Measuring the Effectiveness of Route Origin Validation Filtering via Drop Invalids from the perspective of the End User using a Technique of Broad Scale Reachability Measurement

Geoff Huston
APNIC Labs

Measuring RPKI

Geoff Huston
APNIC

Route Origin Validation Filtering!

Measuring ~~RPKI~~

Geoff Huston
APNIC

Routing Security

What's "the objective" of routing security?

Routing Security

What's "the objective" of routing security?

- ❑Protect the routing system from all forms of operator mishaps?
- ❑Protect the routing system from some forms of operator mishaps?
- ❑Protect the routing system from all hostile attacks?
- ❑Protect the routing system from some hostile attacks?
- ❑Prevent the routing of bogus address prefixes?
- ❑Prevent the use of bogus AS's in the routing system?
- ❑Prevent all forms of synthetic routes from being injected into the routing system?
- ❑Prevent unauthorised route withdrawal?
- ❑Protect users from being directed along bogus routing paths?

Let's not be too ambitious!

Enforcing rules to ensure that the routes carried in BGP are both protocol-wise accurate and policy-wise accurate is well beyond the capabilities of BGP and viable BGP control mechanisms *

Route Origin Validation is designed to prevent BGP speakers from learning and preferring routes that are not authorised by the prefix holder

The intent of not preferring unauthorised routes is to prevent users' traffic from being steered along these bogus routes

* BGP is not a deterministic protocol, but more of a negotiation protocol that attempts to find meta-stable 'solutions to importer / export policy preferences simultaneously. Where the policies are incompatible the BGP "solution" is not necessarily reached deterministically and different outcomes will be seen at different times – see "BGP Wedgies" for an illustration of this form of indeterminism

Routing Security

What's "the objective" of routing security?

- ❑ Protect the routing system from all forms of operator mishaps?
- ❑ Protect the routing system from some forms of operator mishaps?
- ❑ Protect the routing system from all hostile attacks?
- ❑ Protect the routing system from some hostile attacks?
- ❑ Prevent the routing of bogus address prefixes?
- ❑ Prevent the use of bogus AS's in the routing system?
- ❑ Prevent all forms of synthetic routes from being injected into the routing system?
- ❑ Prevent unauthorised route withdrawal?
- ☑ Protect users from being directed along bogus routing paths?

Our Objective

- To measure the “impact” of invalid route filtering on users

- The question we want to answer here is user-centric:
  - **What proportion of users can’t reach a destination when the destination route is invalid according to ROV?**

- We’d like to continue this as a long term whole-of-Internet measurement to track the increasing deployment of RoV filtering* over the coming months and years

* “RoV filtering” is shorthand for “using RPKI validation of published Route Origination Attestations to detect and drop route objects that are invalid according to the conventional RoA interpretation”, which in practice means either the prefix is too specific (shorted than MaxLength) or has an origin AS which is not contained in a valid ROA

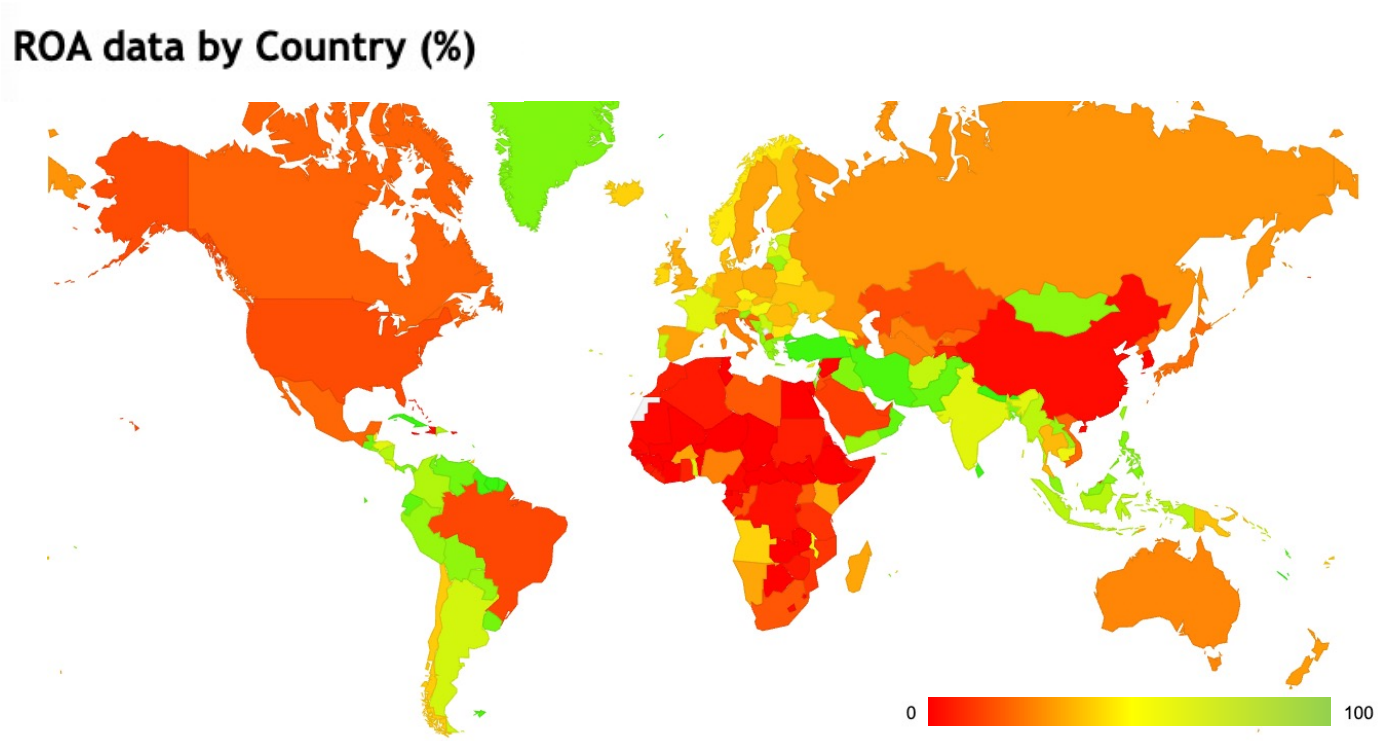
Production vs Consumption

There are two aspects to this framework:

- Generating ROAs to describe the intended origination of prefixes
- Looking for those networks that will admit and propagate invalid routes
  - i.e.: those networks that are not performing some for of “drop invalid” filtering on BGP advertisements

Production

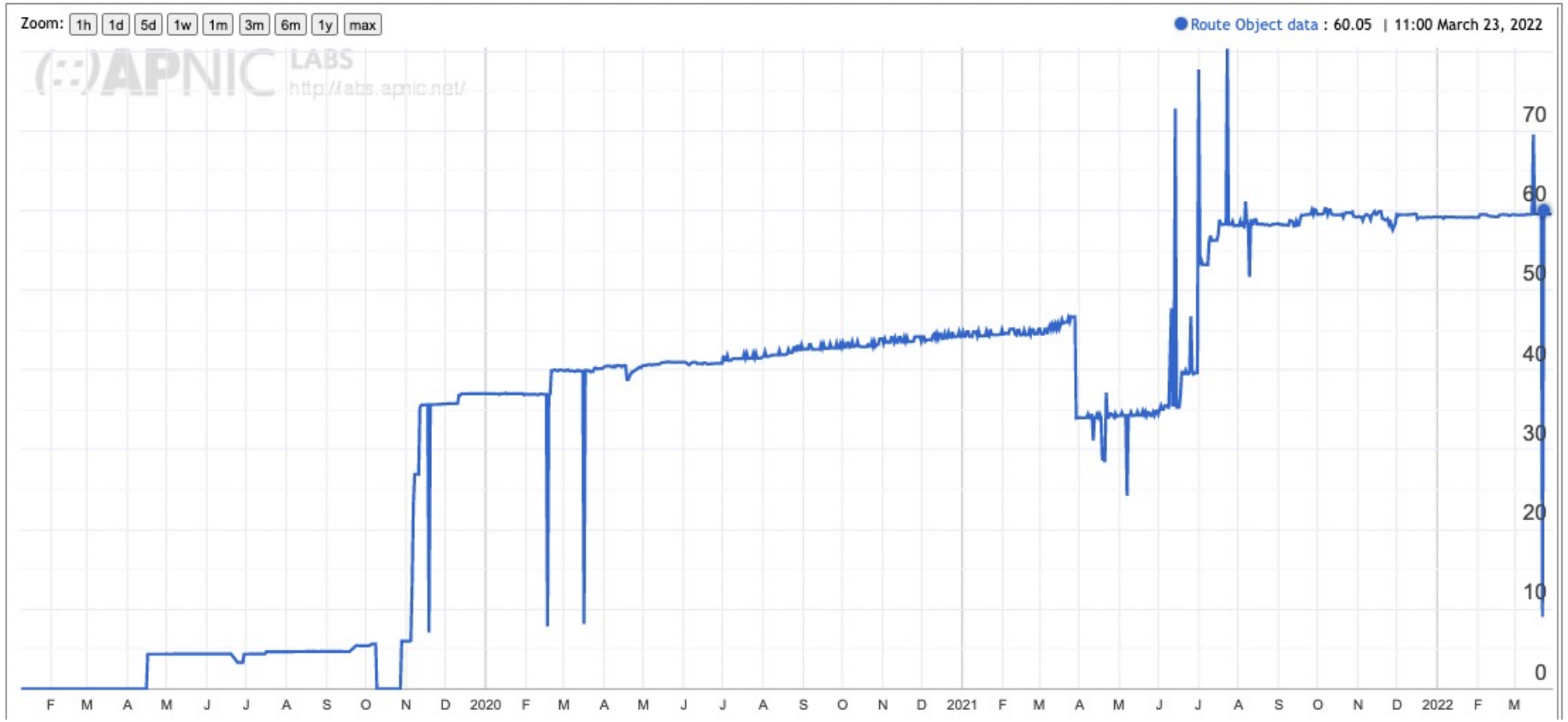
Which national operator communities have been generating ROAs for their announced prefixes?



https://stats.labs.apnic.net/roas

ROAs for Australian Networks

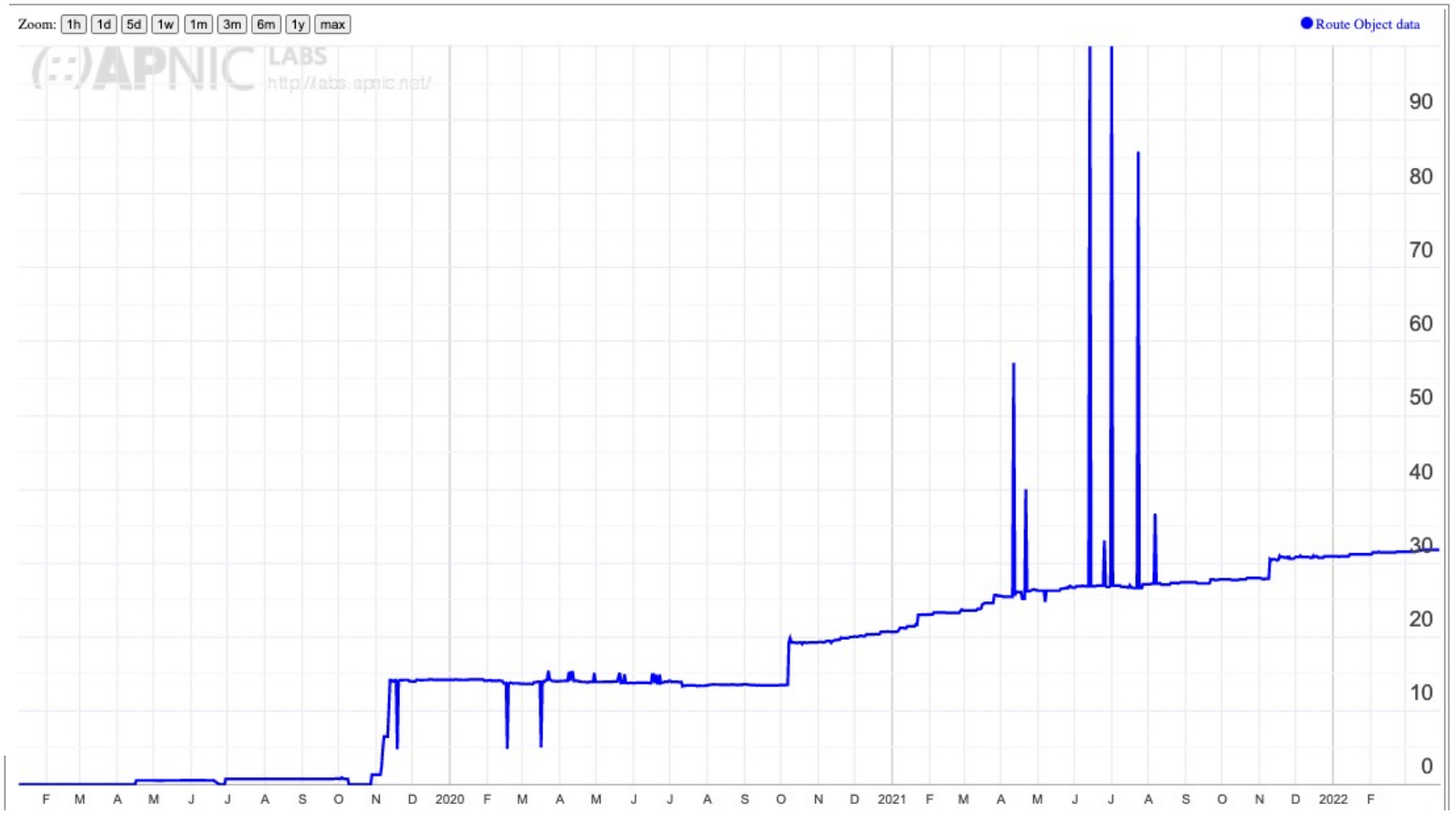
Display: **Addresses** (Advertised ROA-Valid Advertised Addresses), **IPv4**, **Percent** (of Total)



ROAs for individual networks

RPKI ROA-Validation of Advertised Routes for AS1221: ASN-TELSTRA Telstra Corporation Ltd, Australia (AU)

Display: **Route Objects** (Advertised ROA-Validated Route Advertisements), **IPv4**, **Percent** (of Total)

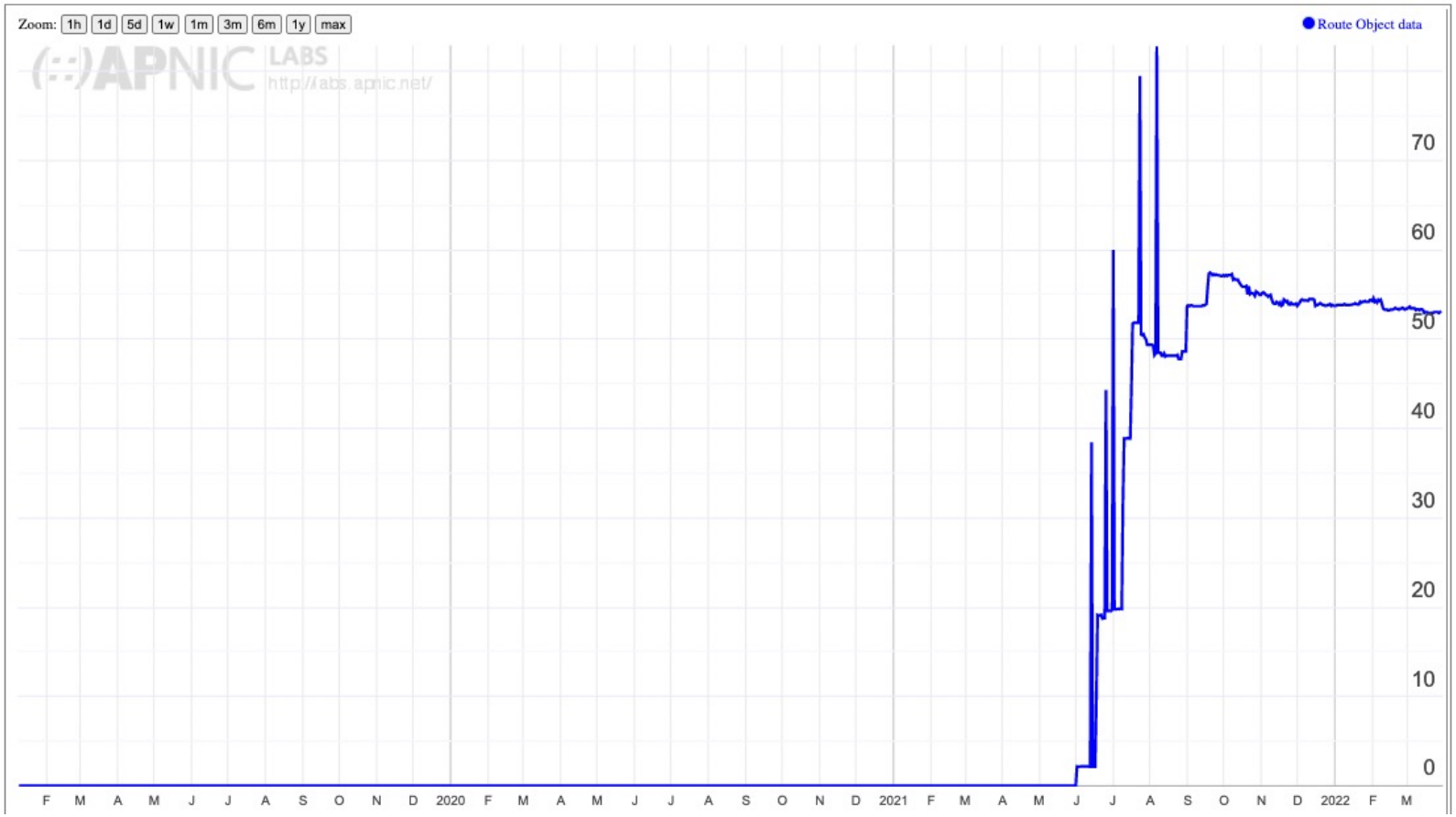


30%

ROAs for individual networks

RPKI ROA-Validation of Advertised Routes for AS4804: MPX-AS Microplex PTY LTD, Australia (AU)

Display: **Route Objects** (Advertised ROA-Validated Route Advertisements), **IPv4**, **Percent** (of Total)



50%

ROAs for individual networks

RPKI ROA-Validation of Advertised Routes for AS7545: TPG-INTERNET-AP TPG Telecom Limited, Australia (AU)

Display: **Route Objects** (Advertised ROA-Validated Route Advertisements), **IPv4**, **Percent** (of Total)

0.1%

Production vs Consumption

- So we are looking at the uptake of the generation of ROAs
- The next question is: Who is using these ROAs to determine whether to accept routes (or not!)

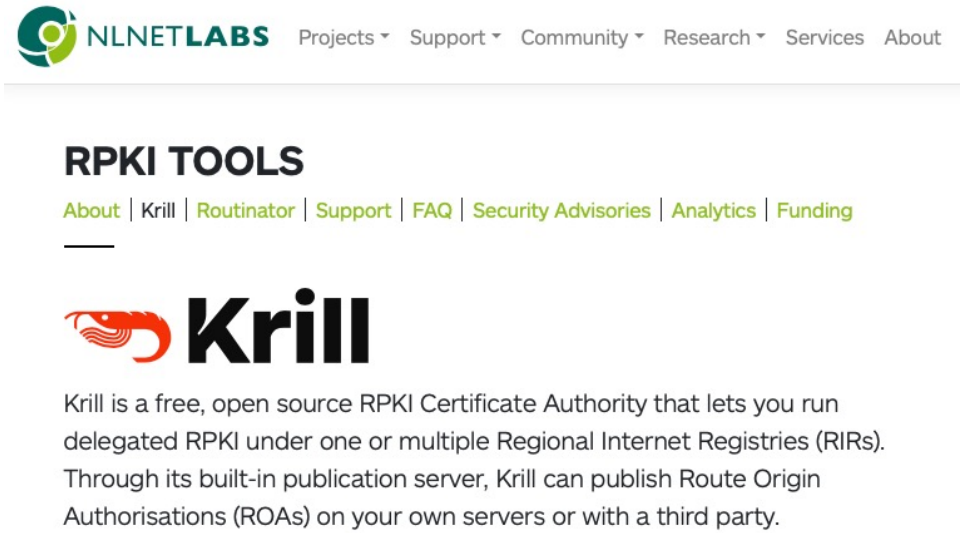
Measurement Approach

If we are looking at the effectiveness of the secure routing system in blocking the ability to direct users along bogus routing paths, then this suggests a measurement approach:

- Set up a bogus (RPKI RoV-invalid) routing path as the only route to a prefix
- Direct a very large set of users from across the Internet to try to reach a web server located at this prefix
- Use a ‘control’ of a valid routing path to the same destination
- Measure and compare

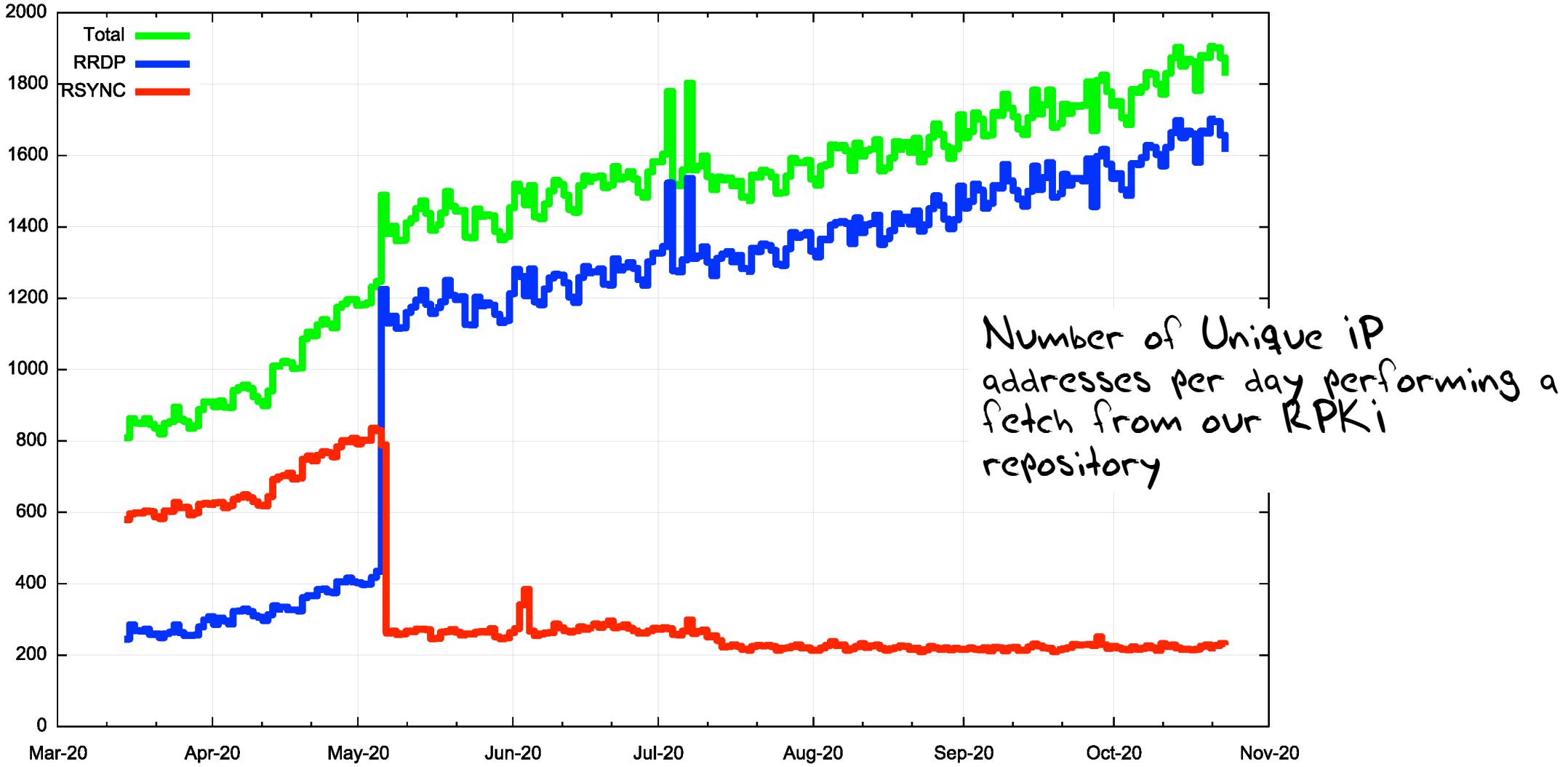
Methodology

- Set up a prefix and AS in a delegated RPKI repository
  - We used the Krill package to achieve this
  - It Just Worked! [tm]



https://www.nlnetlabs.nl/projects/rpki/krill/

Counting RPKI Clients



Number of Unique iP addresses per day performing a fetch from our RPKi repository

Methodology

- Set a prefix and AS in a delegated RPKI repository
- Regularly revoke and re-issue ROAs that flip the validity state between valid and invalid states

```
# Flip to "good" at 00:00 on Fri/Mon/Thu
0 0 * * 1,4,5 /home/krill/.cargo/bin/krillc roas update --delta ./delta-in.txt > /tmp/krillc-in.log 2>&1
# Flip to "bad" at 12:00 on sat/Tue/Thu
0 12 * * 2,4,6 /home/krill/.cargo/bin/krillc roas update --delta ./delta-out.txt > /tmp/krillc-out.log 2>&1
```

These two scripts flip the ROA valid state between ‘good’ and’bad’ origin ASNs for the prifix

Methodology

- Set a prefix and AS in a delegated RPKI repository
- Regularly revoke and re-issue ROAs that flip the validity state between valid and invalid states
- Anycast the prefix and AS pair in a number of locations across the Internet
  - We are using 3 locations: US (LA), DE (FRA), SG
  - We are using 3 transit providers
  - The server at this location delivers 1x1 blots
  - This is IPv4-only at this point

Methodology

- Set a prefix and AS in a delegated RPKI repository
- Regularly revoke and re-issue ROAs that flip the validity state between valid and invalid states
- Anycast the prefix and AS pair in a number of locations across the Internet
  - We started by using 3 locations: US (LA), DE (FRA), SG
  - We then enlisted the assistance of a very large cloud provider and expanded this to more than 200 locations!

Methodology

- Set a prefix and AS in a delegated RPKI repository
- Regularly revoke and re-issue ROAs that flip the validity state between valid and invalid states
- Anycast the prefix and AS pair in a number of locations across the Internet
- Load a unique URL that maps to the destination into a measurement script
  - The DNS component uses HTTPS and a unique DNS label component to try and ensure that the HTTP FETCH is not intercepted by middleware proxies

Methodology

- Set a prefix and AS in a delegated RPKI repository
- Regularly revoke and re-issue ROAs that flip the validity state between valid and invalid states
- Anycast the prefix and AS pair in a number of locations across the Internet
- Load a unique URL that maps to the destination into a measurement script
- Feed the script into the advertising systems
  - This is part of the larger APNIC Labs ad-based measurement system – this test is one URL in a larger collection of URLs

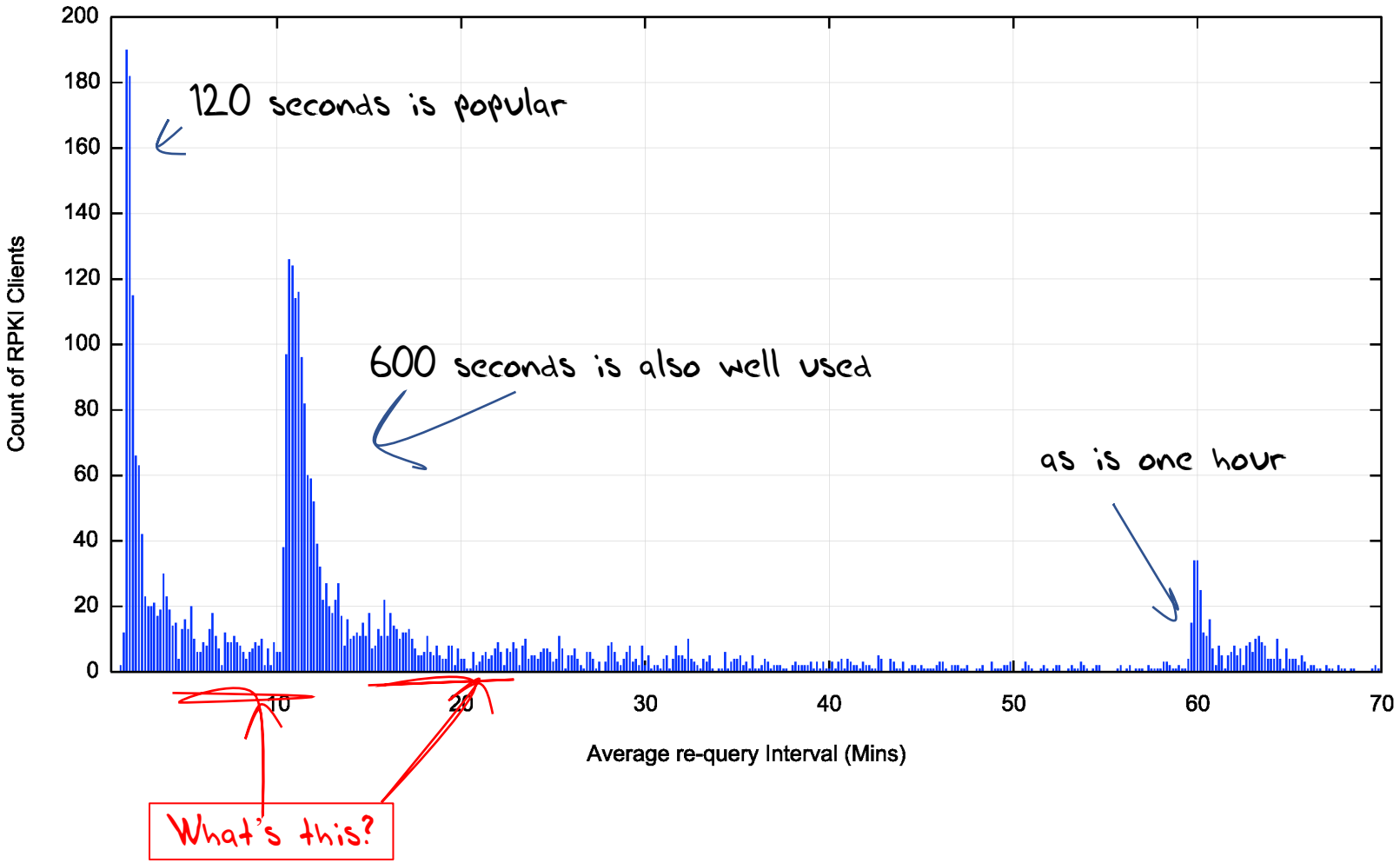
Methodology

- Set a prefix and AS in a delegated RPKI repository
- Regularly revoke and re-issue ROAs that flip the validity state between valid and invalid states
- Anycast the prefix and AS pair in a number of locations across the Internet
- Load a unique URL that maps to the destination into a measurement script
- Feed the script into the advertising systems
- Collect and analyse data
  - We use the user record of successful fetch to avoid zombies and stalkers

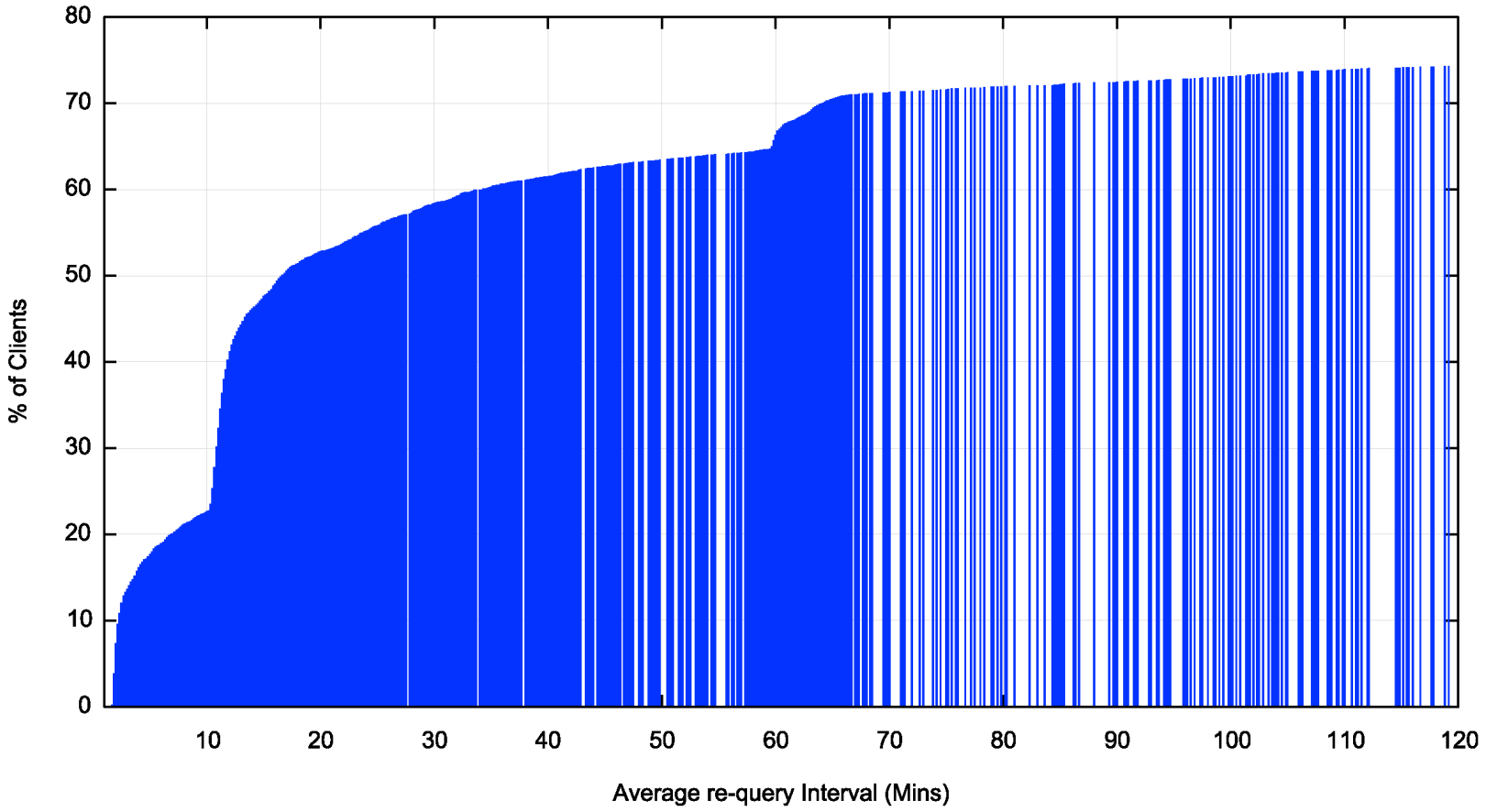
Flipping ROA states

- What's a good frequency to flip states?
  - How long does it take for the routing system as a whole to learn that a previously valid route is now invalid? And how long for the inverse invalid to valid transition
- Validity / Invalidity is determined by what is published at the RPKI publication point
  - Each transition is marked by revocation of the previous ROA's EE certificate and the issuing of a new ROA and EE certificate
- What's the re-query interval for clients of a RPKI publication point?
  - There is no standard-defined re-query interval so implementors have exercised their creativity!

RPKI Publication Point Re-Query Intervals (first hour)

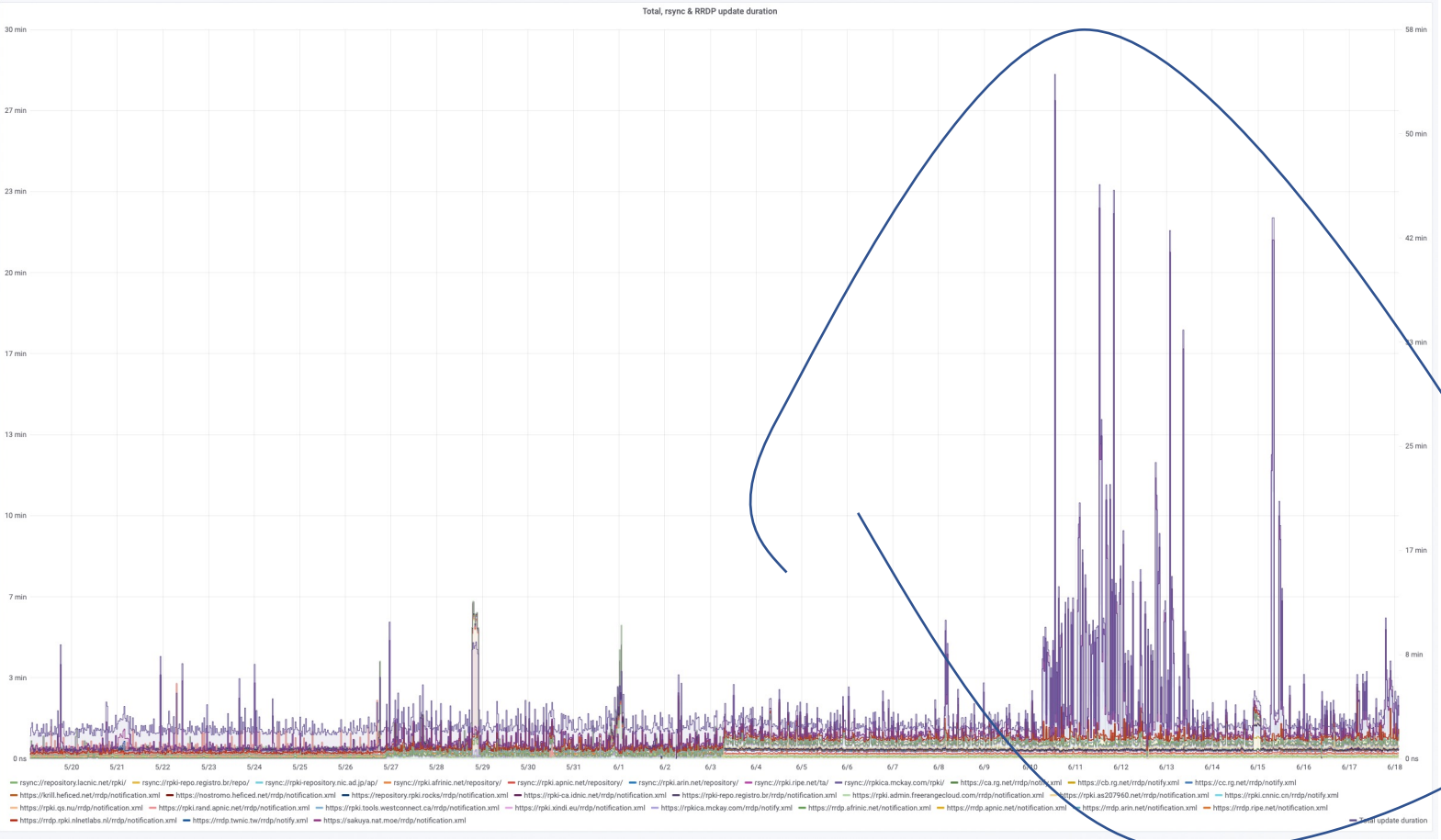


Re-Query - Cumulative Distribution



Within 2 hours we see 75÷ of clients perform a requery

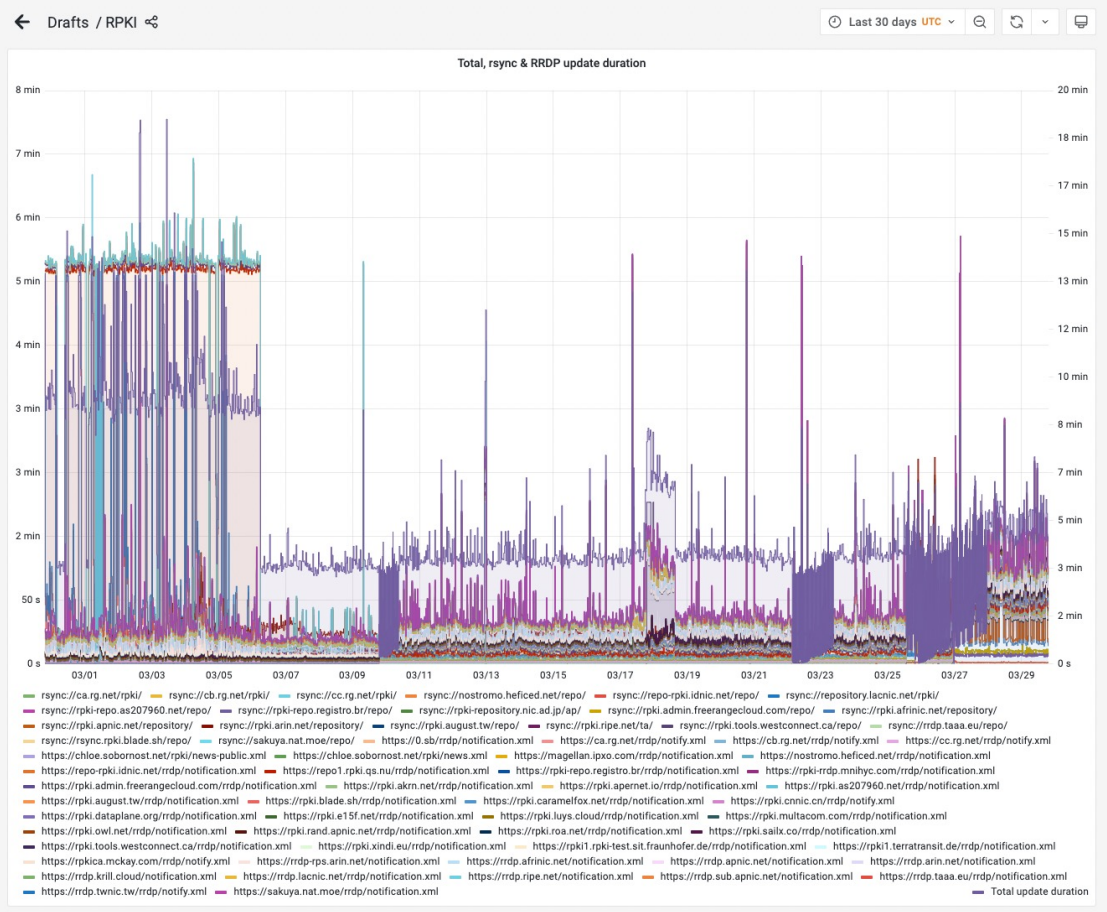
Why the tail lag?



Clients can take a significant amount of time to complete a pass through the entire RPKI distributed repository set, which makes the entire system sluggish to respond to changes

https://grafana.wikimedia.org/d/UwUa77GZk/rpki?panelId=59&fullscreen&orgId=1&from=now-30d&to=now

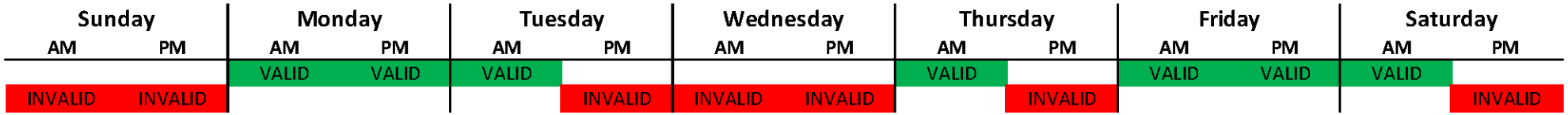
This is NOT scaling well



As more publication points and clients are added to the RPKI, the RPKI scanning function is getting slower (and more variable)

https://grafana.wikimedia.org/d/UwUa77GZk/rpki?from=now-30d&orgId=1&to=now&viewPanel=59

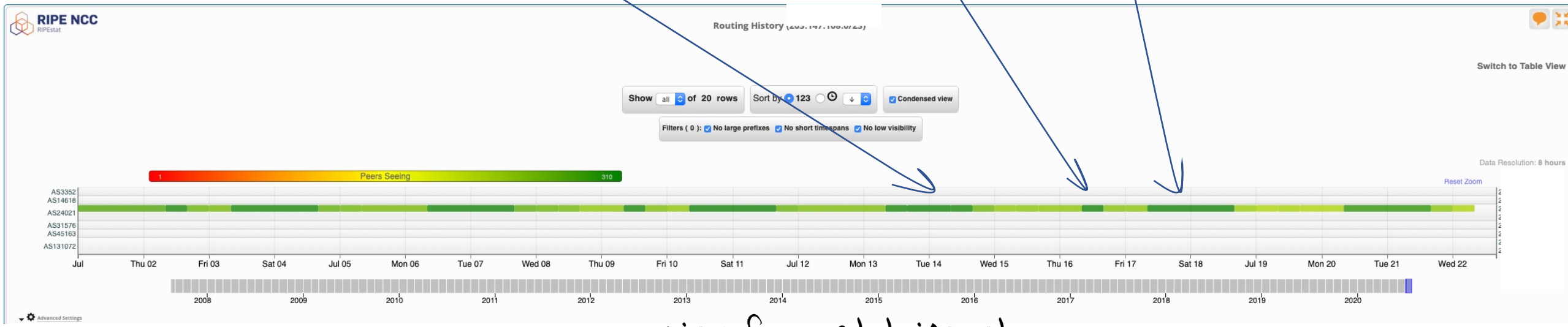
We use 12 and 36 hour held states for ROA validity



The route object validity state cycles over a 7 day period in a set of 12 and 36 hour intervals

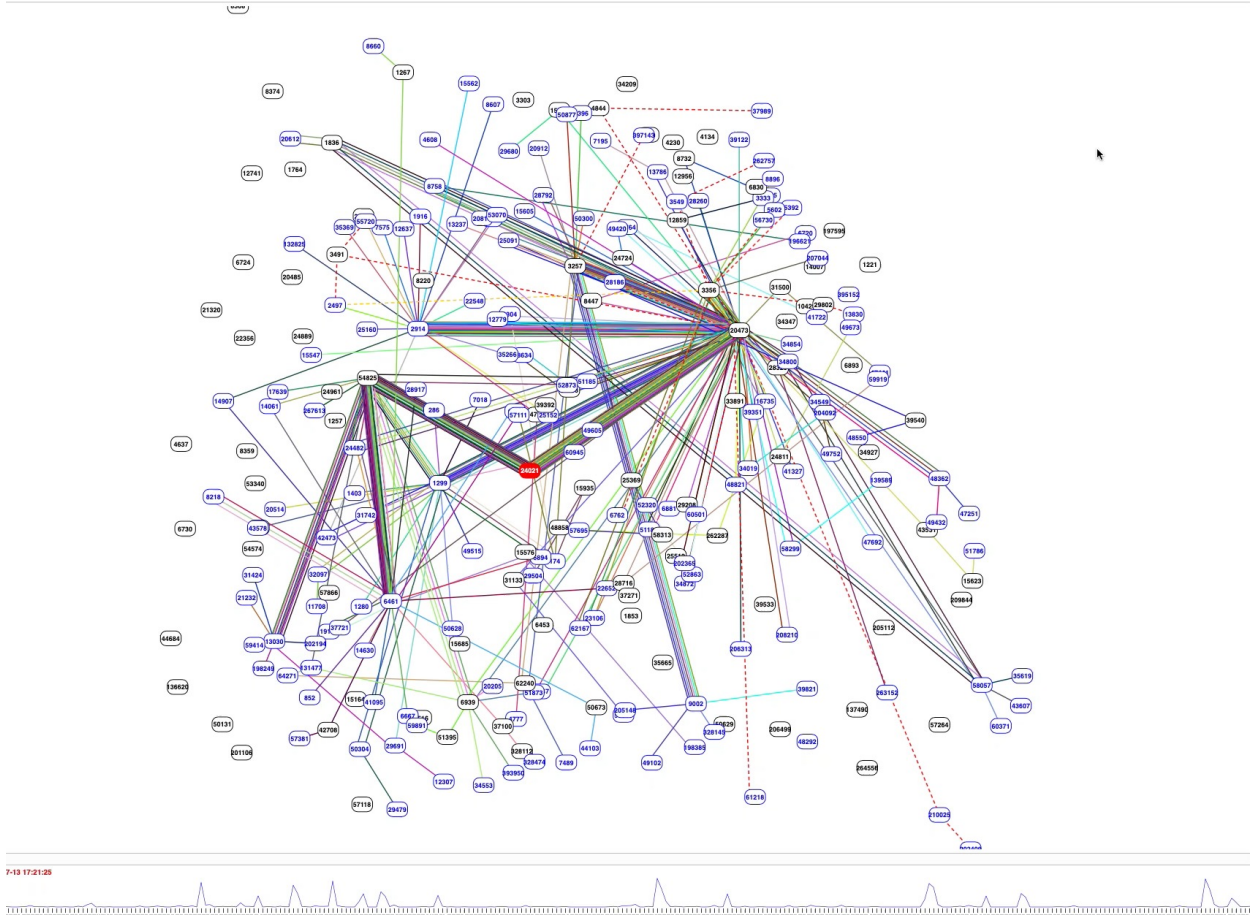
We use 12 and 36 hour held states

Sunday		Monday		Tuesday		Wednesday		Thursday		Friday		Saturday	
AM	PM	AM	PM	AM	PM	AM	PM	AM	PM	AM	PM	AM	PM
		VALID	VALID	VALID				VALID		VALID	VALID	VALID	
INVALID	INVALID				INVALID	INVALID	INVALID		INVALID				INVALID



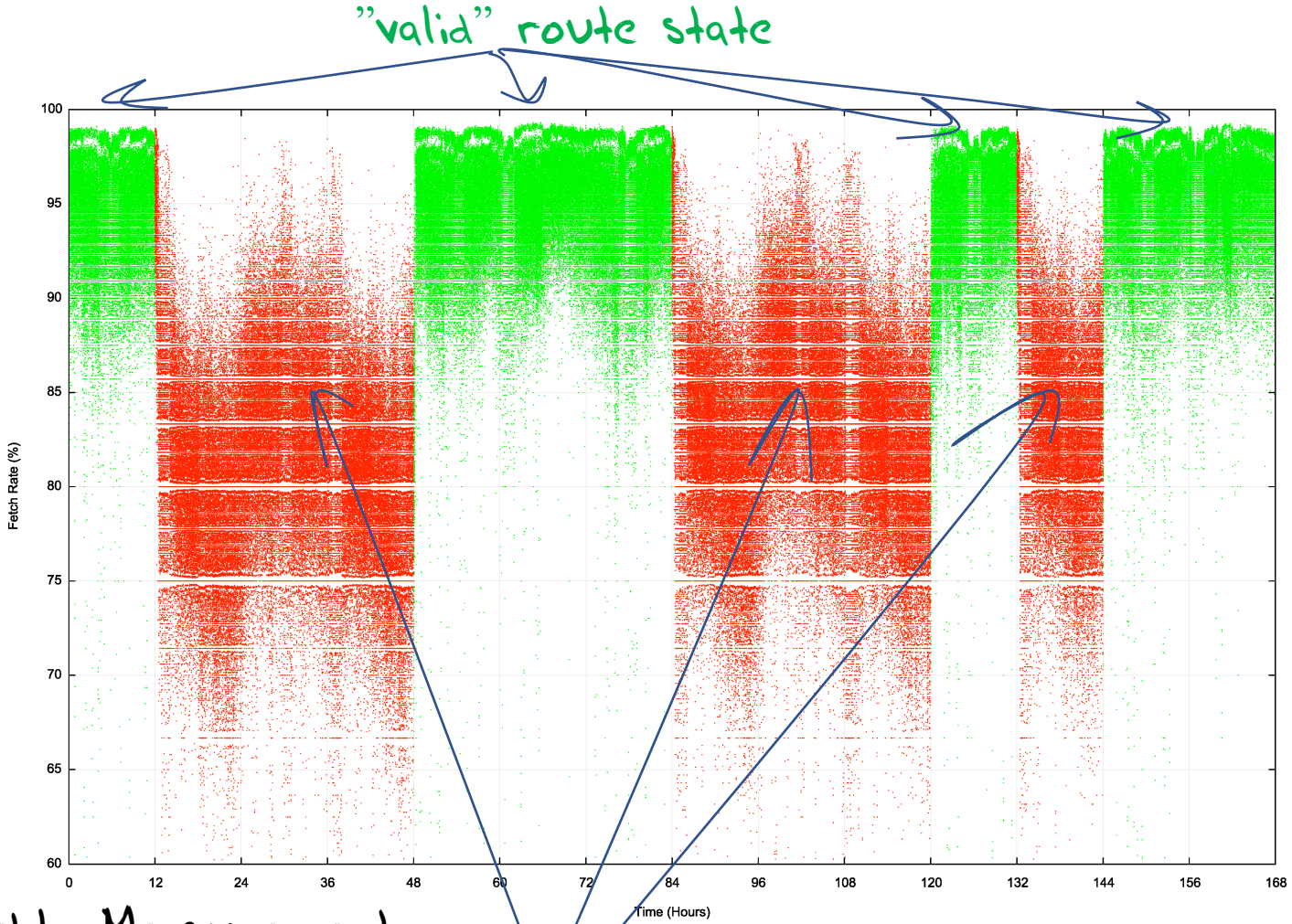
view from stat.ripe.net

We use 12 and 36 hour held states



BGP Play view of the routing changes

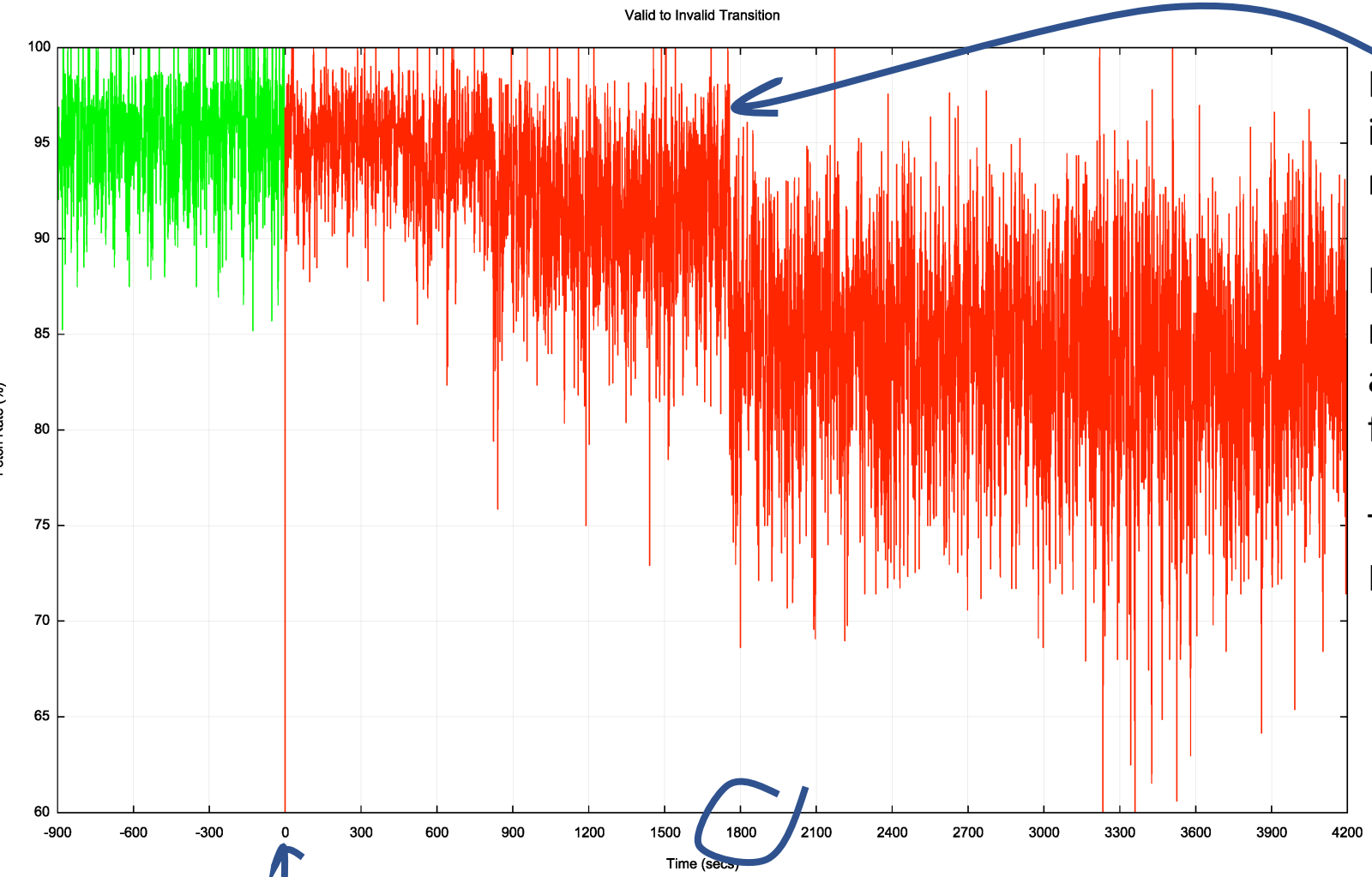
We used 12 and 36 hour states



This shows the per-second fetch rate when the route is valid (green) and invalid (red) over a 7 day window

The route validity switches are clearly visible

Transition - Valid to Invalid



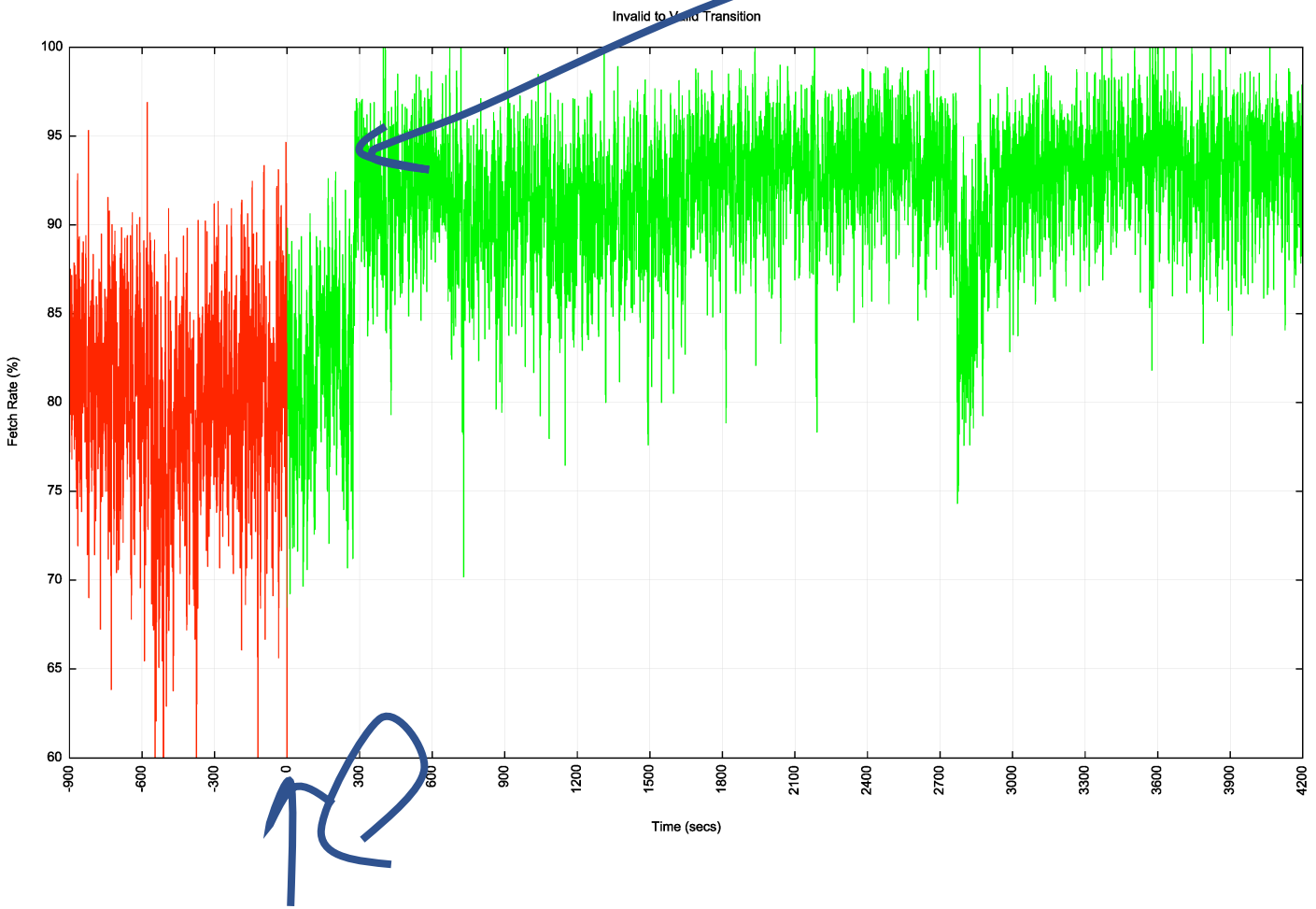
Time of ROA change at the RPKI repository

It takes some **30 minutes** for the valid to invalid transition to take effect in this measurement

It appears that this is a combination of slow re-query rates at the RPKI publication point and some delays in making changes to the filters being fed into the routers

This system is dependant on the last transit ISP to withdraw

Transition - Invalid to Valid



Time of ROA change at the RPKi repository

It takes some **5 minutes** for the invalid to valid transition to take effect in this measurement

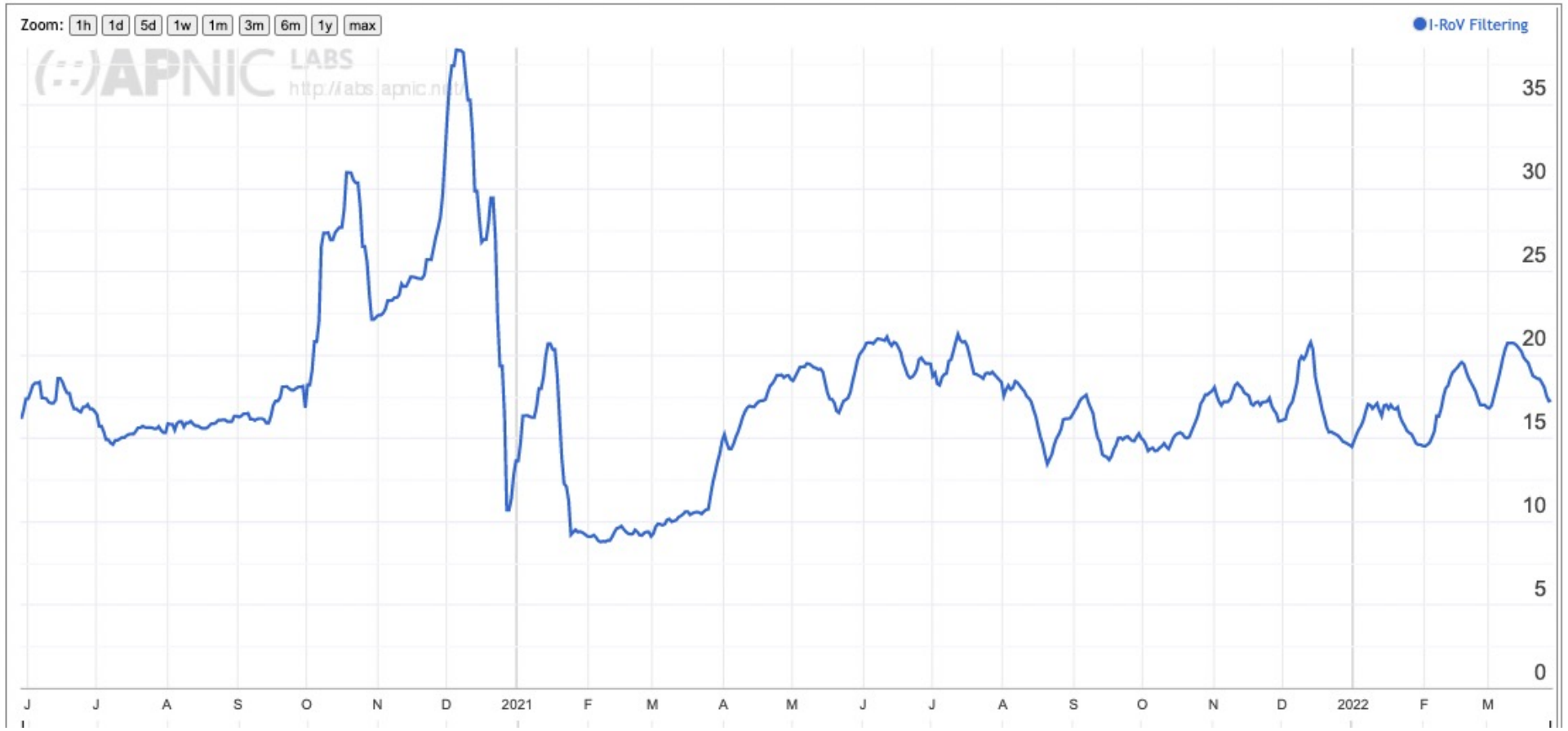
This system is dependant on the first transit ISP to announce, so it tracks the fastest system to react

RPKI "sweep" software

- There is a mix of 2, 10 and 60 minute timers being used
- 2 minutes seems like a lot of thrashing with little in the way of outcome – the responsiveness of the system is held back by those clients using longer re-query timers
- 60 minutes seems too slow

(I'd go with a 10 minute query timer as a compromise here)

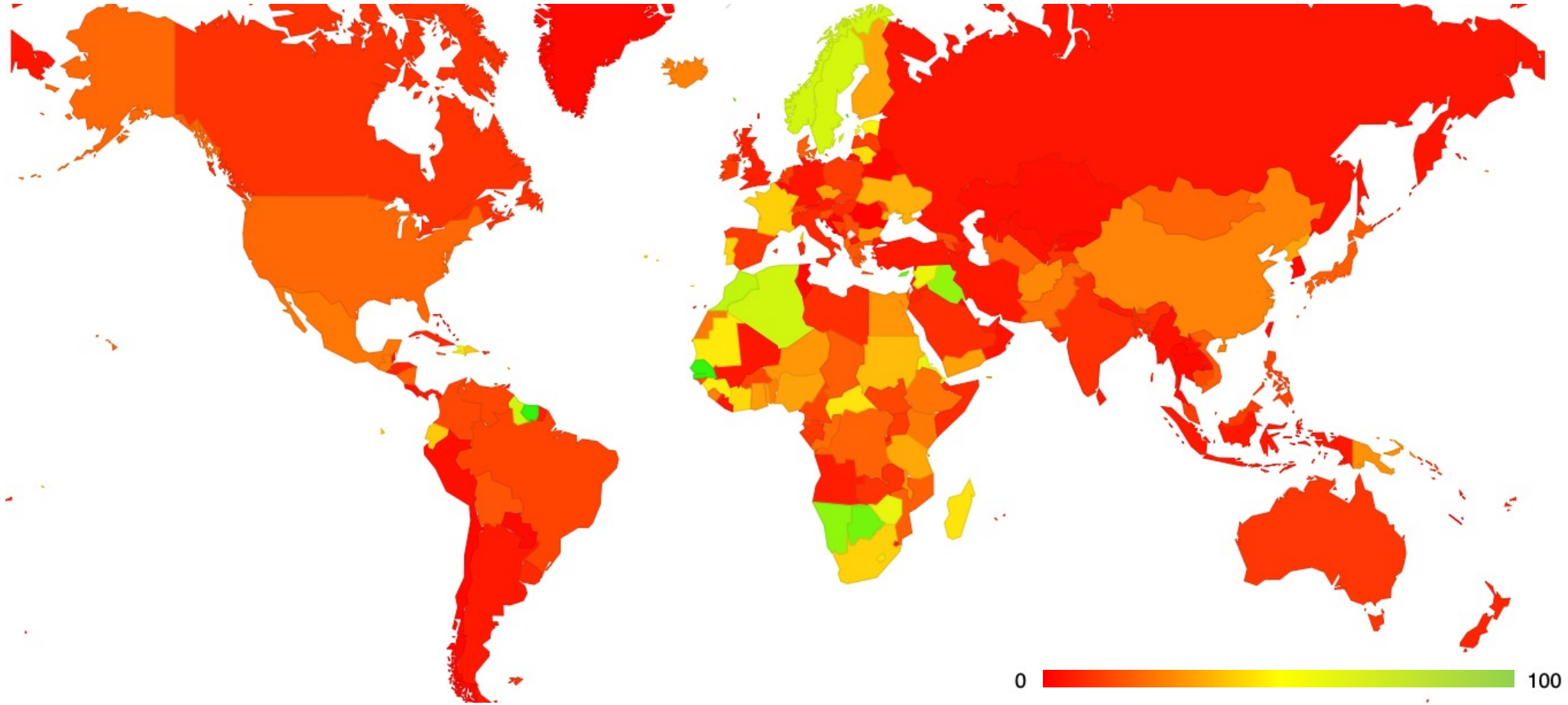
User impact of RPKI filtering



At 20% of users that's a surprisingly large impact for a very recent technology

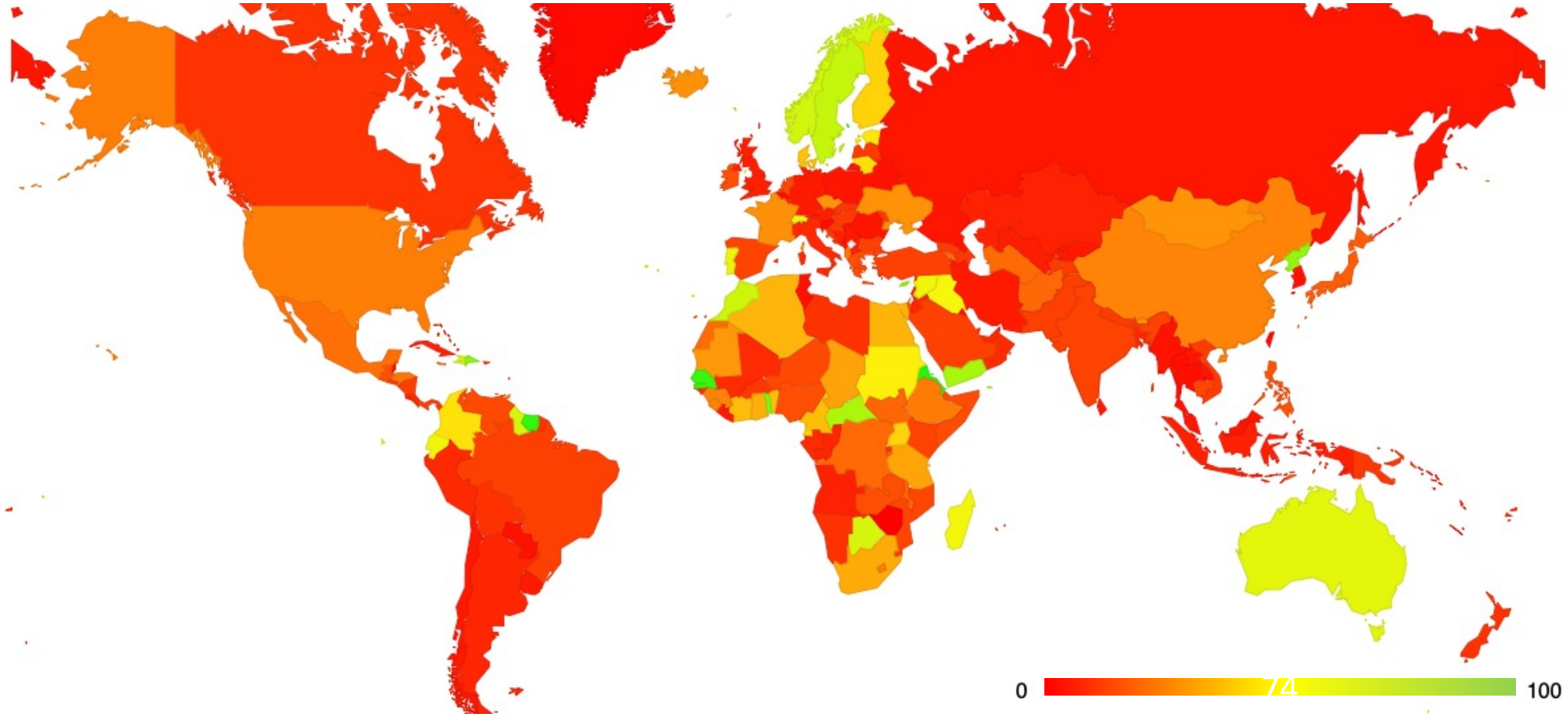
However, nothing much has changed in the past 16 months!

Results: User Impact of RPKI filtering - Jul 2020



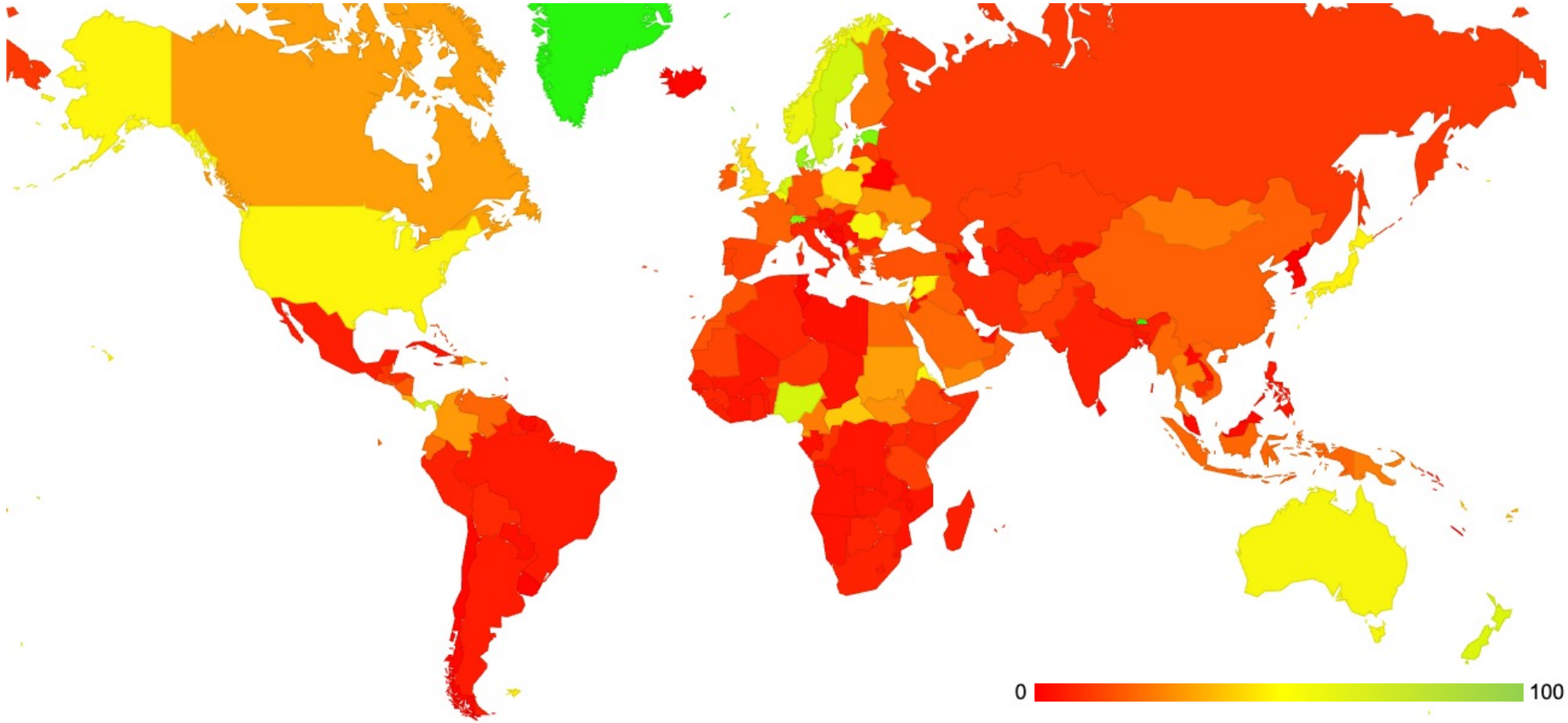
https://stats.labs.apnic.net/rpki

Results: User Impact of RPKI filtering - Oct 2020



https://stats.labs.apnic.net/rpki

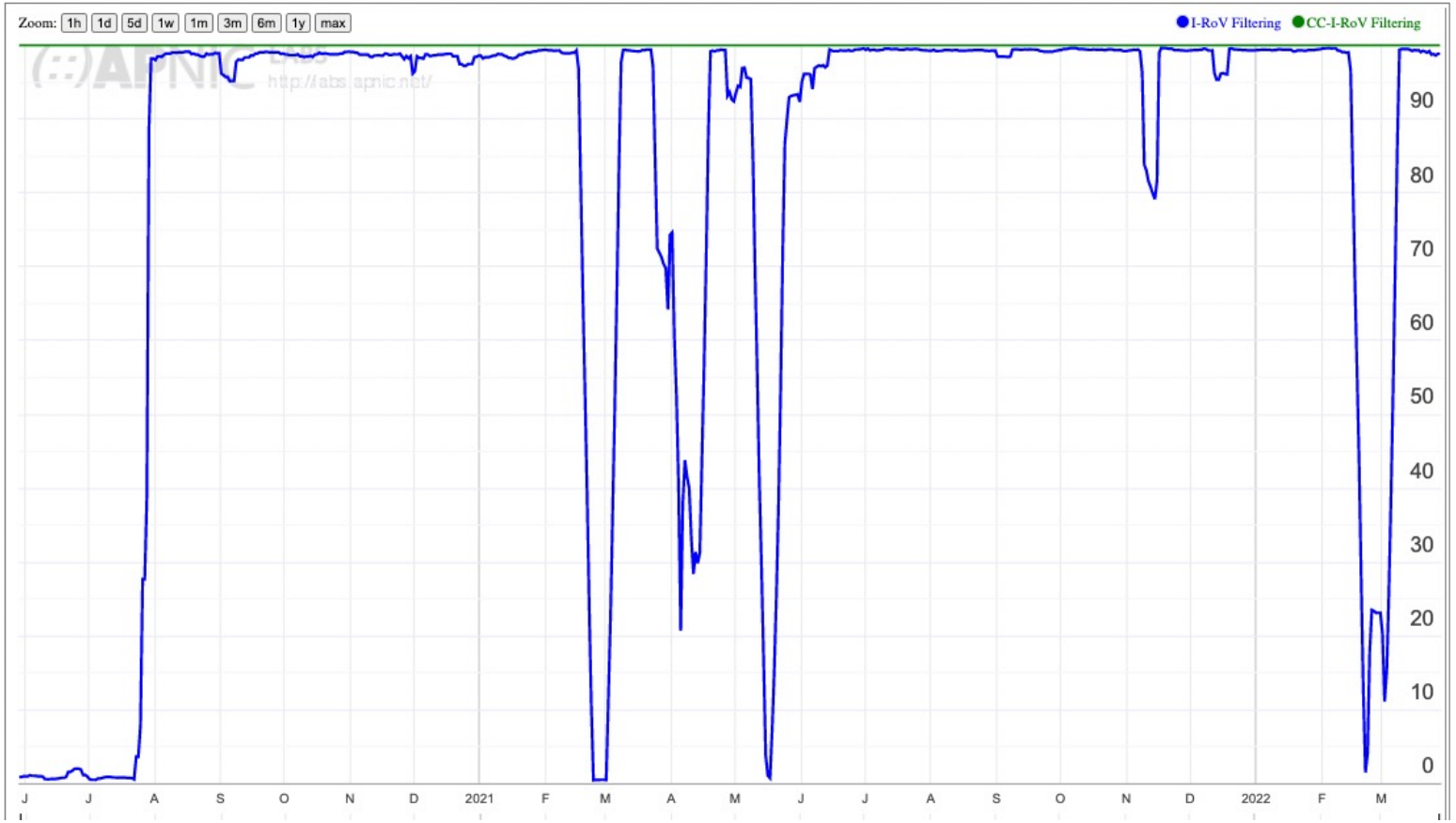
Results: User Impact of RPKI filtering - Mar 2022



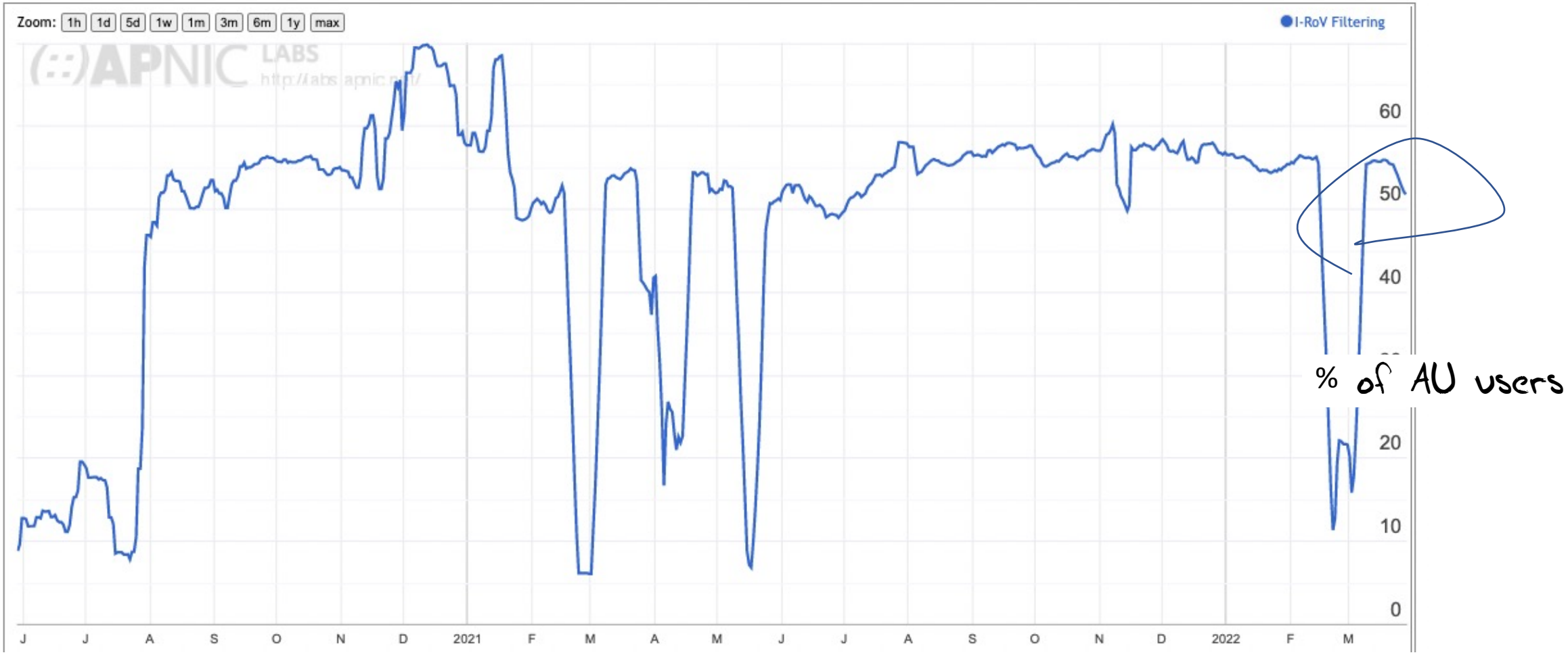
https://stats.labs.apnic.net/rpki

Turning on Drop Invalid Filtering

RPKI I-ROV Per-Country filtering for AS1221: ASN-TELSTRA Telstra Corporation Ltd, Australia (AU)



Australia


Zoom: 1h 1d 5d 1w 1m 3m 6m 1y max
I-RoV Filtering
APNIC LABS
60
50
40
20
10
0
J J A S O N D 2021 F M A M J J A S O N D 2022 F M
% of AU users

Local ISPs

ASN	AS Name	RPKI Validates	Samples
AS1221	ASN-TELSTRA Telstra Corporation Ltd	98.80%	76,646
AS4804	MPX-AS Microplex PTY LTD	1.13%	34,655
AS7545	TPG-INTERNET-AP TPG Telecom Limited	0.38%	28,769
AS133612	VODAFONE-AS-AP Vodafone Australia Pty Ltd	0.81%	11,706
AS9443	VOCUS-RETAIL-AU Vocus Retail	99.66%	9,293
AS4764	WIDEBAND-AS-AP Aussie Broadband	14.07%	9,153
AS135887	TELSTRA-BELONG-AP Belong Telstra Corporation	98.71%	4,949

Does everyone have to play?

There are two factors at play:

- Networks that perform invalid route filtering

and

- Network that do not filter themselves, but are customers of transit providers who filter

In either case the basic RPKI RoV objective is achieved, in that the users within these ISP networks are not exposed to invalid route objects

Next Steps for Measurement

This is a work in progress and would benefit from more refinement, including:

- Could we attempt selective traceroute from the anycast servers to identify the networks that are performing the RoV invalid filter drop?
  - The measurement setup detects the user impact but not the individual networks who are performing drop invalid. Selective traceroute may allow a better way to identify the point of invalid drop
- Should we perform further analysis of BGP route updates in route collectors to determine route withdrawal and announcement patterns when RPKI validity changes?
  - What is the difference between the primary point of route withdrawal / announcement and the consequent propagation in eBGP to the surrounding networks?

Questions we might want to think about

Stub vs Transit

- Is it necessary for every AS to operate RPKI ROV infrastructure and filter invalid routes?
- If not, what's the minimal set of filtering networks that could provide similar levels of filtering for the Internet as a whole
- What's the marginal benefit of stub AS performing RPKI ROV filtering?

Questions we might want to think about (2)

Ingress vs Egress

- Should a stub AS RPKI only RoV filter its own announcements?
- Should every AS filter their own announcements?
- What's more important: Protecting others who DON'T RoV filter from your operational mishaps or protecting yourself from the mishaps of others?
- Does Partial Adoption change your answer?

Questions we might want to think about (3)

Prefix vs AS attestations

- Should an AS be able to enumerate ALL of its originations in a AS-signed attestation?

Questions we might want to think about (4)

When and how will we protect the AS Path?

- What is going in with the ASPA drafts in the IETF?
- Is anyone experimenting with ASPA yet?

- What is the benefit of Origination protection without AS Path protection?

What are we trying to achieve here?

- If this is a routing protection measure then what are you trying to protect? From whom? From what threat?
- If this is guard against operational errors then don't forget that operational mishaps are endlessly varied, and we can't foresee all possible causes of routing accidents!
- If this is a user protection measure then the issue of route filtering is an issue for transit providers, not stub networks
  - A stub network should generate ROAs for its routes, but there is far less of an incentive to perform RoV invalid filtering if the stub's upstreams / IXs are already performing this filtering
  - Is it more important for IXs and Transits to perform drop-invalids than for stubs?

Thanks!