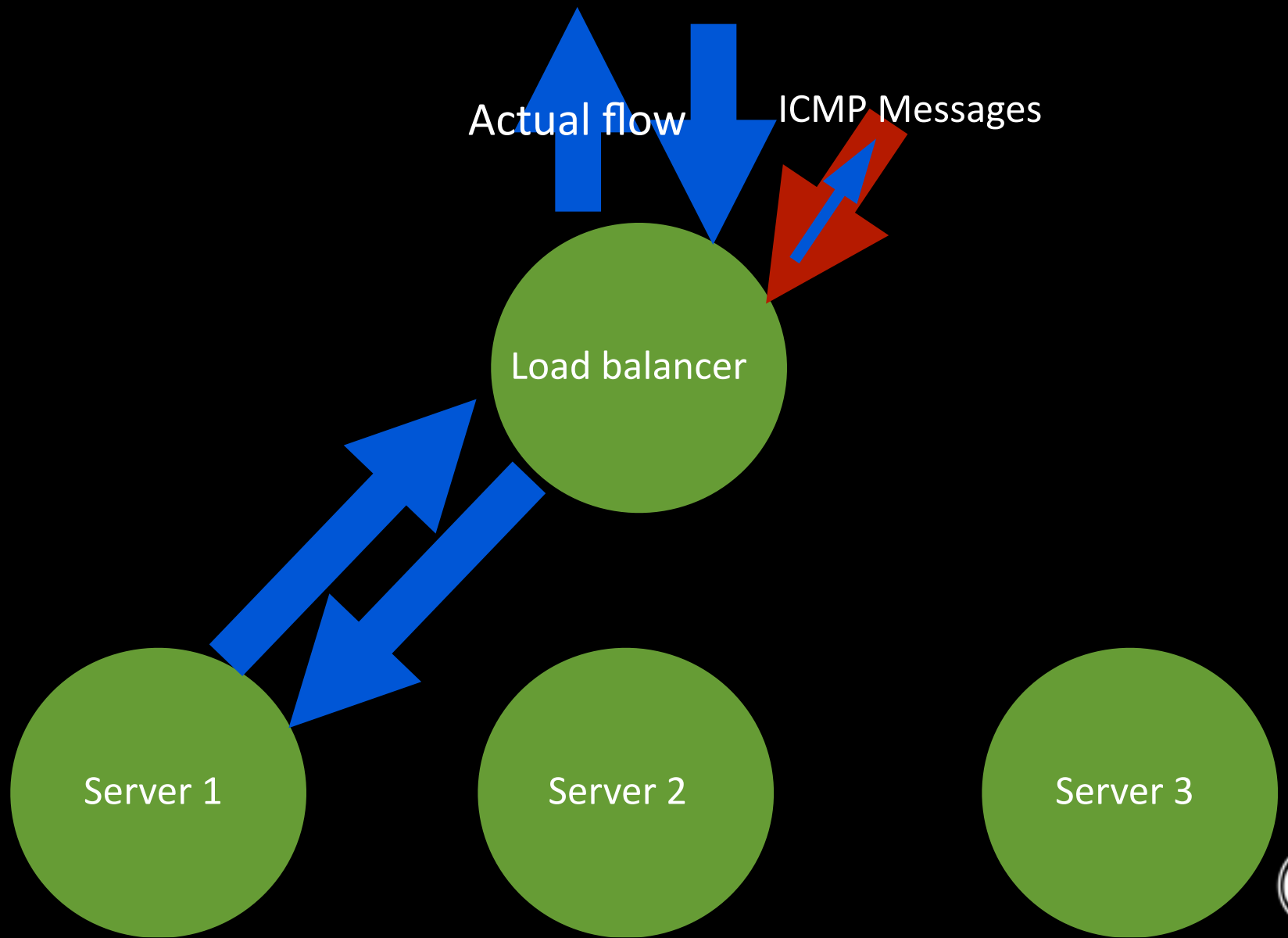
Load balancers using simple hash do not check offending ICMP packets to make the forwarding decision



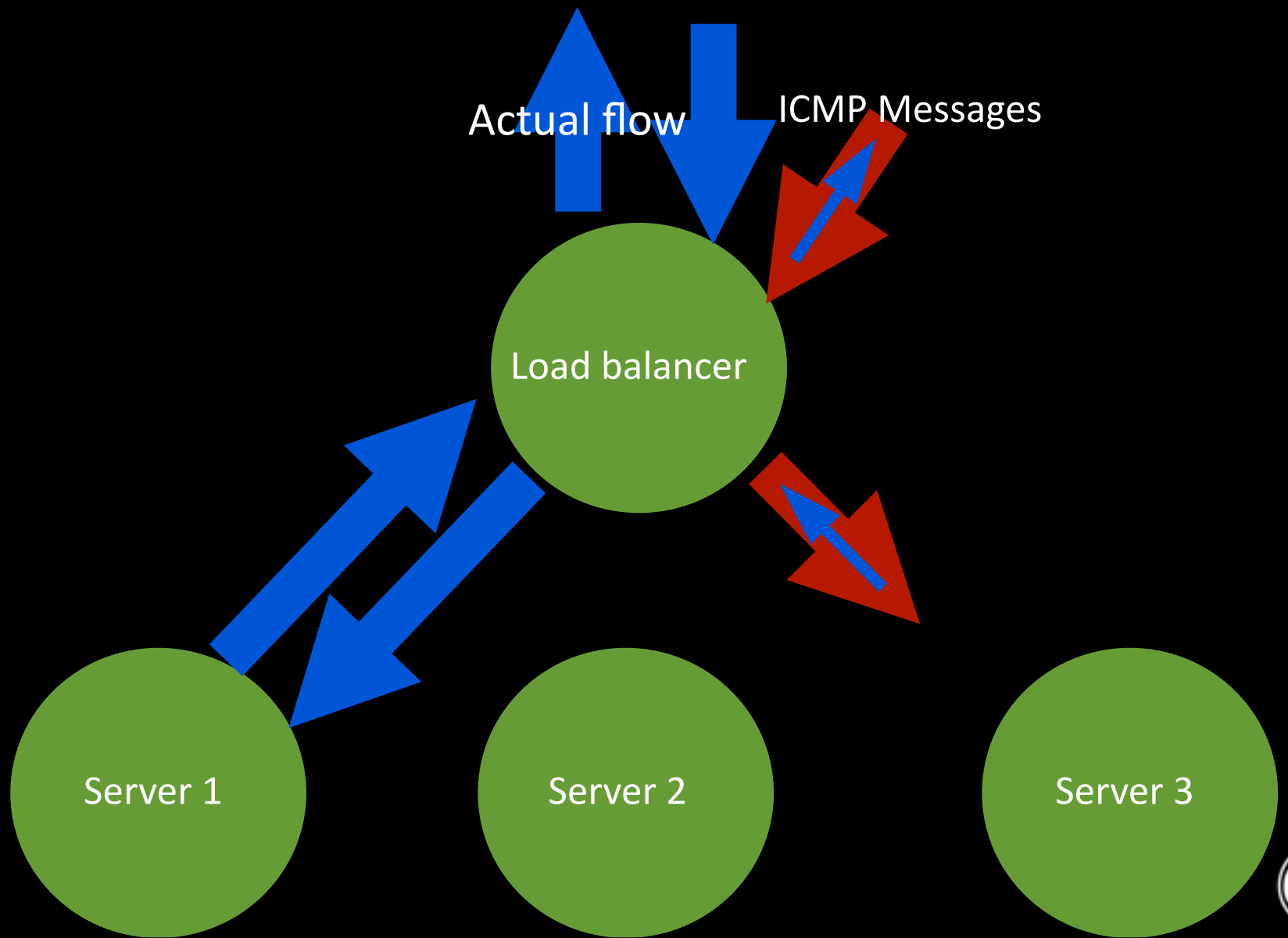
Load balancers using simple hash do not check offending ICMP packets to make the forwarding decision



Load balancers using simple hash do not check offending ICMP packets to make the forwarding decision



Load balancers using simple hash 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.

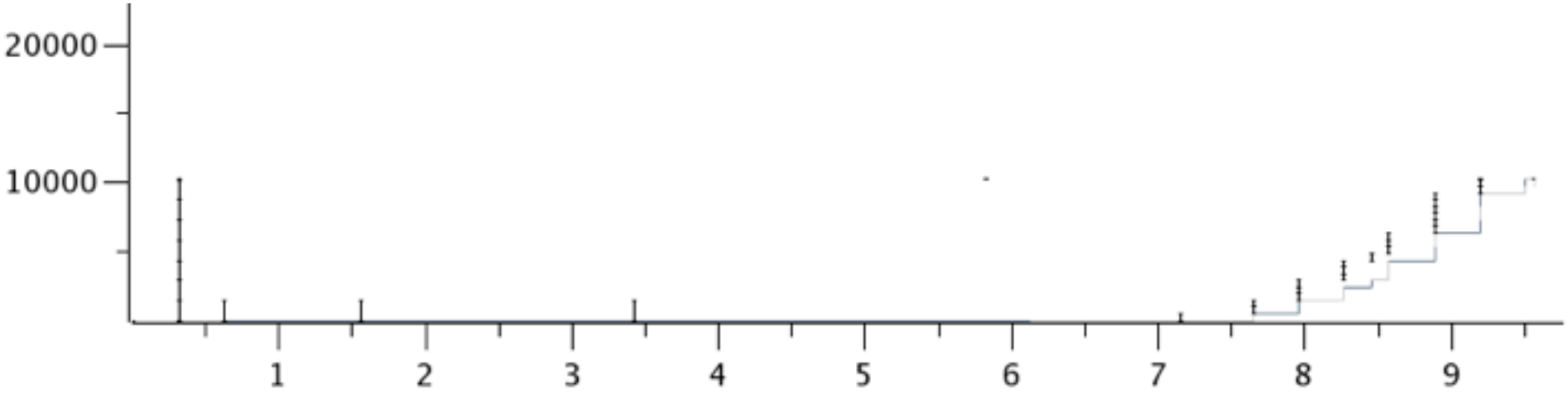
The Bigger Problem:

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

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.

MTU Probing?



Higher Availability at the cost of Response Time

Thank You for your time

HOSSEIN LOTFI

HLOTFI @ EdgeCast .com