Measuring RPKI Effectiveness

Geoff Huston

APNIC

Routing Security

That’s the objective of routing security?

Pick One, and only One?

If you had to pick just one objective what would you choose?

- ❑Protect the routing system from all/some operational mishaps?
- ❑Protect the routing system from all/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?

A Touch of Realism

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

A Touch of Realism

* BGP is not a deterministic protocol, but more of a negotiation protocol that attempts to find meta-stable solutions to matching importer/export policy preferences simultaneously.

  Where the policies are not uniquely compatible the BGP "solution" is not necessarily reached deterministically and different outcomes will be seen at different times – see RFC4264 "BGP Wedgies" for an illustration of this form of indeterminism

My Choice…

If you had to pick just one objective what would you choose?

- ❑ Protect the routing system from all/some operational mishaps?
- ❑ Protect the routing system from all/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?

My Objective

I want to measure the “impact” of invalid route filtering on users

The question I want to answer here is user-centric:

- **What proportion of users can’t reach a destination when the only available destination route is invalid according to ROV?**

I’d like to position 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

* “RoV filtering” is shorthand to “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 too specific or wrong origin AS

Methodology

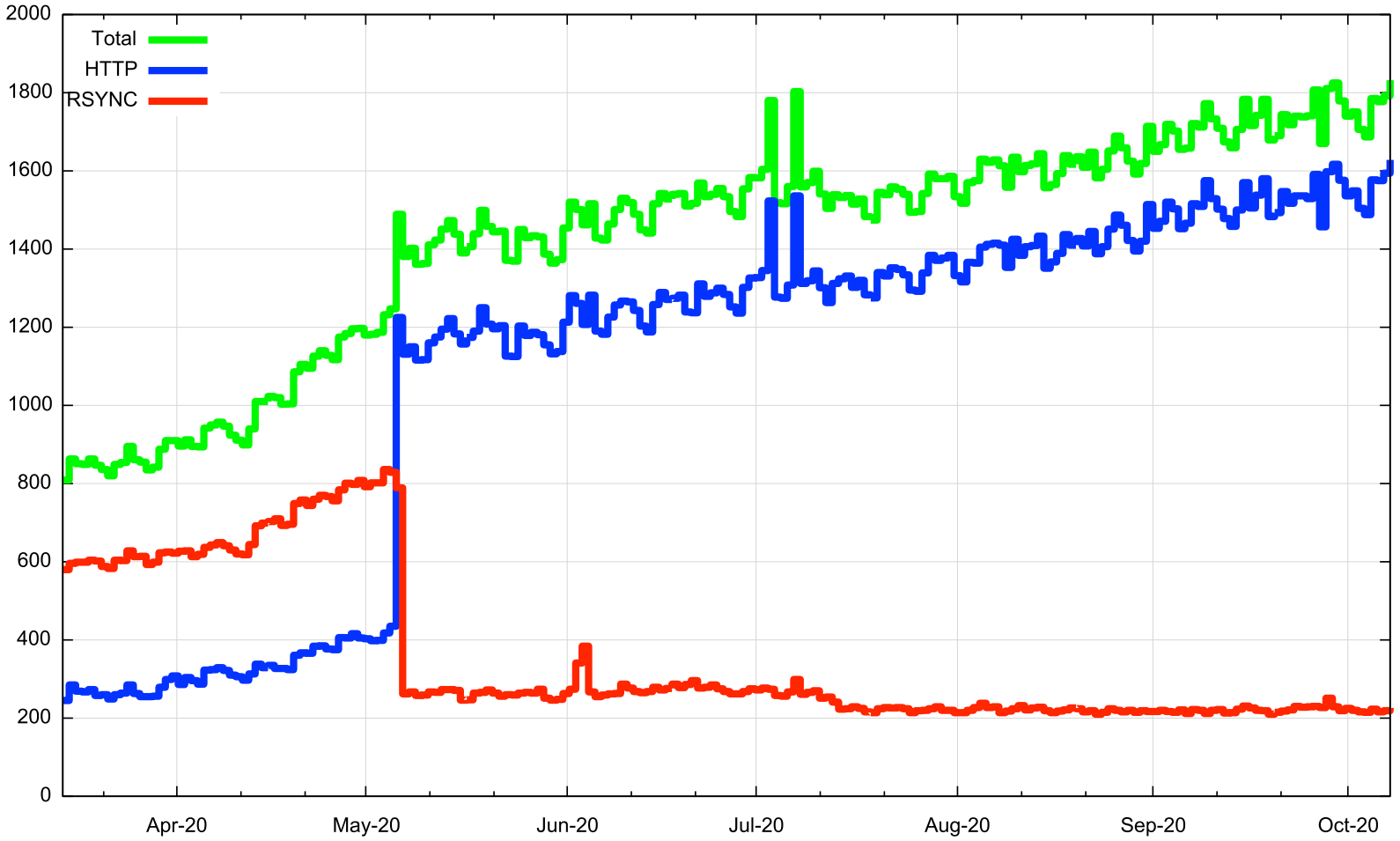
If we are looking at the effectiveness of the secure routing system in blocking the ability to direct users along bogus routing paths, then this suggests a measurement approach:

- 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

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

Aside: Counting RPKI Clients



Number of Unique IP addresses per day performing a fetch from our RPKI repository

Methodology

2. Regularly revoke and re-issue ROAs that flip the validity state between valid and invalid states, using time slicing to flip between measurement case and control case

```
# Flip to "good" at 00:00 on Fri/Mon/Thu
0 0 * * 1,4,5 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 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

3. Anycast the prefix and AS pair in a number of locations across the Internet

- 3 locations: US (LA), DE (FRA), SG
- 3 transit providers
- The server at these locations only delivers 1x1 blots from the anycast service address
- This is IPv4-only at this point

Methodology

4. Load a unique URL that maps to the destination into a measurement script

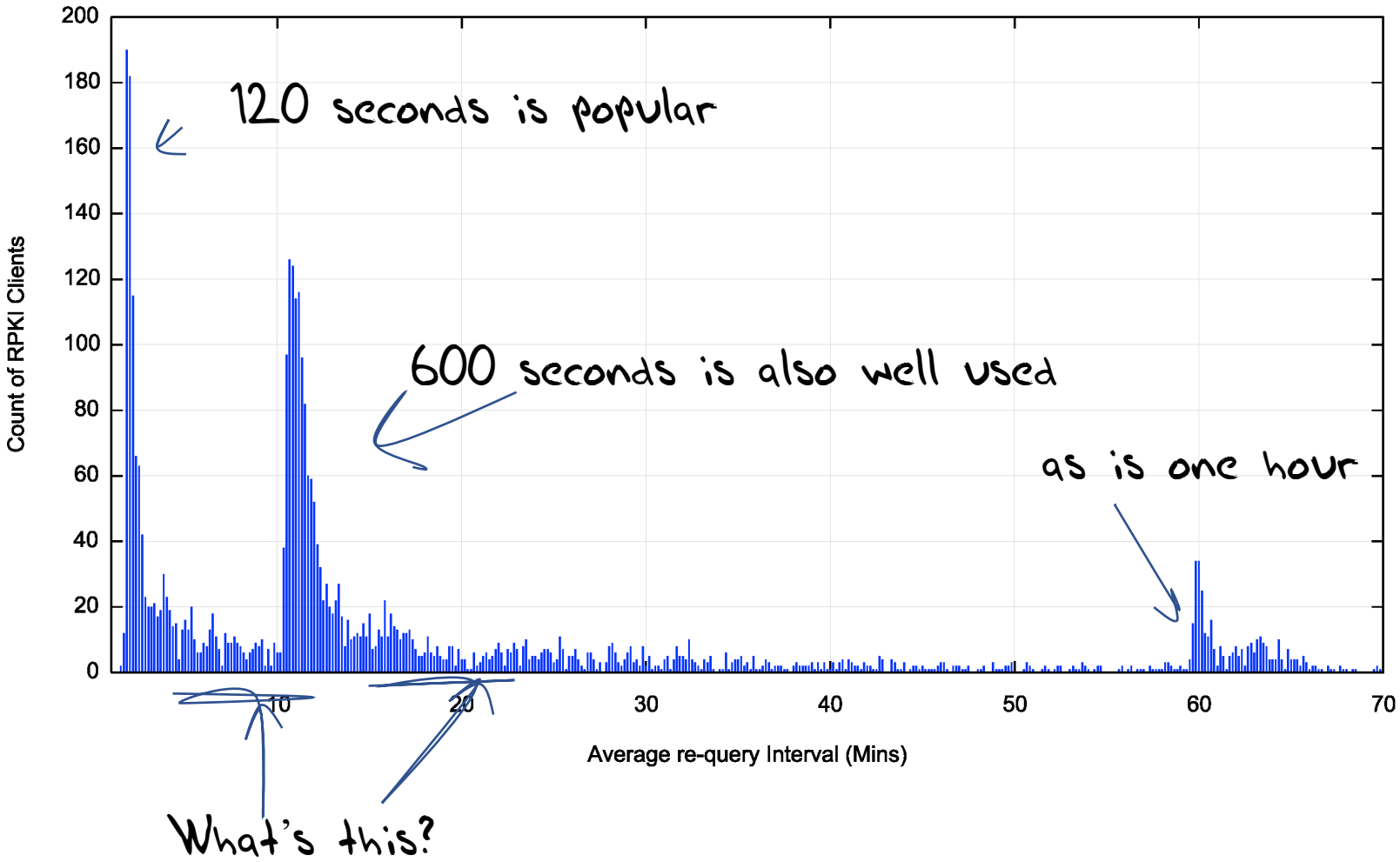
- The DNS component uses HTTPS and a unique DNS label component to try and ensure that the HTTP FETCH (which is the actual RoV test) is not intercepted by caching middleware proxies

Feed a script into an advertising campaign to have this test performed on a large scale

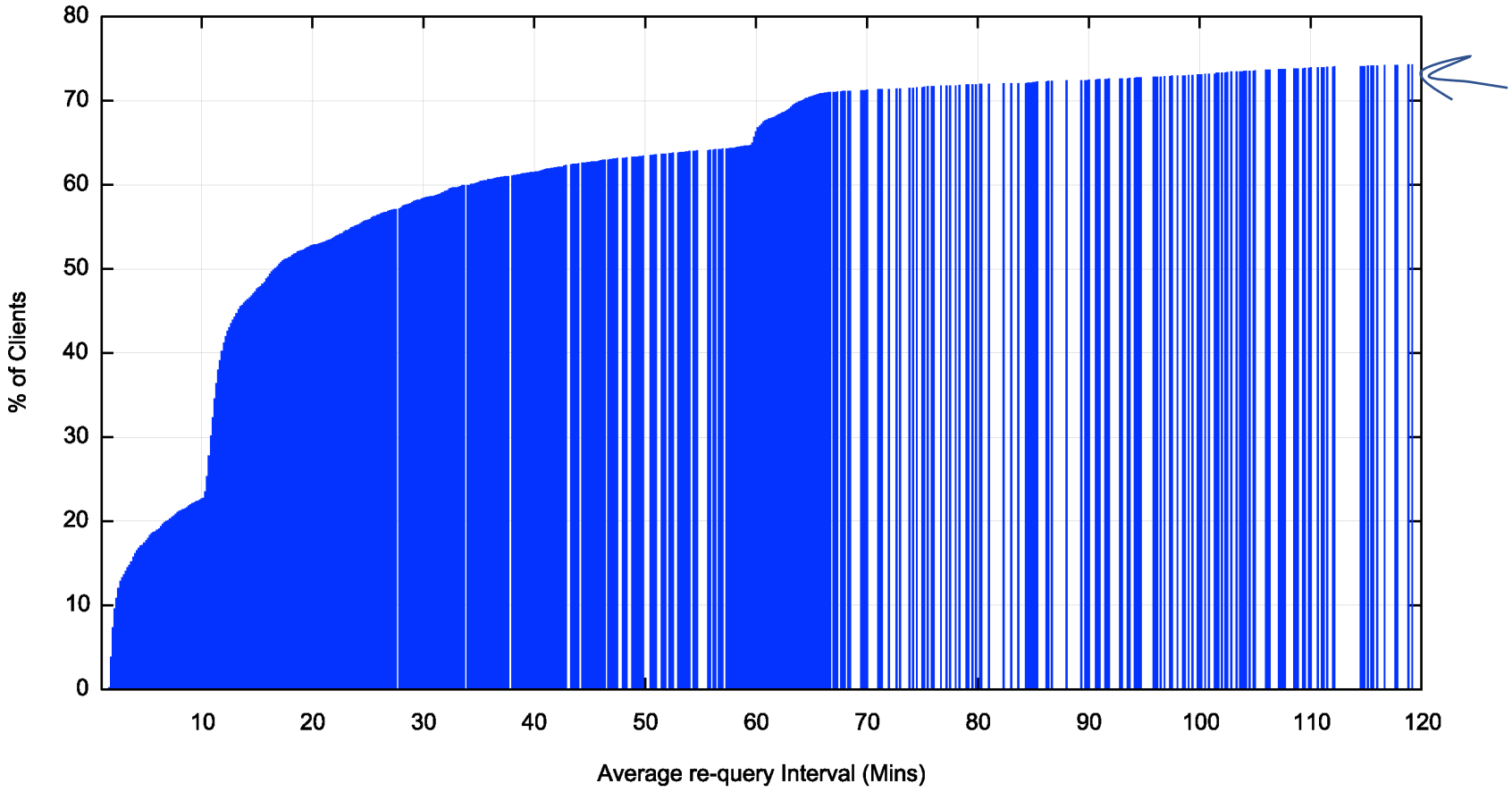
Flipping ROA states

- 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!

Re-Query Intervals (first hour)

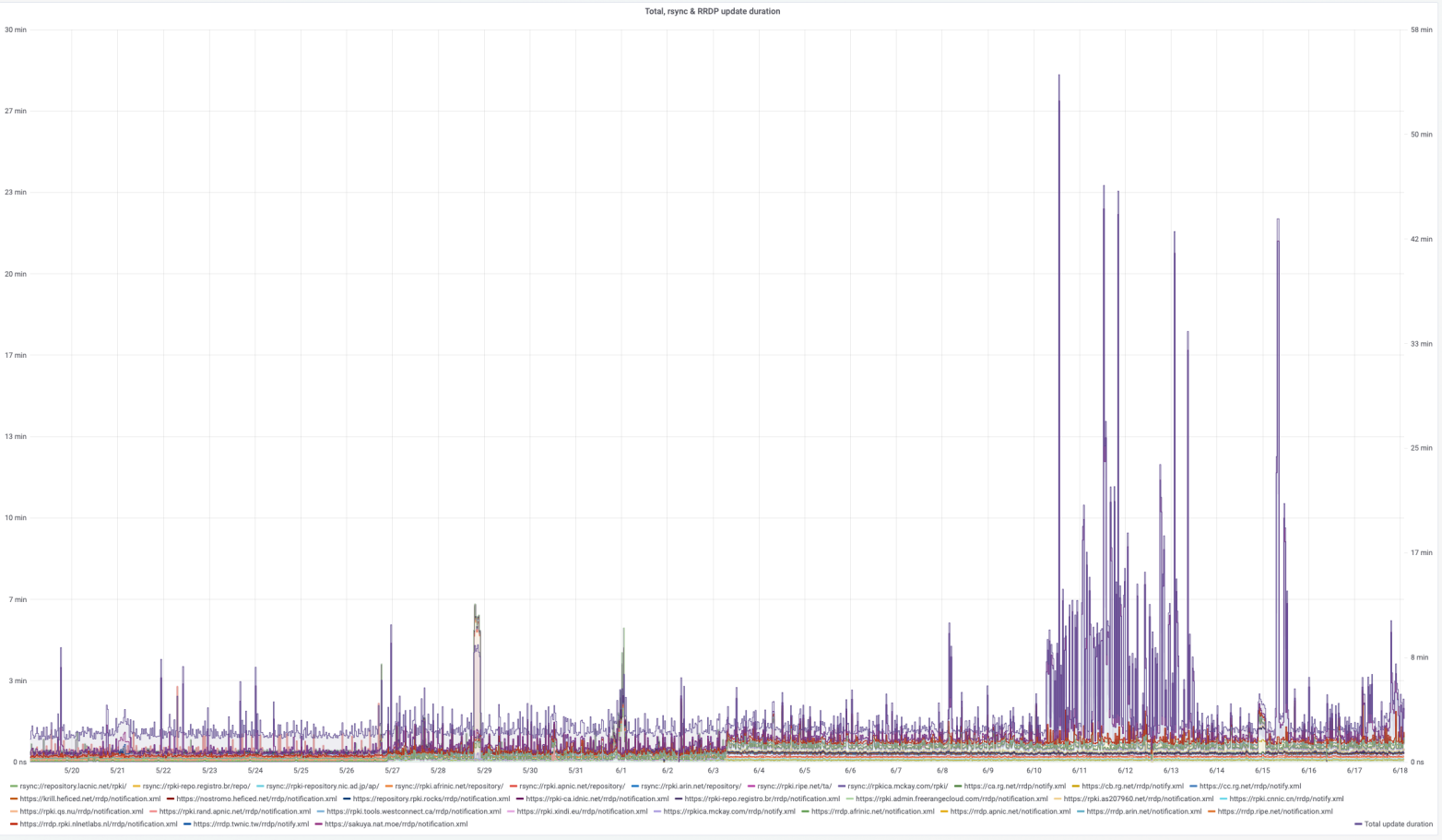


Re-Query - Cumulative Distribution



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

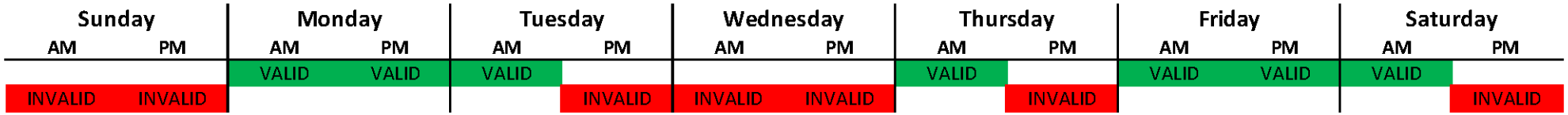
Why the tail lag?



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

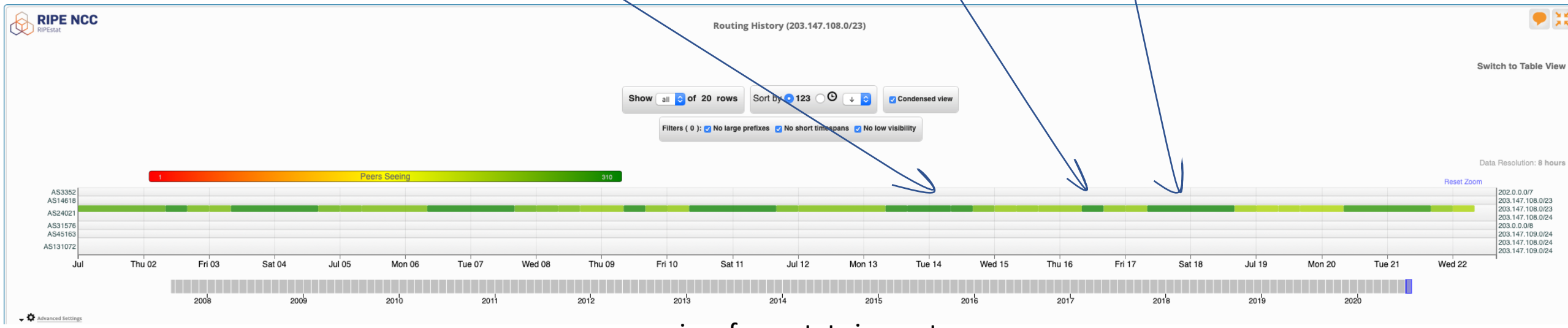
We use 12 and 36 hour held states



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

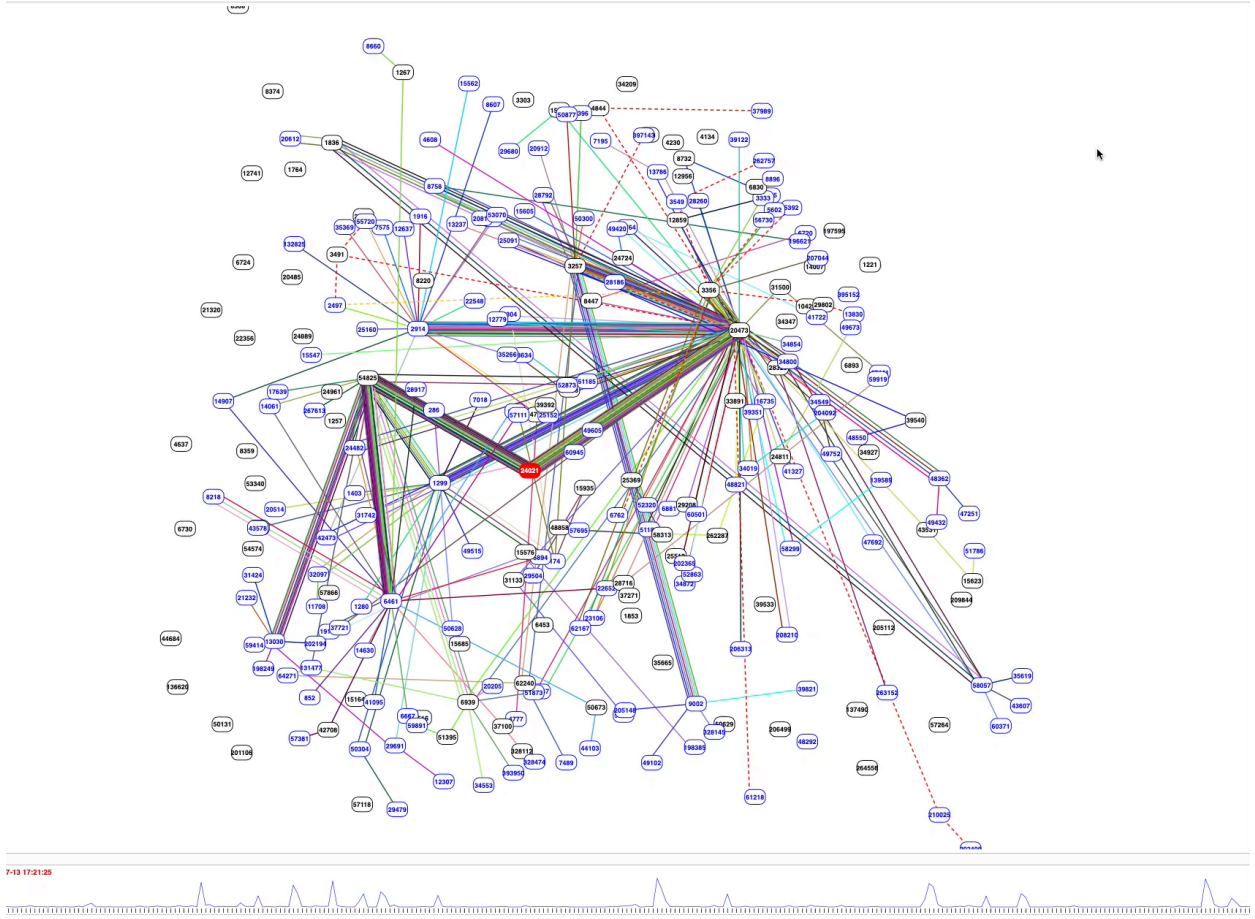
We use 12 and 36 hour held states

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



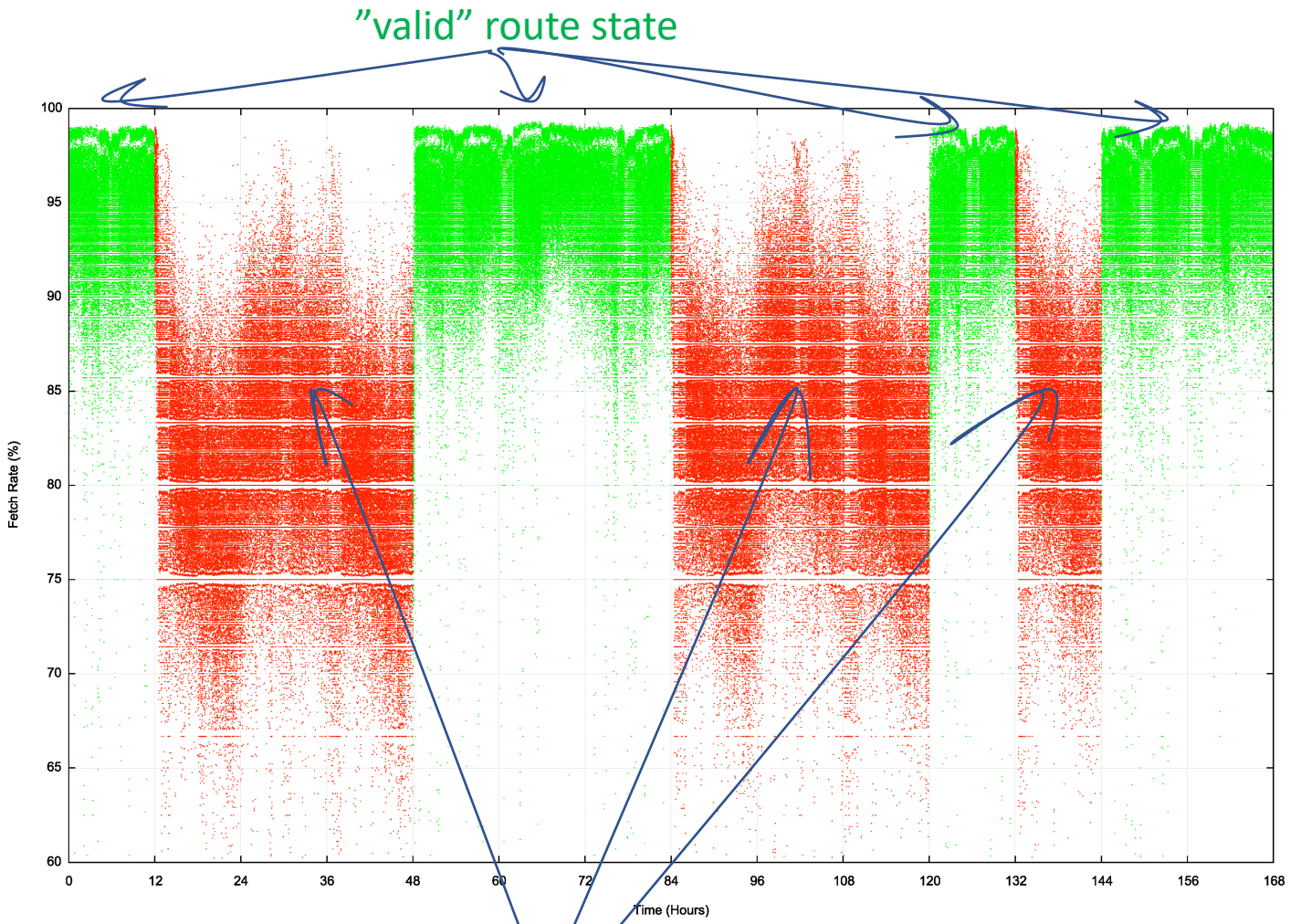
view from stat.ripe.net

We use 12 and 36 hour held states



BGP Play view of the routing changes

We use 12 and 36 hour held states

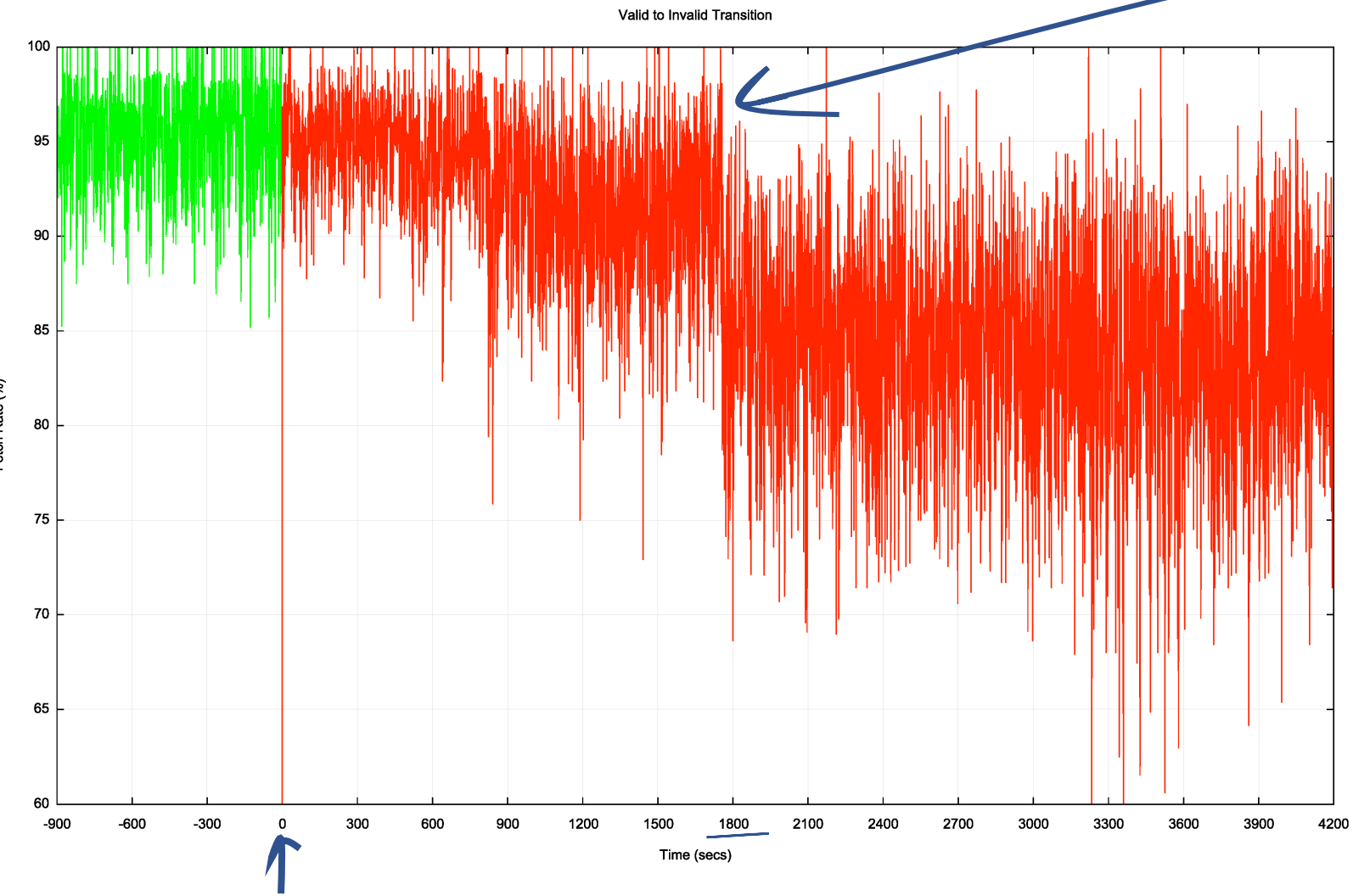


Weekly Measurement

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



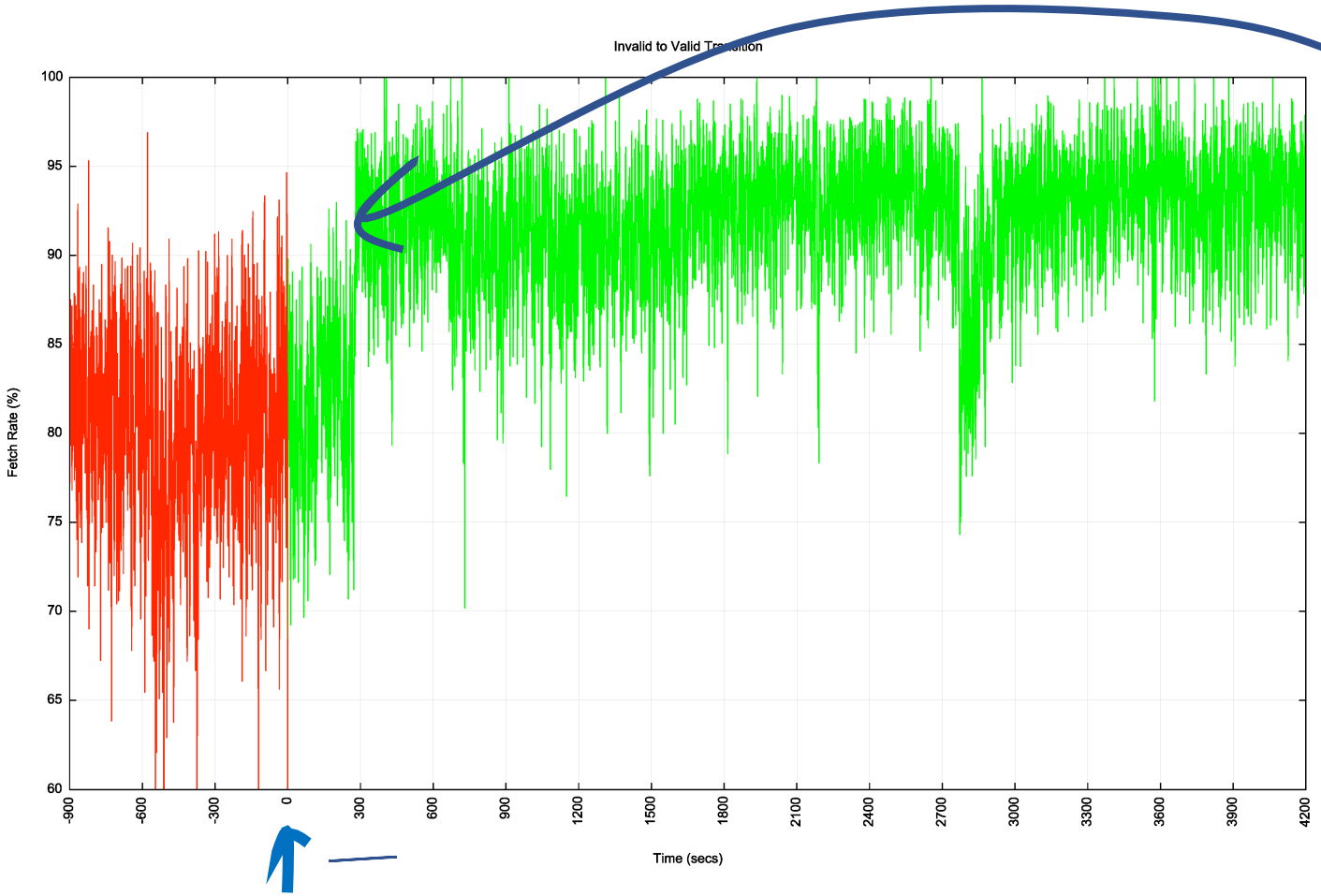
Time of ROA change at the RPKI repository

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

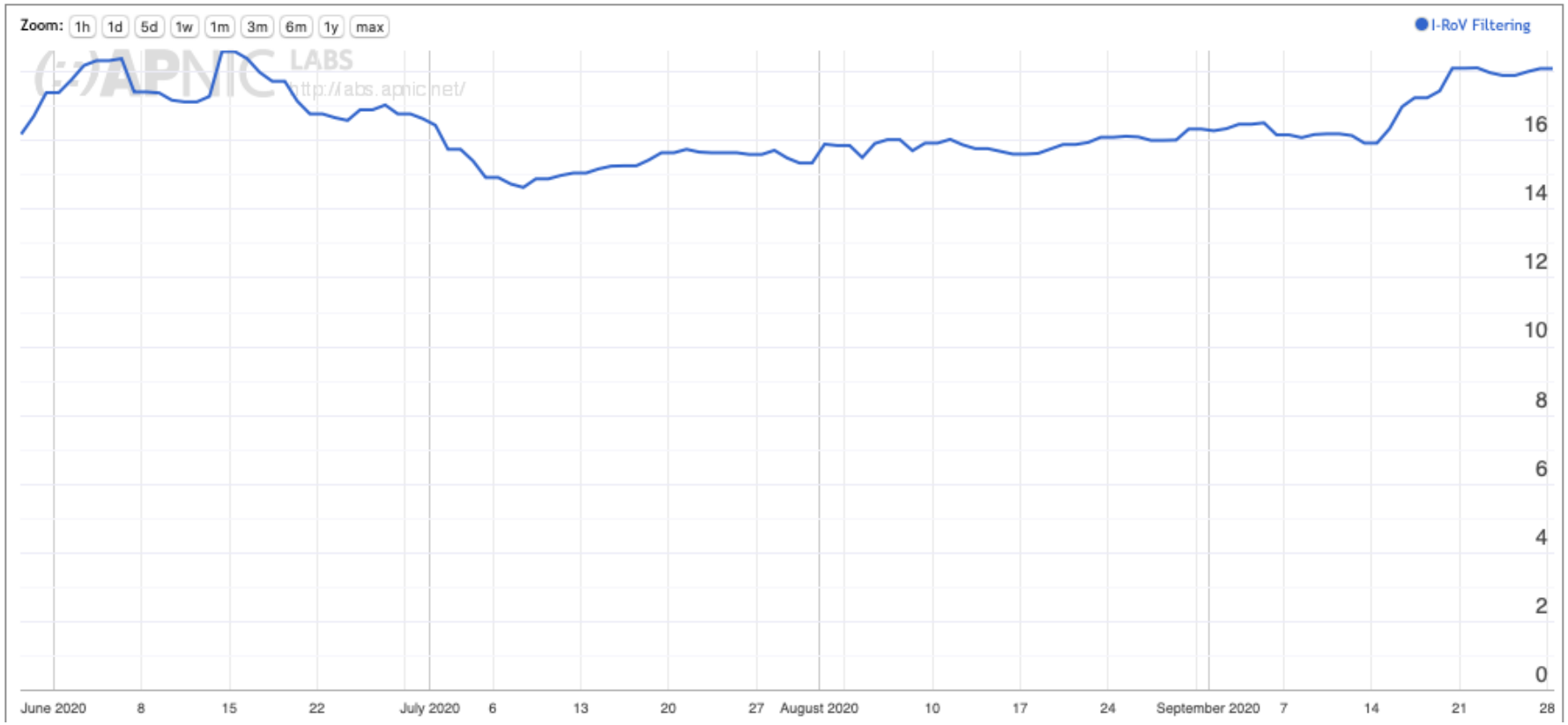
RPKI "sweep" software

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

I'd suggest that we use a 10 minute query timer as a compromise here

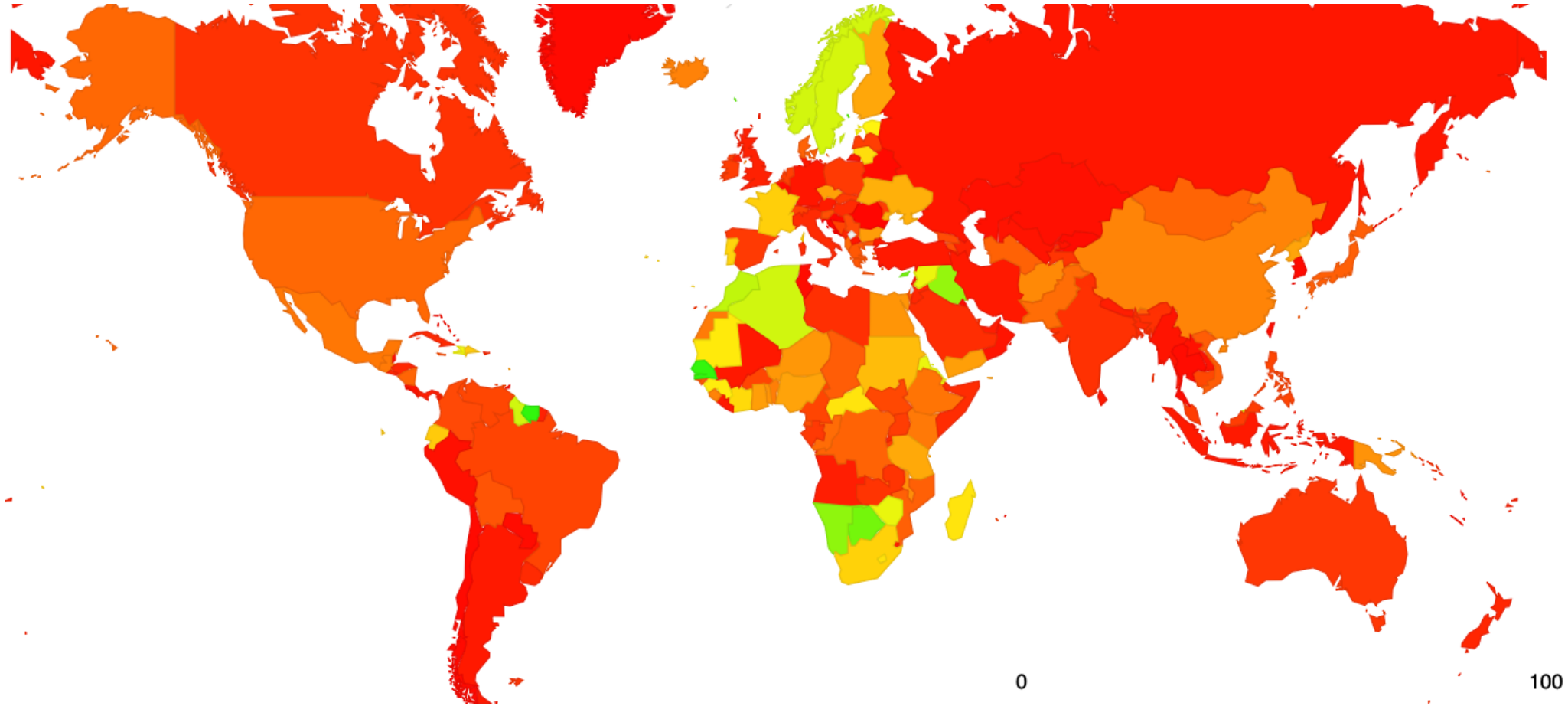
Results: User Impact of RPKI filtering

Use of RPKI Validation for World (XA)

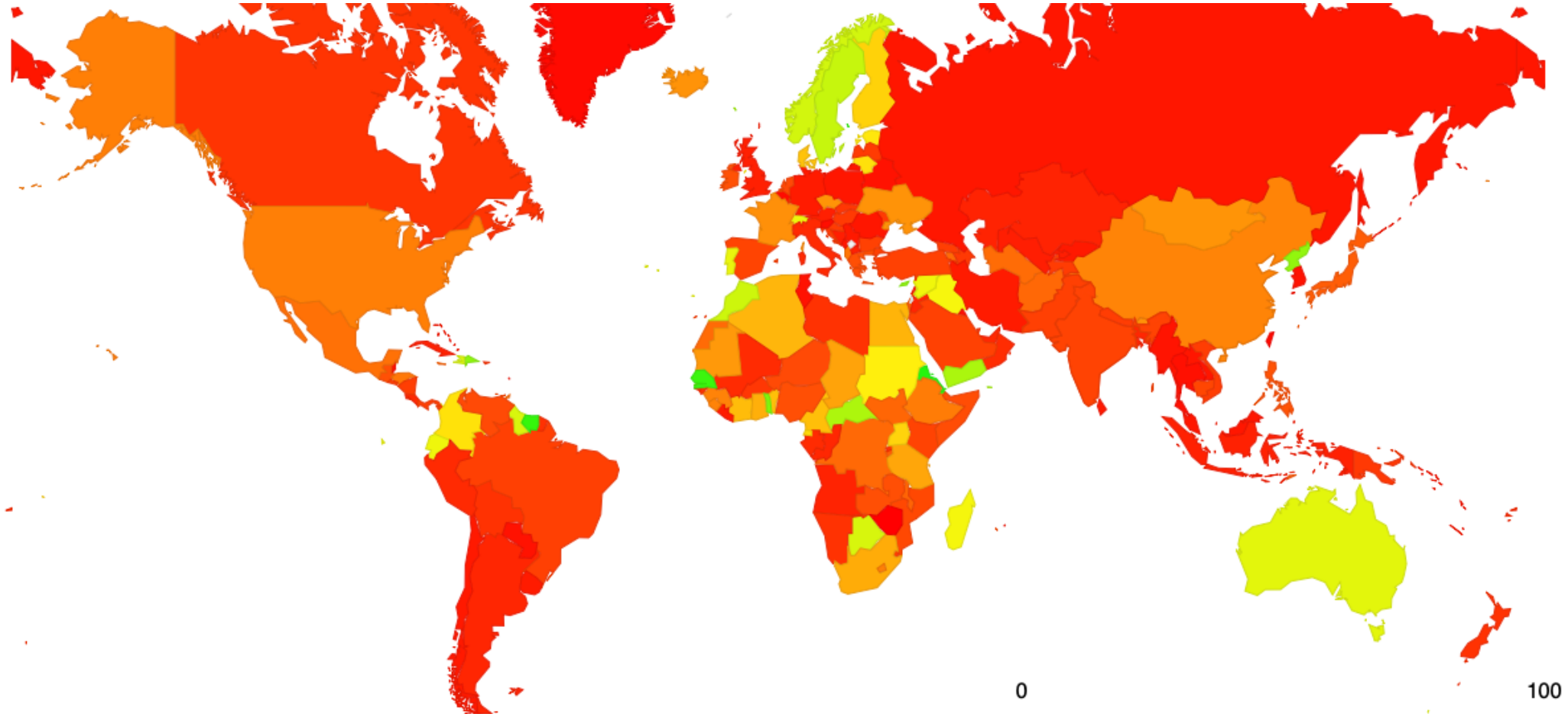


At 17% of users that's a surprisingly large impact for a very recent technology

Results: User Impact of RPKI filtering - Jul 2020



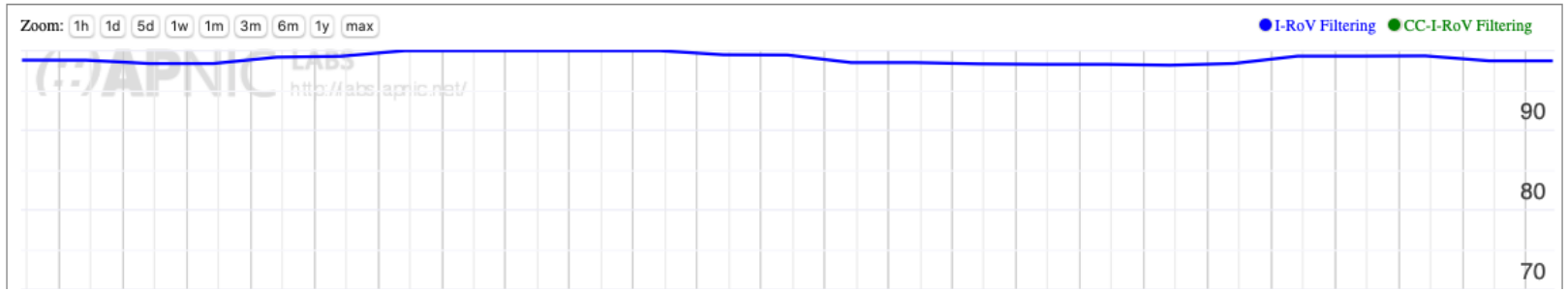
Results: User Impact of RPKI filtering - Oct 2020



Why?

- This map is a mix of two factors
  - Networks that perform invalid route filtering

RPKI I-ROV Per-Country filtering for AS37100: SEACOM-AS, South Africa (ZA)



RPKI I-ROV Per-Country filtering for AS7018: ATT-INTERNET4, United States of America (US)

Why?

- This map is a mix of two factors
  - 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 RoV objective is achieved, in that end users and their 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:

- Adding more anycast servers with more transit diversity

Next Steps for Measurement (2)

- Attempting 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

Next Steps for Measurement (3)

- 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?

Questions we might want to think about (3)

Prefix vs AS

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

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 a user protection measure then the issue of route filtering is potentially a more important issue for transits not stubs
  - A stub should generate ROAs for its routes, but there is far less of an incentive to perform RoV invalid filtering if all of the the stub's upstreams / IXs are already performing this filtering

Thanks!