RootViz: visualizing real-time monitoring of Root and TLD servers

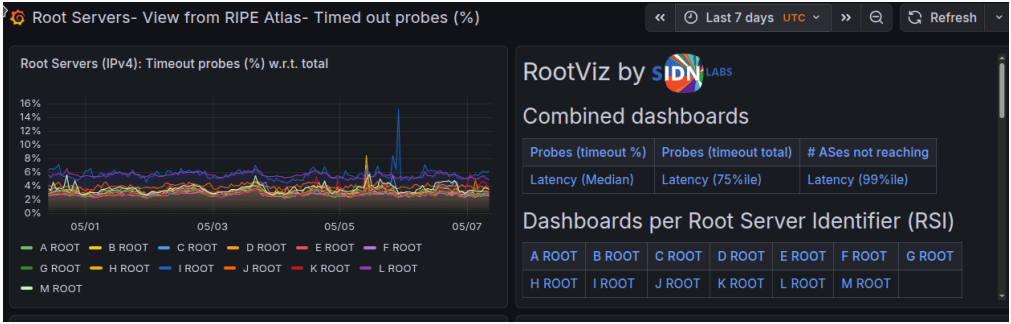
Giovane C. M. Moura
SIDN Labs and TU Delft

OARC 46
Edinburgh, Scotland
2026-05-17

Outline

RootViz dashboard

https://rootviz.sidnlabs.nl



The Root Server System

13 Root Server Identifier (RSIs)

`[a-m].root-servers.net`

- **12 independent operators**

  (Verisign runs both A and J)
- Self-funded, voluntary
- Each operator: **full autonomy**

RSI	Operator	
A		Verisign
B		USC-ISI
C		Cogent
D		U. Maryland
E		NASA OCIO
F		ISC
G		US DoD (DISA)
H		US Army (ARL)
I		Netnod
J		Verisign
K		RIPE NCC
L		ICANN
M		WIDE Project

Why monitor the Root?

The **Root DNS** and TLDs are the foundation of the DNS.

- Every uncached lookup eventually depends on them
- If the Root breaks ⇒ **the entire DNS suffers**, globally

You can't measure global reachability from *inside* the system.

You need external vantage points.

External VPs probing Root servers (RIPE Atlas Probes)

https://atlas.ripe.net

What are RIPE Atlas probes?

The largest active Internet measurement network.

Probes are diverse: hardware, VMs, rack servers

- Run by **RIPE NCC** since 2010
- **∼14K probes** in 180 countries
- Continuously measure DNS, ping, traceroute
- Data is **public & free**

A RIPE Atlas (v3) hardware probe

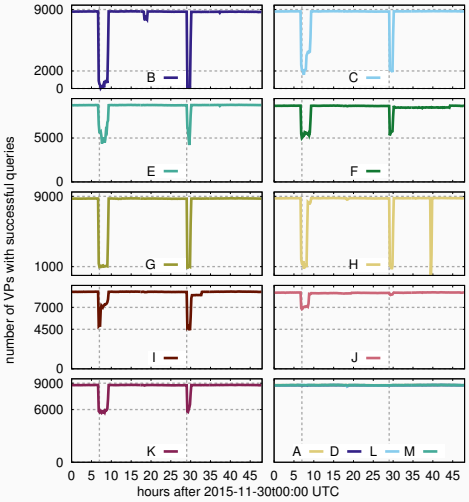
Context

Ten years ago, we published:

> **Anycast vs. DDoS: Evaluating the November 2015 Root DNS Event**
>
> https://dl.acm.org/doi/10.1145/2987443.2987446

RootViz uses the same data sources — now visualized in real time.

- RIPE Atlas probes (All of them)

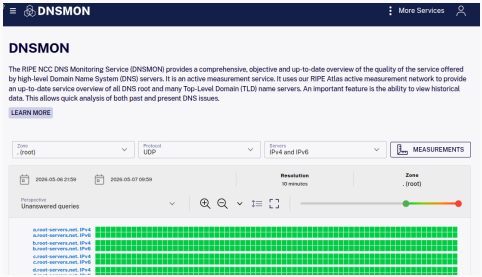


Motivation

The data is already there — but it's *dark data*.

- RIPE Atlas continuously measures Root servers from **all Atlas probes** (∼14K worldwide)
- Most operators only look at **DNSMON**
- But DNSMON uses only **anchors**

Why not use *all* of them? and have more metrics? and Grafana?



DNSMON — anchors only

https://dnsmon.ripe.net/

Anchors vs. probes: trade-offs

Anchors

datacenters · controlled

✓ Pros

- Datacenter-grade, clean networks
- Highly reliable individually

× Cons

- Few (∼1K), concentrated
- Miss the “messy” real Internet

Rest of probes

homes, offices · messy reality

✓ Pros

- Many (∼13K), wide coverage
- See the *real* user experience

× Cons

- Local quirks, DNS hijacking[1]
- Islands/peninsulas[2]
- Single probe = unreliable

[1] Intercept and Inject (Nosyk et al. (2023)
[2] Saluja et al.(2022)

Anchors vs. all probes: how much do we gain?

	Anchors (DNSMON)	All probes (RootViz)	Gain
Vantage points	999	14,528	**14.5×**
ASNs (IPv4)	743	4,505	**6.1×**
ASNs (IPv6)	681	2,304	**3.4×**
Prefixes (IPv4)	964	9,911	**10.3×**
Prefixes (IPv6)	858	3,496	**4.1×**
Countries	109	180	**1.7×**

Source: RIPE Atlas probe file, 2026-05-06

Anchors vs. all probes: how much do we gain?

	Anchors (DNSMON)	All probes (RootViz)	Gain
Vantage points	999	14,528	**14.5×**
ASNs (IPv4)	743	4,505	**6.1×**
ASNs (IPv6)	681	2,304	**3.4×**
Prefixes (IPv4)	964	9,911	**10.3×**
Prefixes (IPv6)	858	3,496	**4.1×**
Countries	109	180	**1.7×**

Source: RIPE Atlas probe file, 2026-05-06

Using all probes: *a wider view of reachability.*

Our approach: crowdsourcing reachability

Same idea as those airport restroom **HappyOrNot** terminals[1]:

- One press is noise

If many probes agree something's wrong — something *is* wrong.

HappyOrNot dashboard

[1] D. Owen, "The Happiness Button," *The New Yorker*, Feb. 8, 2018

RootViz: goals

Light up dark data

Atlas measures all RSIs from ∼14 K probes — most results are never seen.

Real-time monitoring

30-min aggregates, live Grafana dashboard.

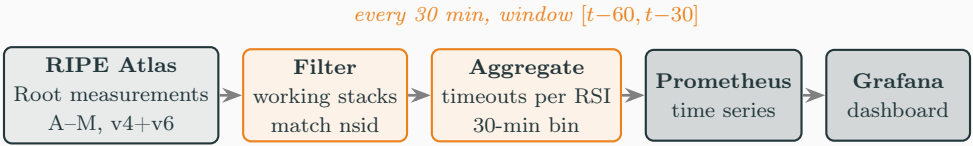
Crowdsourced reachability

All probes, not just anchors — **14.5**× more vantage points.

Not a blame tool. Failures can be at the probe, in-network, or at the server location — not necessarily **the root server operator**.

Outline

RootViz pipeline



Input

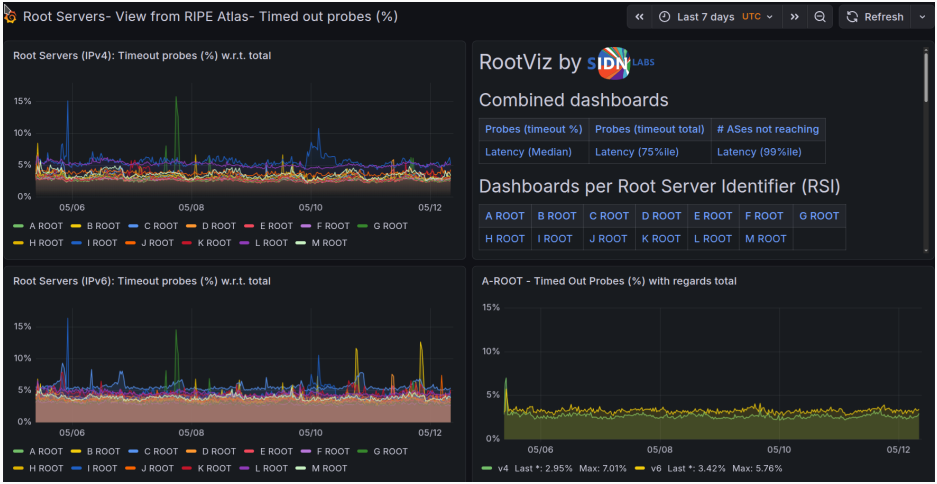
- `chaos TXT hostname.bind`
- Sent direct to each RSI (no resolver)
- All probes, IPv4 & IPv6

Reduction

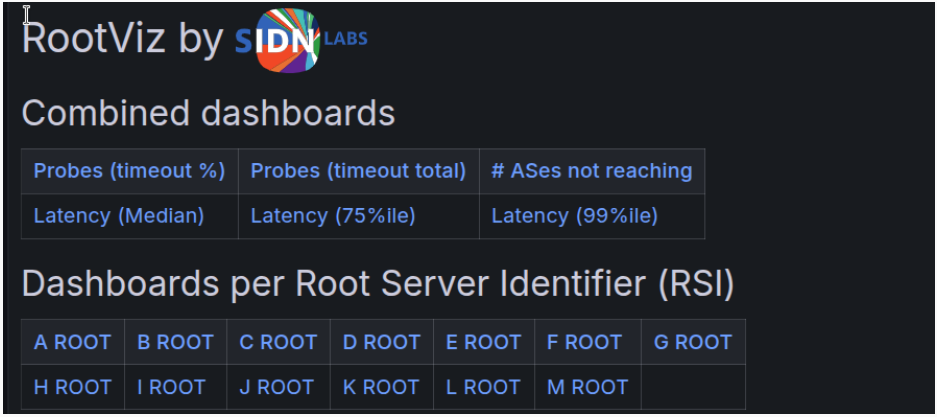
- Keep probes with working stack `system-ipv{4,6}-works`
- Drop responses with mismatched `nsid`
  - queries being intercepted
- Count unique `prb_id` timeouts within 30min

Outline

Dashboards

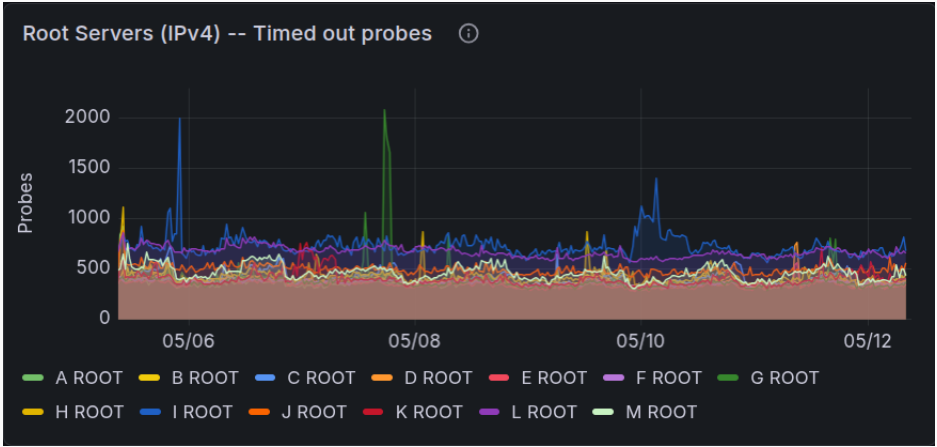

Root Servers- View from RIPE Atlas- Timed out probes (%)
Last 7 days UTC
Refresh
Root Servers (IPv4): Timeout probes (%) w.r.t. total
15%
10%
5%
0%
05/06
05/08
05/10
05/12
A ROOT B ROOT C ROOT D ROOT E ROOT F ROOT G ROOT
H ROOT I ROOT J ROOT K ROOT L ROOT M ROOT
RootViz by SIDN LABS
Combined dashboards
Probes (timeout %)
Probes (timeout total)
ASes not reaching
Latency (Median)
Latency (75%ile)
Latency (99%ile)
Dashboards per Root Server Identifier (RSI)
A ROOT B ROOT C ROOT D ROOT E ROOT F ROOT G ROOT
H ROOT I ROOT J ROOT K ROOT L ROOT M ROOT
Root Servers (IPv6): Timeout probes (%) w.r.t. total
A-ROOT - Timed Out Probes (%) with regards total
v4 Last *: 2.95% Max: 7.01%
v6 Last *: 3.42% Max: 5.76%

Metrics



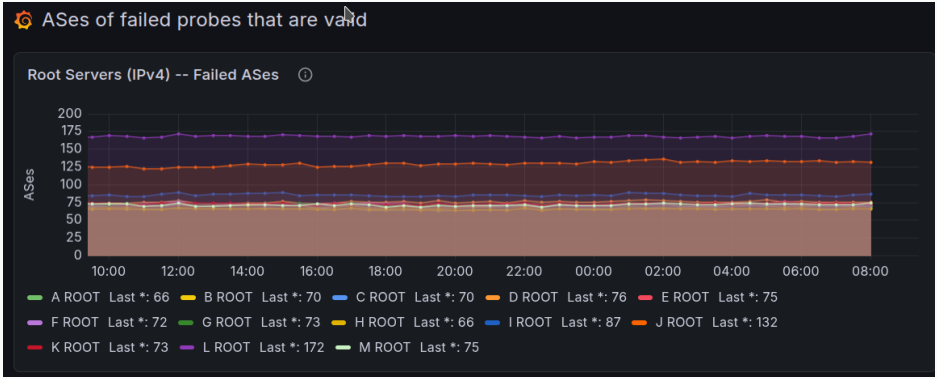
For Roots: timeout (reachability) is key

Timeout probes dashboard



15 graphs: combined and separated...

ASes of timedout probes



No root cause, only smoke detector...

RootViz — what we learned by looking

The data was always there.

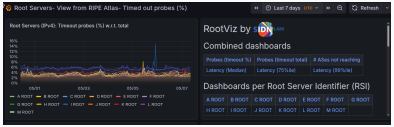
RootViz raises many questions:

- RSIs have **different baseline timeout levels** — why?

Every answer raises more questions.

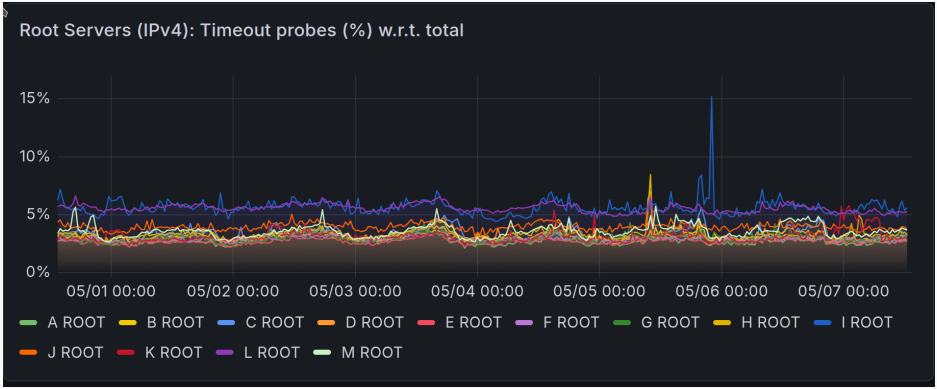
Come look with us.

https://rootviz.sidnlabs.nl



Outline

Finding 1: baseline timeouts differ per RSI



Each RSI has its **own steady-state** timeout level — not zero, and not equal.

Finding 2: stacks fail independently

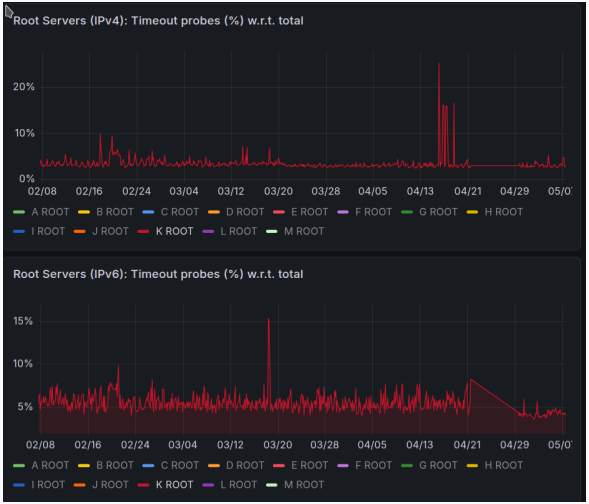
K-Root, April 26

IPv4 only — IPv6 unaffected

We reached out to **K-Root operators**.

> Issue with a **transit provider** connecting some Atlas probes to K-Root.

⇒ v4 and v6 paths fail independently.

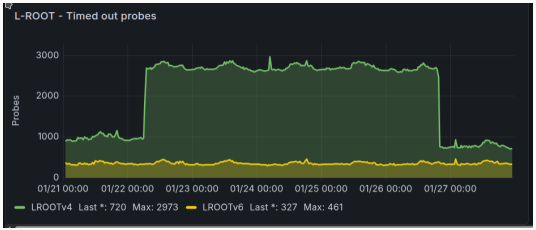


Finding 3: a starting point for deeper analysis

RootViz **flags** when something is off — but **not why**.
Once an event is detected, you can drill down:

- Which **probes** timed out?
- Which **anycast site** were they catching before?
- Which **ASNs / countries** are affected?

RootViz is the **smoke detector** — the investigation comes next.



LROOT Jan 2026 event

Outline

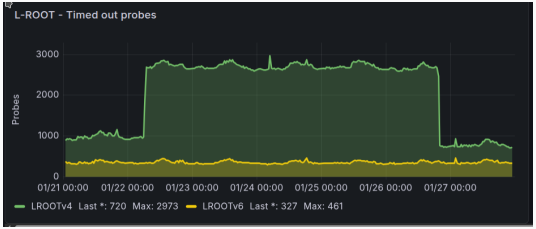
Case study: L-Root, January 2026

2026-01-22, ∼06:00 UTC

lasted until 01-26 20:00 UTC

- **∼2,000 Atlas probes** suddenly time out
- Reaching **L-Root over IPv4**
- IPv6 **unaffected**
- Other RSIs **unaffected**

Sharp spike, then back — not noise.

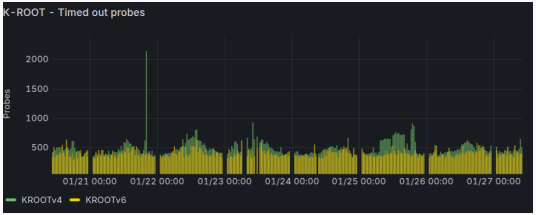


L-Root, IPv4 timeouts, Jan 21–27

“But probes are noisy — nothing to see here”

If it were just probe noise, *all* RSIs would spike together.
They didn’t.

- Same probes, same time window
- K-Root: **stable**
- Other RSIs: **stable**
- Probes weren’t offline — **timeouts only**
- IPv6 worked fine

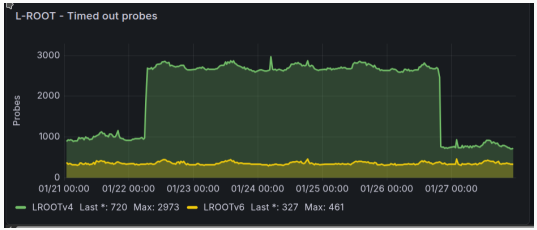


K-Root, same time window — flat

Drill down: who are the timing-out probes?

Compare two 10-min windows of the same measurement (https://atlas.ripe.net/measurements/10308):

- **Baseline**: Jan 22nd, 2026: 04:00–04:10 UTC (10min)
- **Event**: Jan 22nd, 2026: 13:00–13:10 UTC (10min)

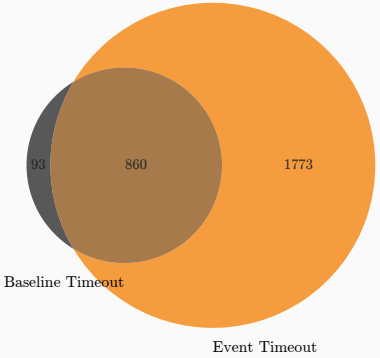


L-Root, IPv4 timeouts, Jan 21–27

Drill down: event window

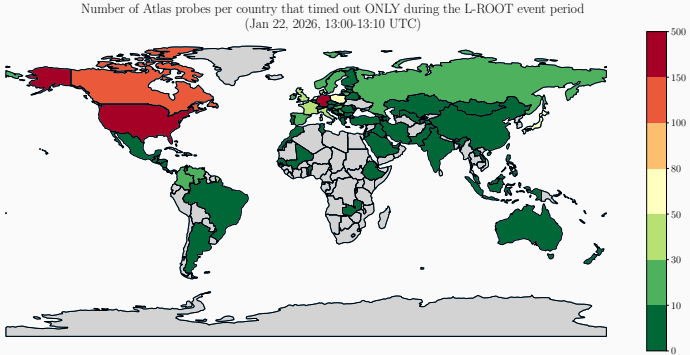
Same measurement, 9 hours later:

- 1,773 probes timed out only during the event



Probes timing out: baseline

Where are these probes? (countries)



Country	Probes
Germany	464
United States	348
Canada	119
Japan	75

Spread across **97 countries** — not a localized issue.

Which networks? (top 10 ASNs)

ASN	Network	Timed out	Reached OK	Ratio OK
3320	Deutsche Telekom (DTAG)	353	3	0.8%
33915	Vodafone Libertel (NL)	53	15	22.1%
396982	Google Cloud Platform	40	39	49.4%
16509	Amazon AWS	37	35	48.6%
36352	ColoCrossing	37	5	11.9%
14593	SpaceX Starlink	35	63	64.3%
3292	TDC (DK)	23	0	0.0%
5617	Orange Polska	23	0	0.0%
31898	Oracle Cloud	22	159	87.8%

Server side: which L-Root sites were they hitting?

Look at the affected probes' **anycast catchments** from the baseline:

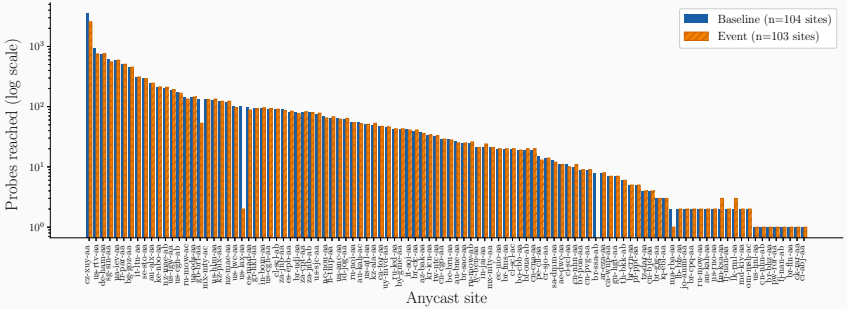
L-Root site	Probes
`cz-xuy-aa` (Czechia)	962
`us-rtv-aa` (US)	390
`sg-sin-aa` (Singapore)	127
`us-lax-aa` (US)	94
`gb-brf-aa` (UK)	80

Most affected probes used to hit **Czechia** or **US** sites.

Confirmed by L-Root operators: an **upstream provider issue** affecting those sites.

⇒ Crowdsourced detection → real-world root cause.

Were these sites down? (Atlas sees 103 of 140)



Biggest ratio drops:

Site	Base	Event	Ratio
`br-ssa-ab`	8	0	0.0
`us-lax-aa`	101	2	0.0
`gb-brf-aa`	135	53	0.4

Sites stayed up — valid answers throughout.

Failure was upstream — not at the site.

If L-root was up, who's to blame?

- Anycast sites stayed up: probes that *did* reach L-root got valid answers.
- So the failures weren't on L-root's side — the queries never arrived.
- Something **between the probe and the anycast site** dropped them.

So where did the packets die?

→ Let's look at the **traceroute data** (well, see extra slides).

Next: per-probe path comparison · baseline vs. event

Recap

RootViz works for eyeballing.

- Real-time view across **all** RIPE Atlas probes
- Already **confirmed events** with Root operators
- Points to **where** to look: country, ASN, anycast site, v4/v6

Step	Done by
Spike detected	RootViz
IPv4 only?	RootViz
Which probes?	Manual
Which anycast sites?	Manual
BGP state check	Manual
Transit AS scan	Manual
Data plane failure	Manual

But the L-Root case shows the limit:

Detection was easy. Root cause took **hours of manual work**.

Ongoing work: two student groups @ TU Delft

Group 1

Event detection &
automated reporting

- Automatic anomaly detection
- Trigger & generate reports
- Reduce manual eyeballing

Group 2

Per-RSI deep dive

- Each member: **2 root letters**
- Some show periodic timeouts
- Others don't — why?

Builds on past student work like **NTPINFO**[1] — now a public service.

[1] ntpinfo.sidnlabs.nl

Roadmap: RootViz v2

- **Full automation** of event detection
- **Event labeling** & classification
- **Public datasets** for the community

From dashboard to a measurement service.

Try it

https://rootviz.sidnlabs.nl

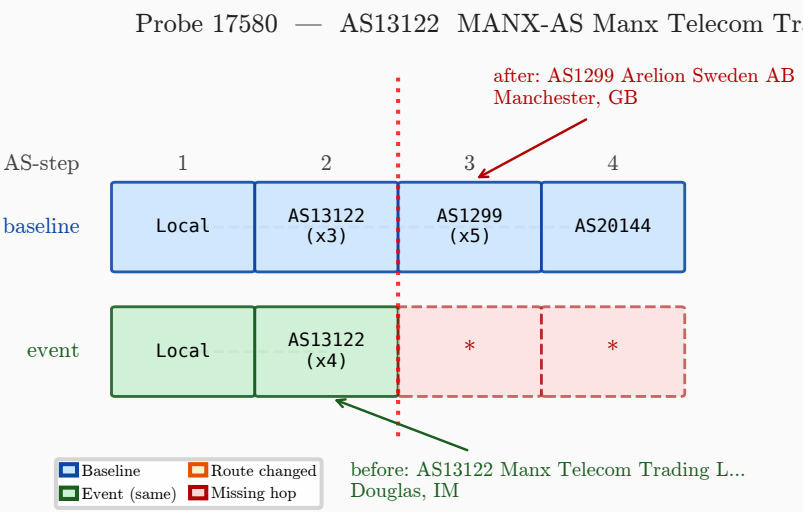
Questions?

giovane.moura@sidn.nl

❤ Thanks to RIPE NCC for Atlas data
and feedback from Root Operators
(A,B,G,J,K,L so far)

Outline

Case study: Probe 17580 (AS13122)



Step 1: Who are AS13122's upstreams?

```
requests.get("https://stat.ripe.net/data/asn-neighbours/data.
    json", params={"resource": "AS13122"})
```

Top 5 upstreams (BGP power = how often this AS appears on AS13122's left in routing tables):

ASN	Power	Name
AS3356	610	Lumen (Level3)
AS1299	391	Arelion
AS6939	213	Hurricane Electric
AS9002	75	RETN
AS13030	21	Init7

This data is **not time-bound** — RIPEstat returns the long-term neighbor view (here: 2019-03 to 2026-05). It tells us *who could carry traffic for AS13122*, not used one.

Step 2: Did the BGP state for the prefix change?

```
requests.get("https://stat.ripe.net/data/bgp-state/data.json",
    params={"resource":  "199.7.83.0/24",
            "timestamp": "2026-01-22T04:00:00"})
```

BGP state = every AS path RIPE's collectors saw being announced for the prefix at that moment.

Metric	Baseline	Event
Total paths to prefix	423	423

Same *count* could hide different *content* — we check that in Steps 3 & 4.

Step 3: Did *any* transit AS shift paths?

Now we widen the check to **every other transit AS** that appears in any path to `199.7.83.0/24`.

Transit AS	Base	Event	Δ
AS2914 NTT	6	6	0
AS33891 Core-Backbone	5	5	0
AS3257 GTT	3	3	0
AS25091 IP-Max	3	3	0
AS8298 IPng Networks	2	2	0
All others	—	—	**0**

Reading the table. Each number is how many of the 423 BGP paths to L-root pass through that AS. *Base* = 04:00, *Event* = 13:00.

Verdict. 0 transit ASes flagged. Every AS carries L-root traffic for the same set of networks before and during the event.

Where we stand

What we ruled out:

- L-root anycast sites — they stayed up, served queries fine.
- BGP withdrawals — 423 → 423 paths, no change.
- AS13122's upstreams — all 5 retained the route.
- Other transit ASes — zero path shifts globally.
- Verizon route leak — not present in any path.

What's left:

- The control plane seems healthy.
- Failure should live on the **data plane**.

The route existed — it just wasn't forwarded.

Zooming out: 419 failed probes

We applied the same baseline-vs-event analysis to **every probe that failed**. For each, we recorded:

- The probe's home AS
- The **last AS** where its traceroute responded
- The **first AS** that was *supposed* to be next (from baseline)

That gives us a map of **where the network broke**, repeated 419 times.

Where did traceroutes die?

Last AS	Probes affected	Share
AS3320 Deutsche Telekom	124	30%
Local (own network)	46	11%
AS1299 Arelion	46	11%
AS? (unresolved router)	28	7%
AS3257 GTT	16	4%

Striking pattern: 30% of all failed probes stop responding *inside Deutsche Telekom* — before traffic even exits their network.

What was the next AS *supposed* to be?

Expected next AS	Probes	Share
AS1299 Arelion	181	43%
AS20144 ICANN (L-root)	106	25%
AS3257 GTT	93	22%
AS174 Cogent	13	3%
AS33920 Cologix	6	1%

Three ASes account for **90% of the missing handoffs**: Arelion, ICANN itself, and GTT.

The dominant failure paths

Last seen	Expected next	Probes
AS3320 DTAG	AS3257 GTT	67
AS3320 DTAG	AS1299 Arelion	57
AS1299 Arelion	AS20144 ICANN	45
Local	AS1299 Arelion	25
AS3257 GTT	AS20144 ICANN	16

Three interconnection points caused most failures:

- **DTAG** ↔ **GTT** and **DTAG** ↔ **Arelion** (124 probes)
- **Arelion** ↔ **ICANN** (45 probes)

What the aggregate tells us

- Failures **cluster on a few interconnection points**, not spread randomly.
- **Deutsche Telekom probes** were hit hardest — 30% of failures originated there.
- Three transit handoffs caused 90% of the breaks: **DTAG↔GTT**, **DTAG↔Arelion**, **Arelion↔ICANN**.
- BGP showed everything was healthy at all these ASes.

A localized data-plane failure
at a handful of border routers.