A New Anycast DNS Measurement

RIPE 72

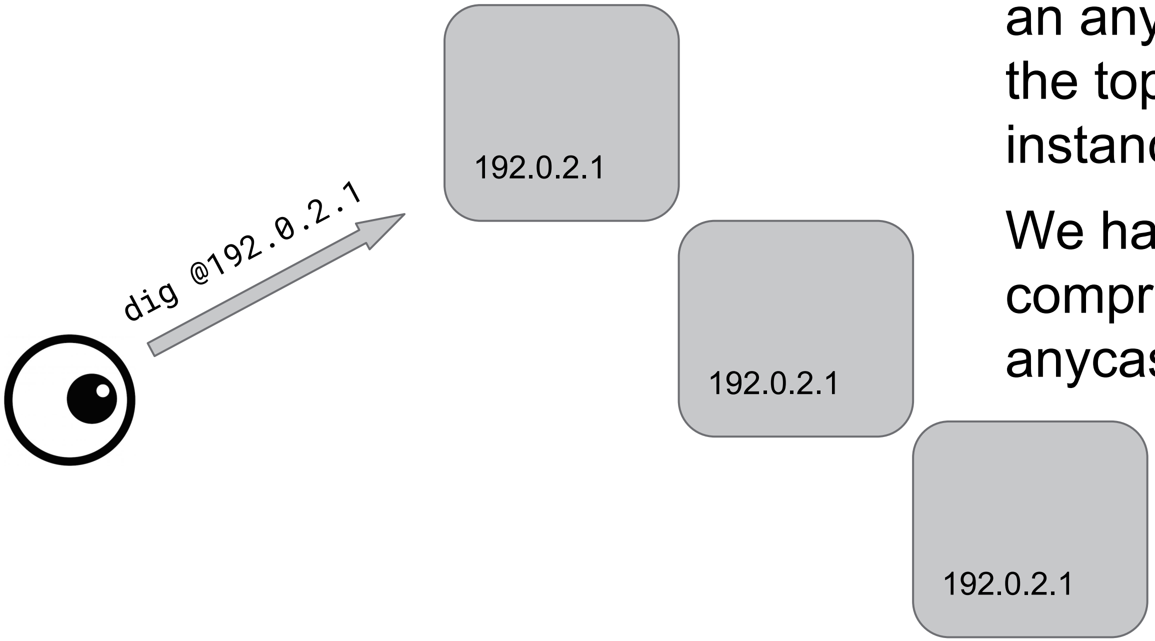
Dave Knight

May 2016

INTERNET
PERFORMANCE.
DELIVERED.

dyn.com @dyn

Problem statement

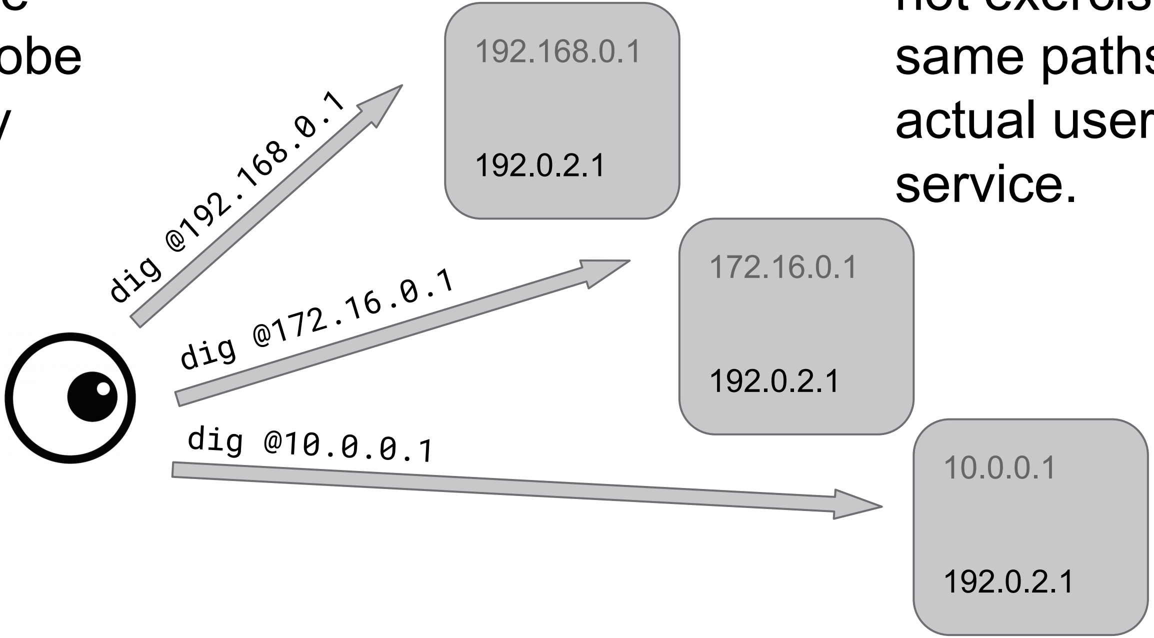


A monitoring node cannot directly probe all instances of an anycast service, so only the topologically least distant instance is visible.

We have to make compromises to monitor an anycast service.

A compromise

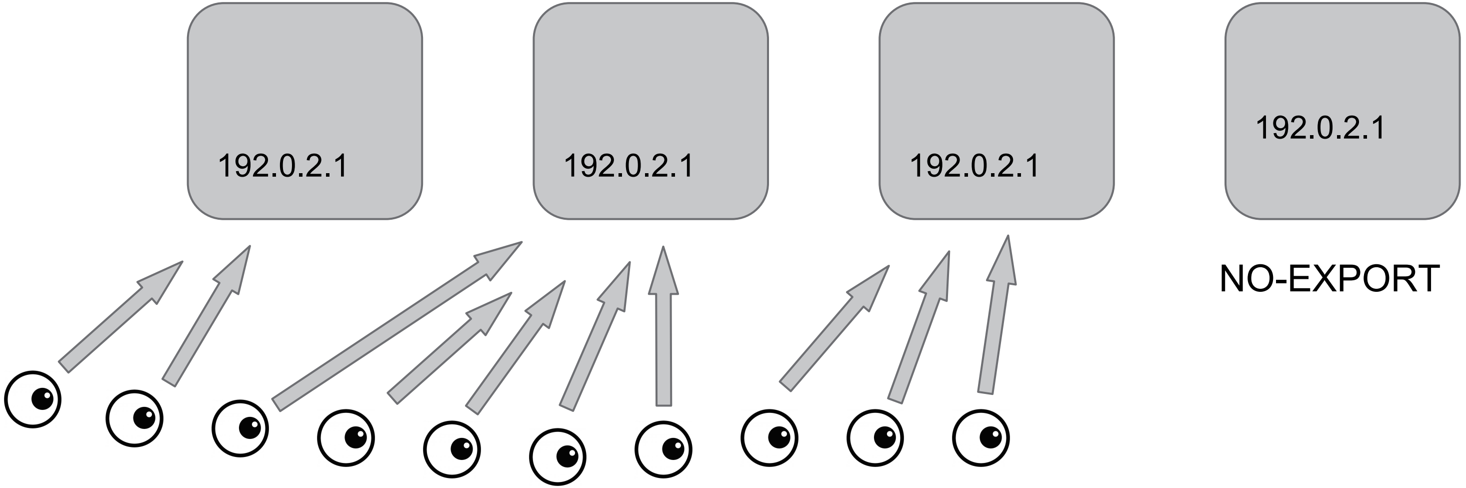
A single monitoring node may directly probe all instances by their **management address**...



...but then it's likely not exercising the same paths as actual users of the service.

Another compromise

Many monitors (RUM, RIPE Atlas, etc), well distributed in the topology may succeed in probing all service instances, but nondeterministically.



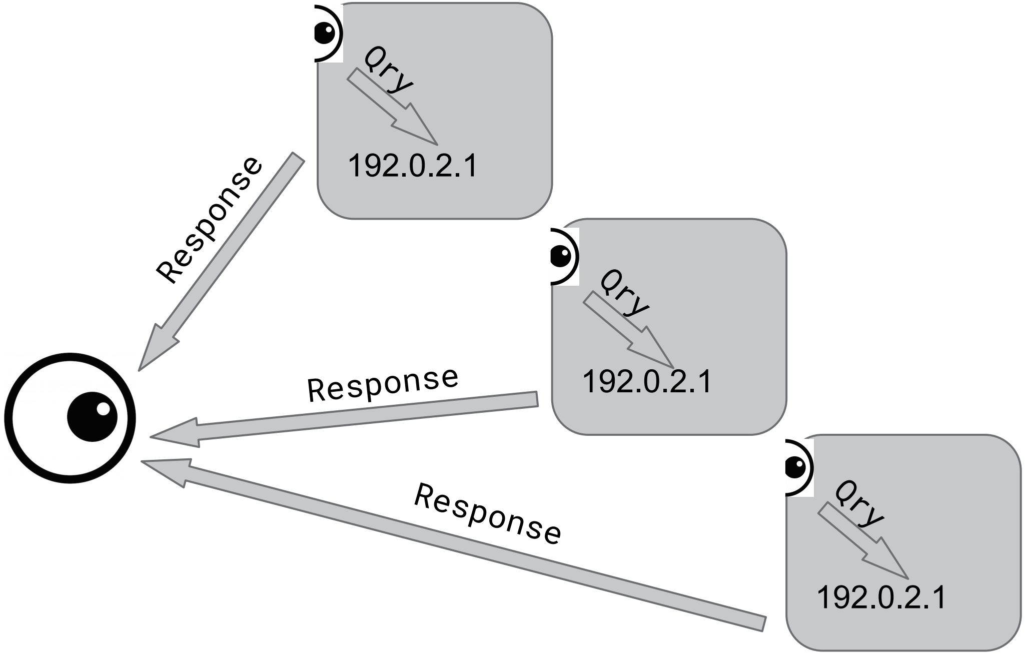
An instance of an anycast service with constrained route propagation may remain invisible to all but the most widely distributed probes.

A new compromise

If we **generate a query local to the anycast service** instance, we can probe it directly.

If we **spoof the source address** of that query we can direct the response to our single monitoring node.

We can probe all instances of anycast service deterministically and **gather responses at one node**.



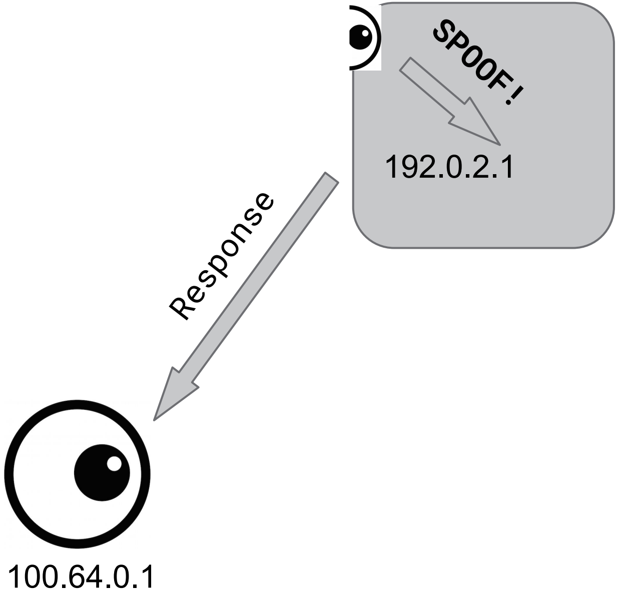
Spoofing!

This sounds more exciting than it actually is.

Spoofing takes place inside the server and results in a completely unsurprising packet on the wire:

```
192.0.2.1 : 53 => 100.64.0.1 : 54321
```

No violation of the provisions of BCP38, or MANRS, etc is being perpetrated here.



Spoofing a query

DNS Message

```
1463295169321.dyndns.com IN SOA ? +NSID
```

Encode current time in ms.

Set NSID option so we can tell where the query was answered

UDP

```
src: 53  dst: 4653
```

Collector listens on port 4653

IP

```
dst: 192.0.2.1   src: 100.64.0.1  ttl: 1
```

Guard against locally unanswerable queries confusingly going elsewhere with IP TTL=1

Implemented in Perl because Net::RawIP and Net::DNS are easy to use

Deconstructing a response

```
;; OPT PSEUDOSECTION:
; NSID: hivecast-11-usiad.as15135.net
;; QUESTION SECTION:
;1463406752123.dyndns.com. IN  SOA
;; AUTHORITY SECTION:
dyndns.com. 0 IN SOA ns0.. host.. 2016051200 ..
```

Collector implemented in Ruby, writes metrics via Collectd into Graphite.

IP source address tells us which anycast service was tested.

NSID tells us which node answered the query.

QNAME tells us when the query was generated.

What use is this?

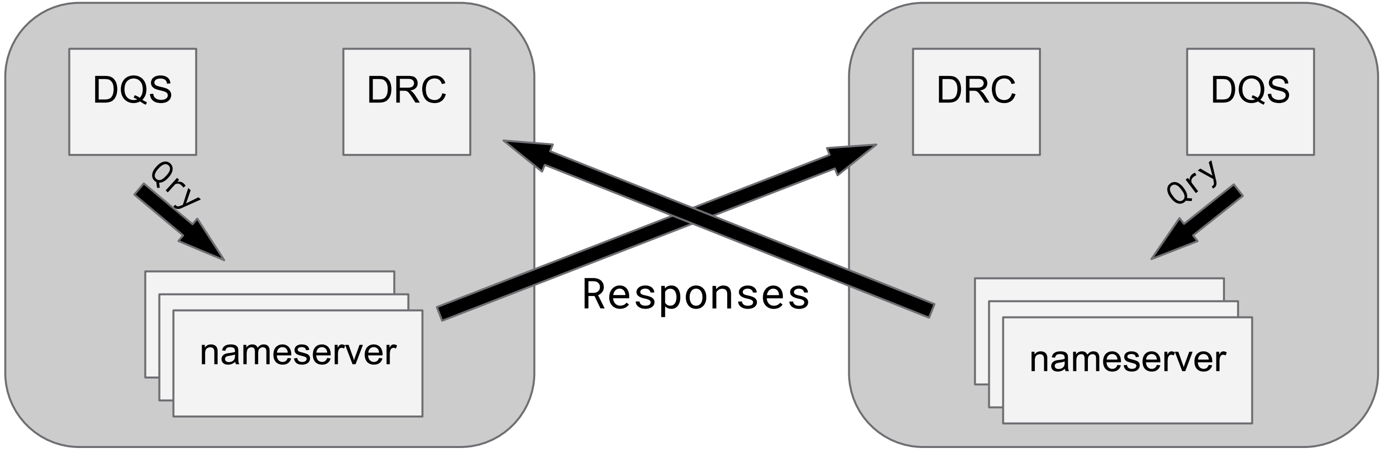
We have a heartbeat!

We can watch for changes in the SOA serial.

Subtract the query generation time from the current time and we get the **single trip time** for the response to get from the anycast instance to the monitor node.

This assumes excellent clock synchronisation. This can otherwise still be useful in detecting aberrant behaviour if the clocks are at least consistently dyssynchronous.

Scaling up



DNS
Query
Spoofer

DNS
Response
Collector

Probe all of the nameservers on a node

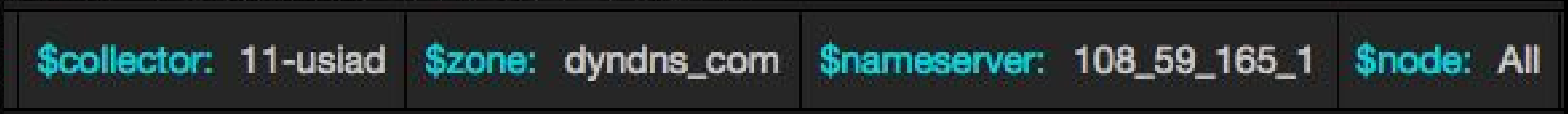
Send responses to collectors running on other nodes.

Build a full mesh of single trip latencies.

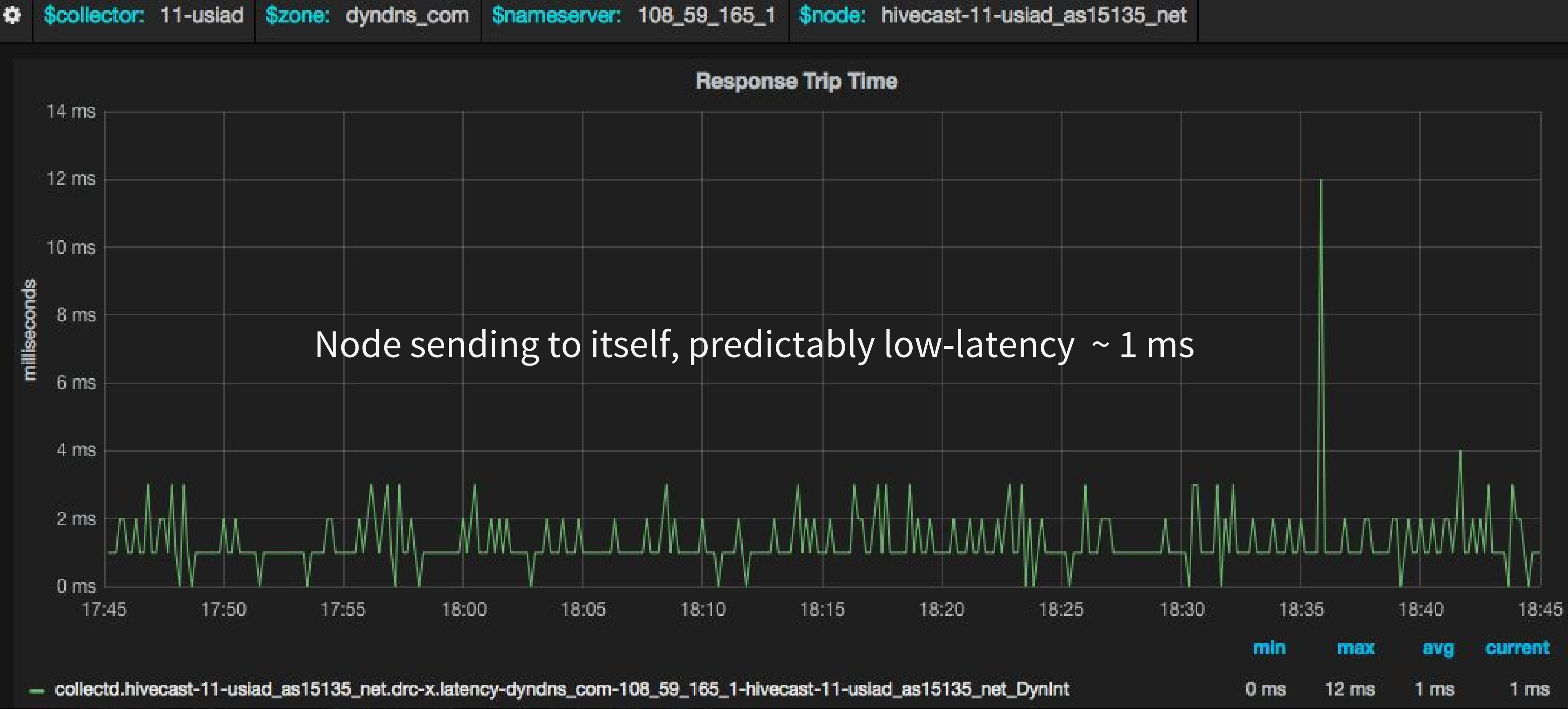
Graphs!

Metrics sent to Collectd are viewable in a Grafana dashboard with templated queries

[**collector**].drc-x.latency-[**zone**]-[**nameserver**]-[**node**]-[container]
= single trip time in milliseconds



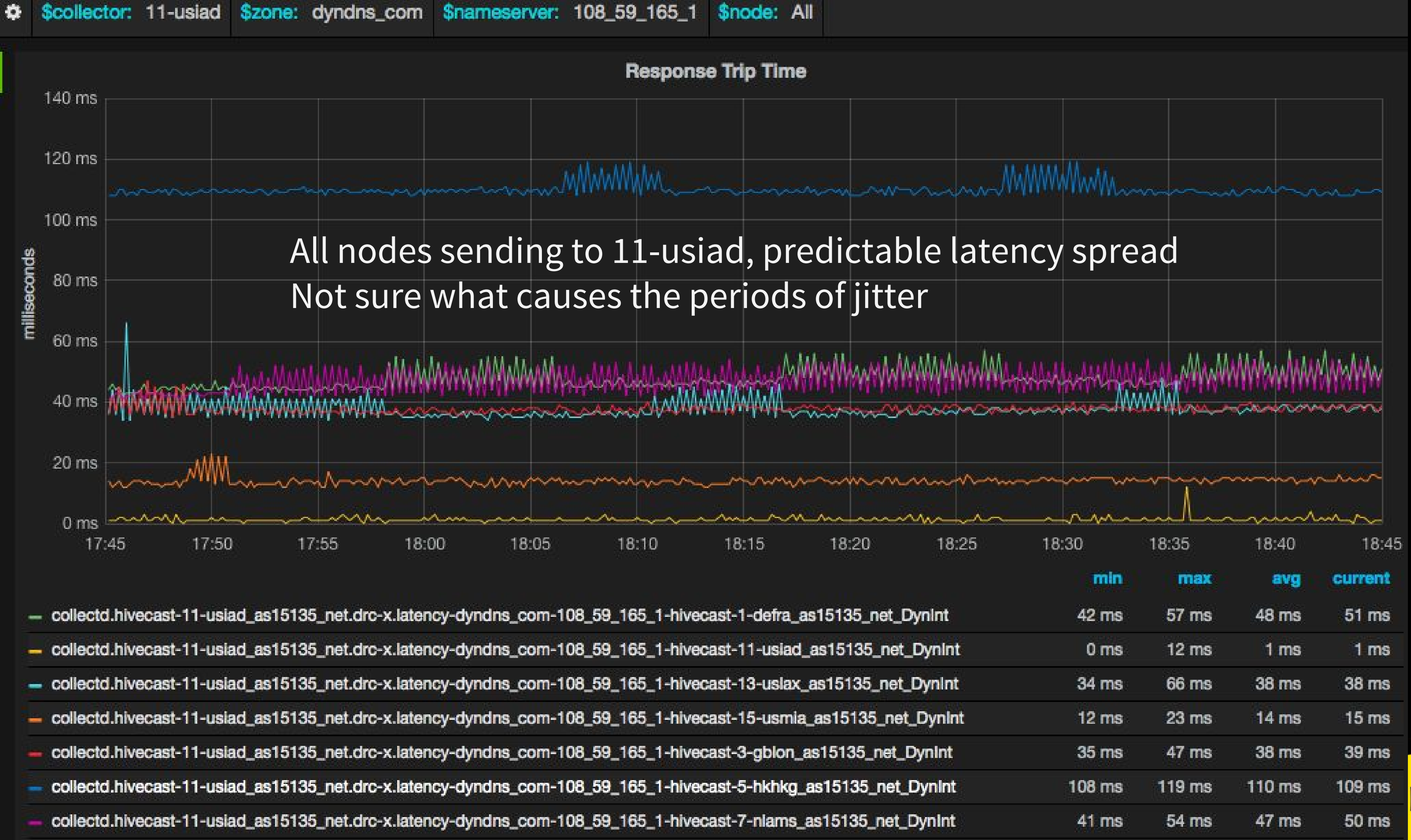
$collector: 11-usiad
$zone: dyndns_com
$nameserver: 108_59_165_1
$node: All



$collector: 11-usiad $zone: dyndns_com $nameserver: 108_59_165_1 $node: hivecast-11-usiad_as15135_net

Node sending to itself, predictably low-latency ~ 1 ms

	min	max	avg	current
collectd.hivecast-11-usiad_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-11-usiad_as15135_net_DynInt	0 ms	12 ms	1 ms	1 ms



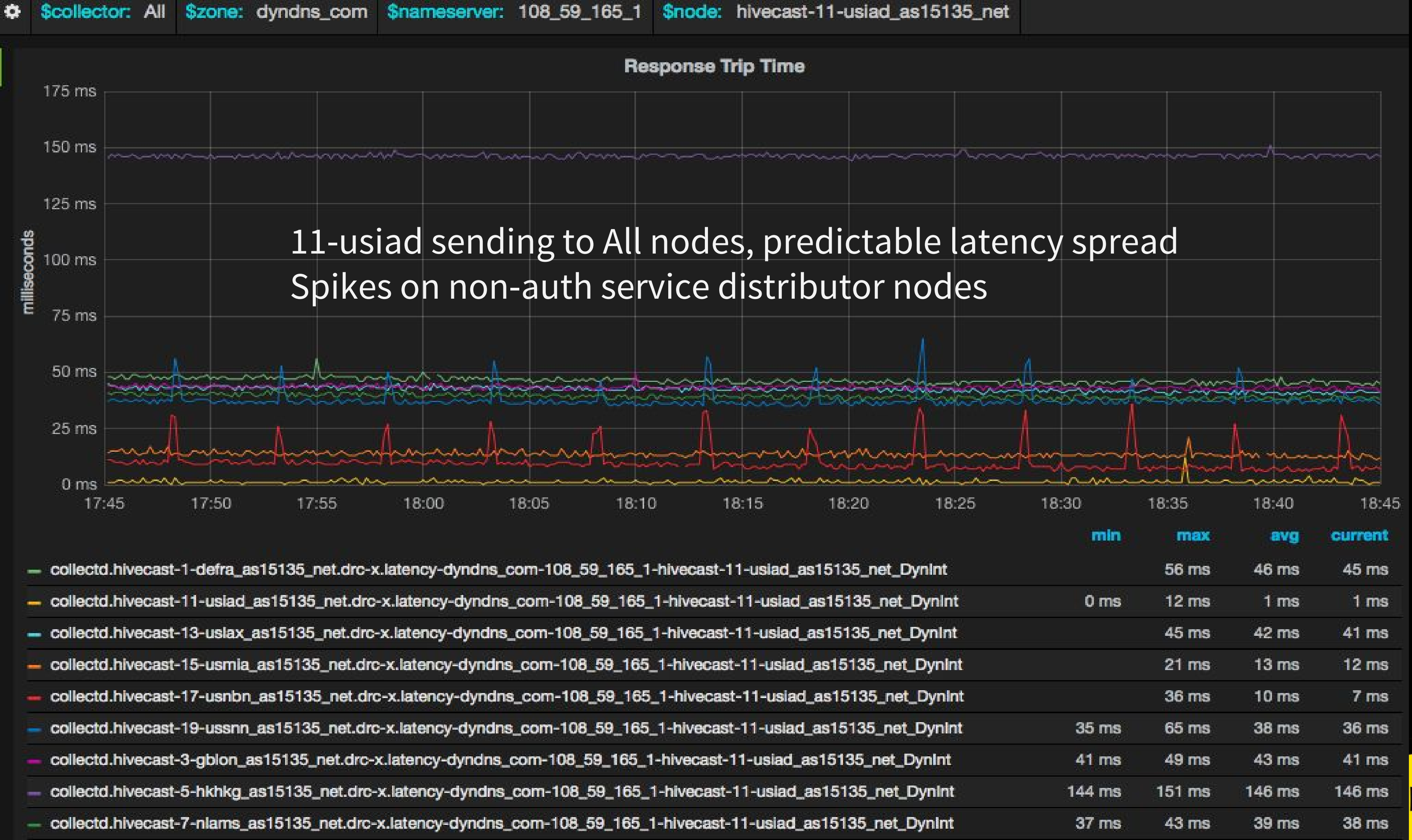
$collector: 11-usiad $zone: dyndns_com $nameserver: 108_59_165_1 $node: All

All nodes sending to 11-usiad, predictable latency spread

Not sure what causes the periods of jitter

	min	max	avg	current
collectd.hivecast-11-usiad_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-1-defra_as15135_net_Dynlnt	42 ms	57 ms	48 ms	51 ms
collectd.hivecast-11-usiad_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-11-usiad_as15135_net_Dynlnt	0 ms	12 ms	1 ms	1 ms
collectd.hivecast-11-usiad_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-13-uslax_as15135_net_Dynlnt	34 ms	66 ms	38 ms	38 ms
collectd.hivecast-11-usiad_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-15-usmia_as15135_net_Dynlnt	12 ms	23 ms	14 ms	15 ms
collectd.hivecast-11-usiad_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-3-gblon_as15135_net_Dynlnt	35 ms	47 ms	38 ms	39 ms
collectd.hivecast-11-usiad_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-5-hkhkg_as15135_net_Dynlnt	108 ms	119 ms	110 ms	109 ms
collectd.hivecast-11-usiad_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-7-nlams_as15135_net_Dynlnt	41 ms	54 ms	47 ms	50 ms

$collector: All $zone: dyndns_com $nameserver: 108_59_165_1 $node: hivecast-11-usiad_as15135_net



11-usiad sending to All nodes, predictable latency spread

Spikes on non-auth service distributor nodes

	min	max	avg	current
🟩 collectd.hivecast-1-defra_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-11-usiad_as15135_net_DynInt		56 ms	46 ms	45 ms
🟨 collectd.hivecast-11-usiad_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-11-usiad_as15135_net_DynInt	0 ms	12 ms	1 ms	1 ms
🟦 collectd.hivecast-13-uslax_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-11-usiad_as15135_net_DynInt		45 ms	42 ms	41 ms
🟧 collectd.hivecast-15-usmia_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-11-usiad_as15135_net_DynInt		21 ms	13 ms	12 ms
🟥 collectd.hivecast-17-usnbn_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-11-usiad_as15135_net_DynInt		36 ms	10 ms	7 ms
🟦 collectd.hivecast-19-ussnn_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-11-usiad_as15135_net_DynInt	35 ms	65 ms	38 ms	36 ms
🟪 collectd.hivecast-3-gblon_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-11-usiad_as15135_net_DynInt	41 ms	49 ms	43 ms	41 ms
🟪 collectd.hivecast-5-hkhkg_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-11-usiad_as15135_net_DynInt	144 ms	151 ms	146 ms	146 ms
🟩 collectd.hivecast-7-nlams_as15135_net.drc-x.latency-dyndns_com-108_59_165_1-hivecast-11-usiad_as15135_net_DynInt	37 ms	43 ms	39 ms	38 ms

Closing thoughts

Limitations

Only useful for UDP

Currently only IPv4 is implemented

No authentication

Further work

Compare with traditional measurements

Address known limitations

Publish the tools

Further explore the observations

Advantages

A new tool in the box

Auto discovery, monitors don’t need to know of anycast instances in advance

Probing can scale horizontally (though maybe not with a full mesh)

No state means no timeouts, this may reveal previously hidden weirdness

Can measure latency in a single direction

QUESTIONS?

dknight@dyn.com