Measuring the Effectiveness of Route Origin Validation Filtering using Invalid Drops from the perspective of the End User by using a Technique of Broad Scale Reachability Measurement

Geoff Huston
APNIC Labs

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 to bogus destinations?

Routing Security

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 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?

Routing Security

What's "the objective" of routing security?

- ❑Protect the routing system from all forms of oper aps?
- ❑Protect the routing system from som haps?
- ❑Protect the routing system
- ❑Protect the rou ostile attacks?
- ❑Prev address prefixes?
- bogus AS's in the routing system?
- all forms of synthetic routes from being injected into the routing system?
- ❑Prevent unauthorised route withdrawal?
- ☑Protect users from being directed to bogus destinations?

All these other objectives are fine, but the primary purpose of the routing system is to deliver packets to their intended destination

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

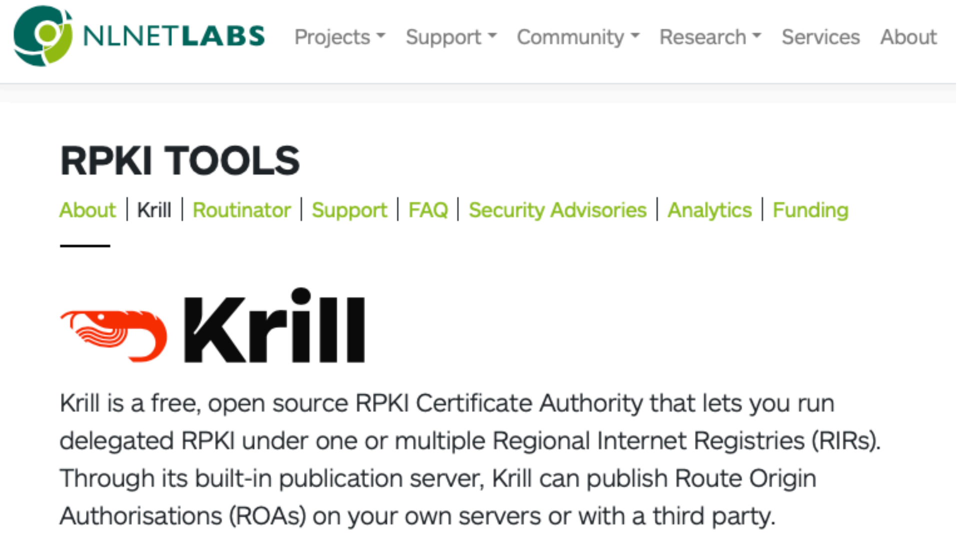
Measurement Approach

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



RPKI TOOLS

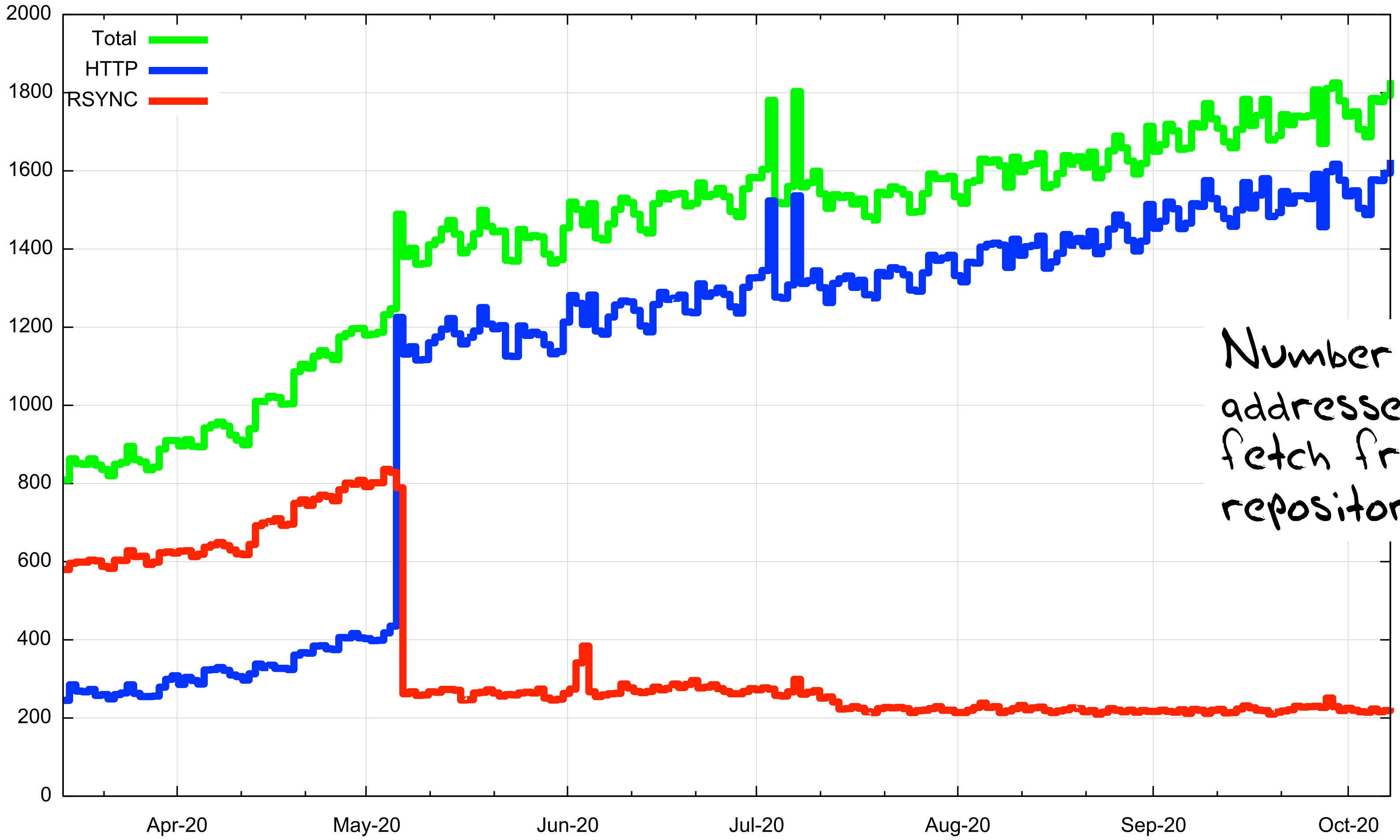
About | Krill | Routinator | Support | FAQ | Security Advisories | Analytics | Funding

Krill

Krill is a free, open source RPKI Certificate Authority that lets you run delegated RPKI under one or multiple Regional Internet Registries (RIRs). Through its built-in publication server, Krill can publish Route Origin Authorisations (ROAs) on your own servers or with a third party.

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 with diverse 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
- 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

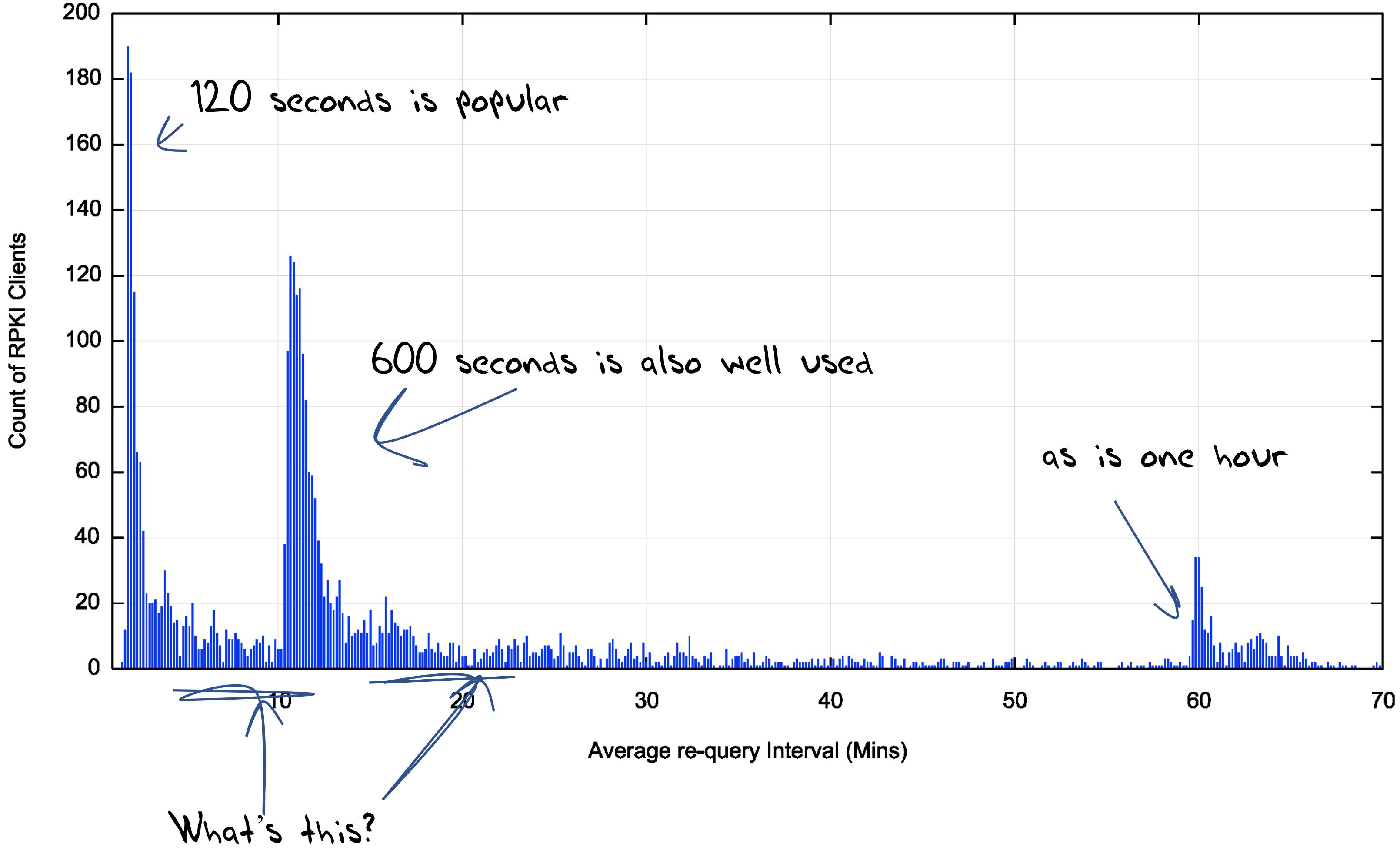
- Set a prefix and AS in a delegated RPKI repository
- Regularly revoke and re-issue ROAs that flip the validity state between valid and invalid states
- Anycast the prefix and AS pair in a number of locations across the Internet
- Load a unique URL that maps to the destination into a measurement script
- 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

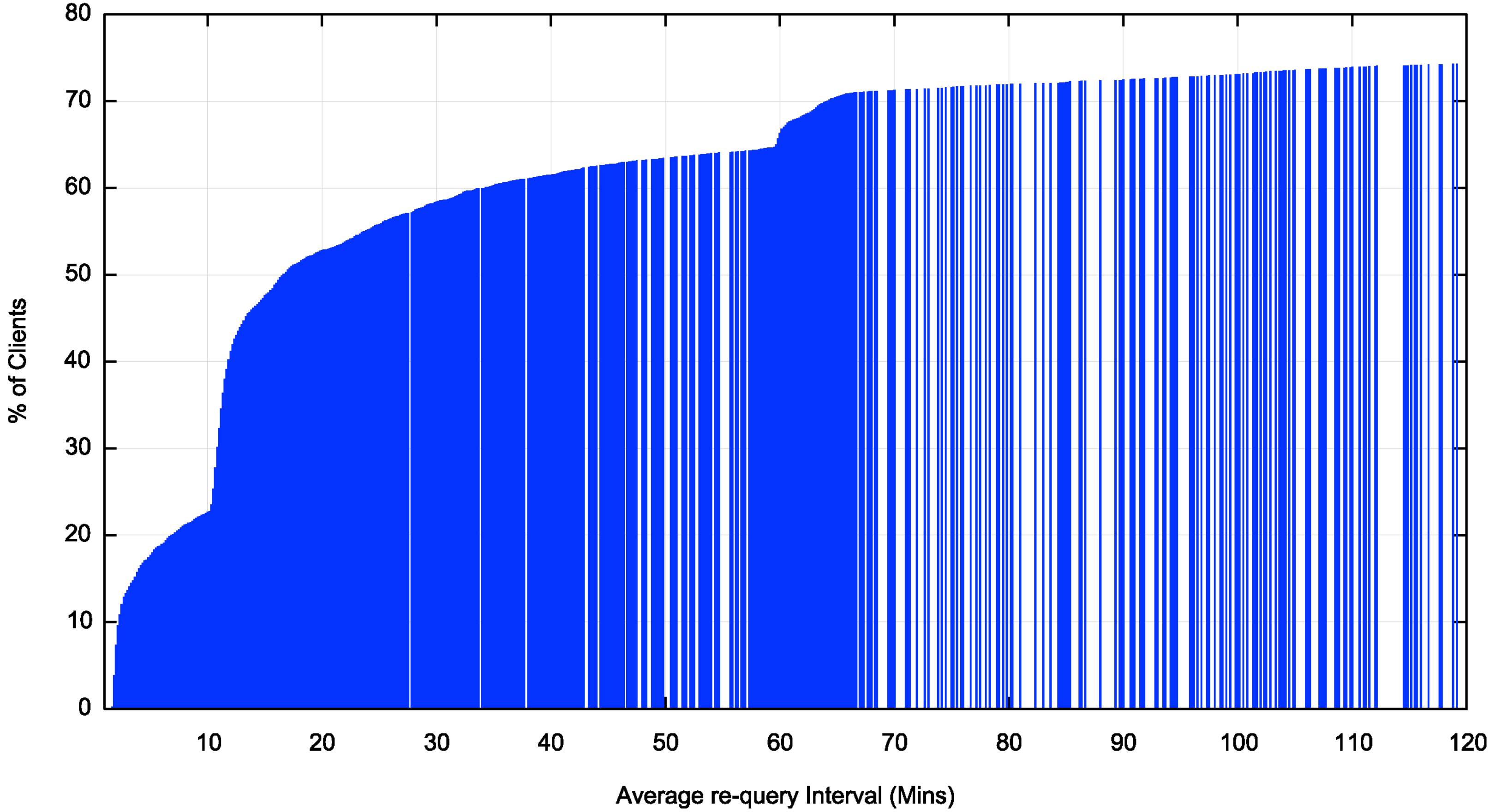
- What’s a good frequency to flip states?
  - How long does it take for the routing system as a whole to learn that a previously valid route is now invalid? And how long for the inverse invalid to valid transition
- Validity / Invalidity is determined by what is published at the RPKI publication point
  - 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 Pub Point Re-Query Intervals

We are looking here at the average elapsed time between successive visits to the RPKI publication point server (krill logs

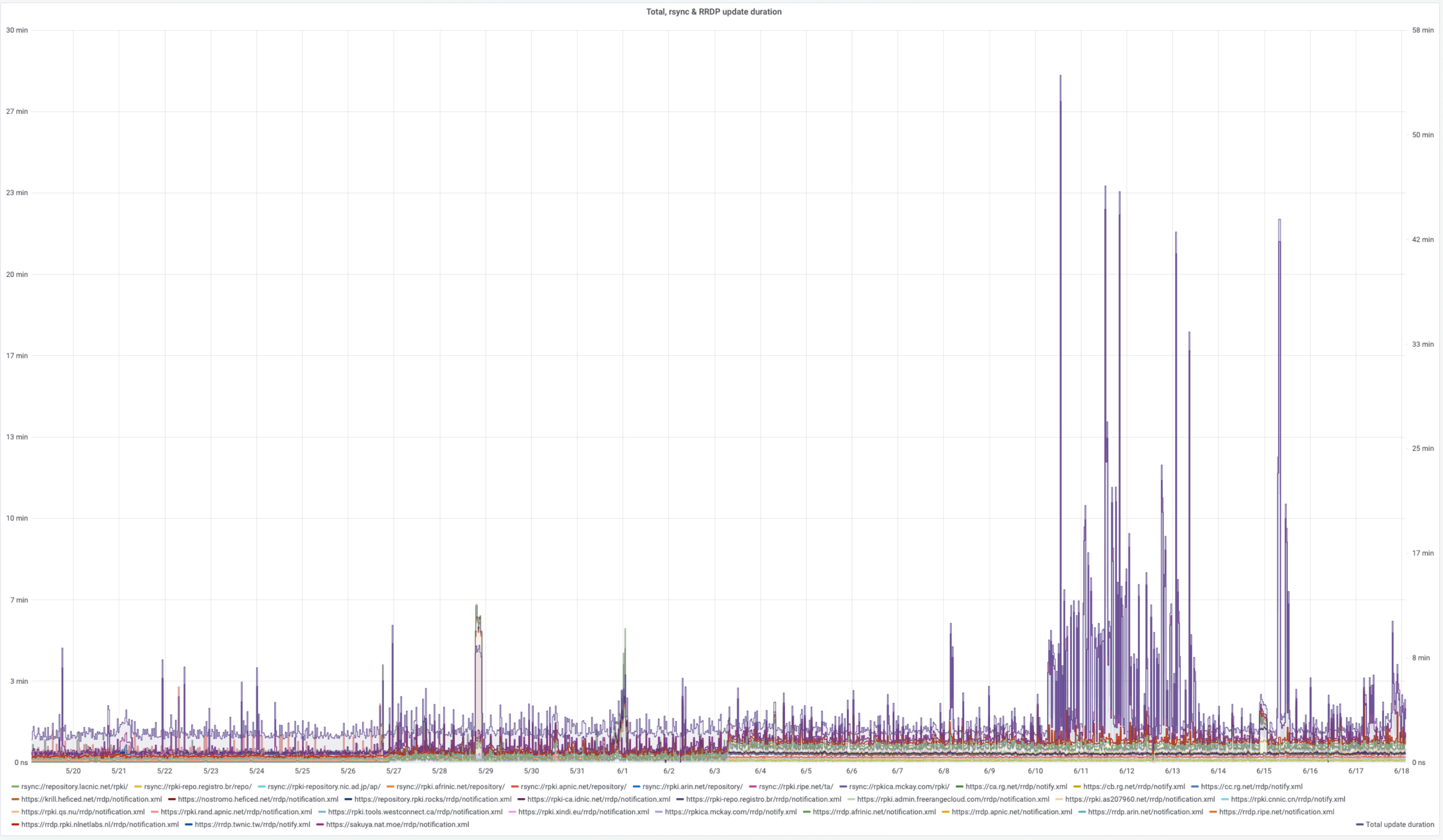


Re-Query - Cumulative Distribution



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

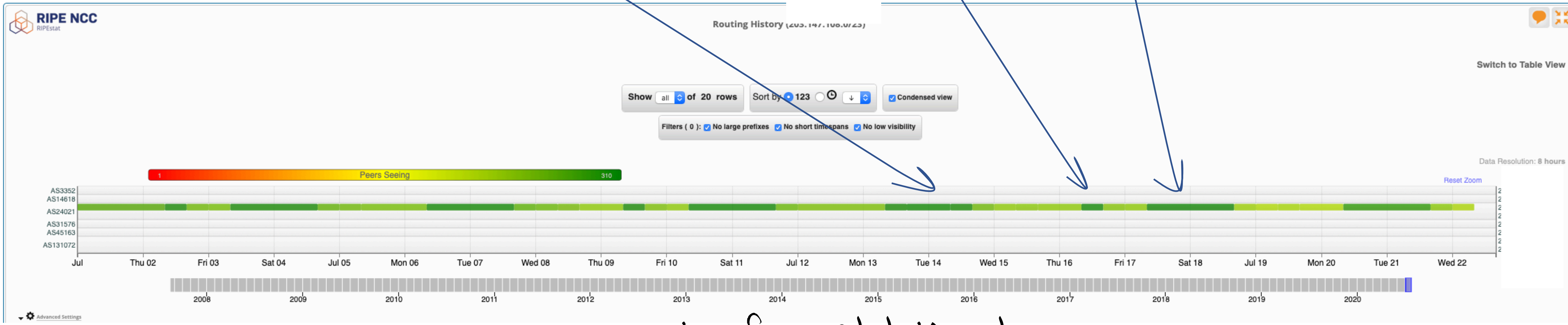
We used 12 and 36 hour held states

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

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

We used 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



RIPE NCC
RIPEstat

Routing History (203.147.108.0/23)

Switch to Table View

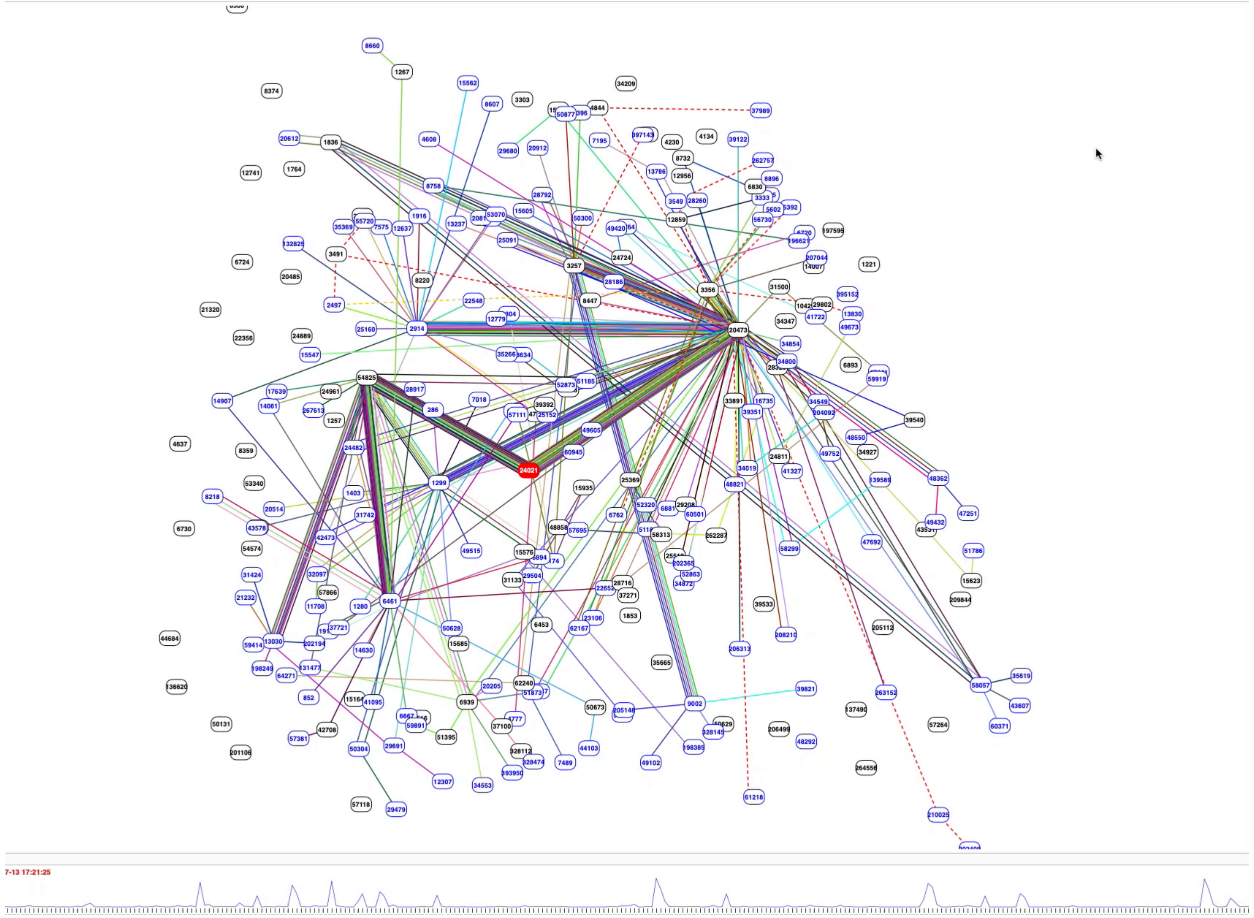
Show all of 20 rows | Sort by ● 123 ○ | ☑ Condensed view

Filters (0): ☑ No large prefixes ☑ No short timespans ☑ No low visibility

Data Resolution: 8 hours

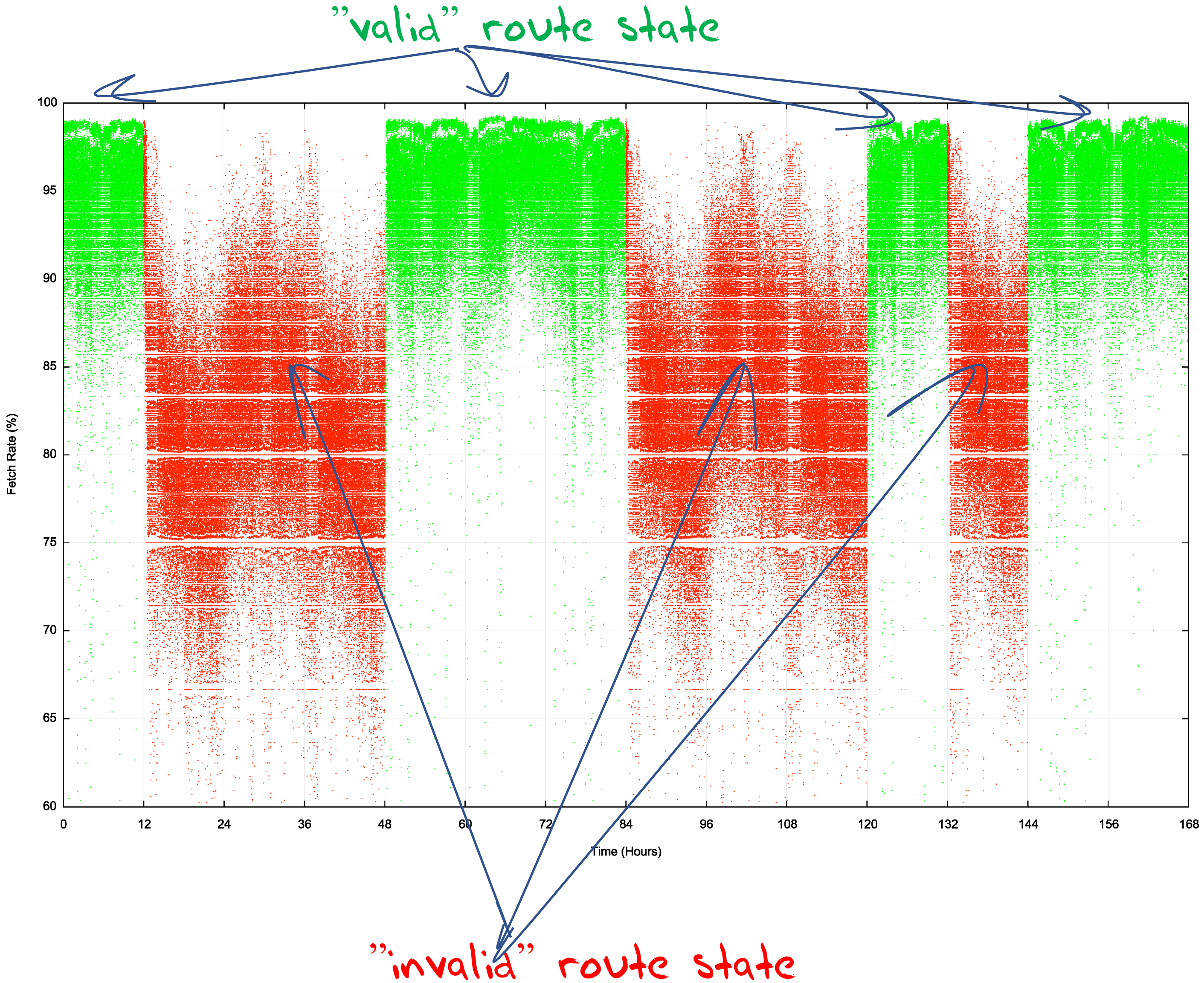
view from stat.ripe.net

We used 12 and 36 hour held states



BGP Play view of the routing changes

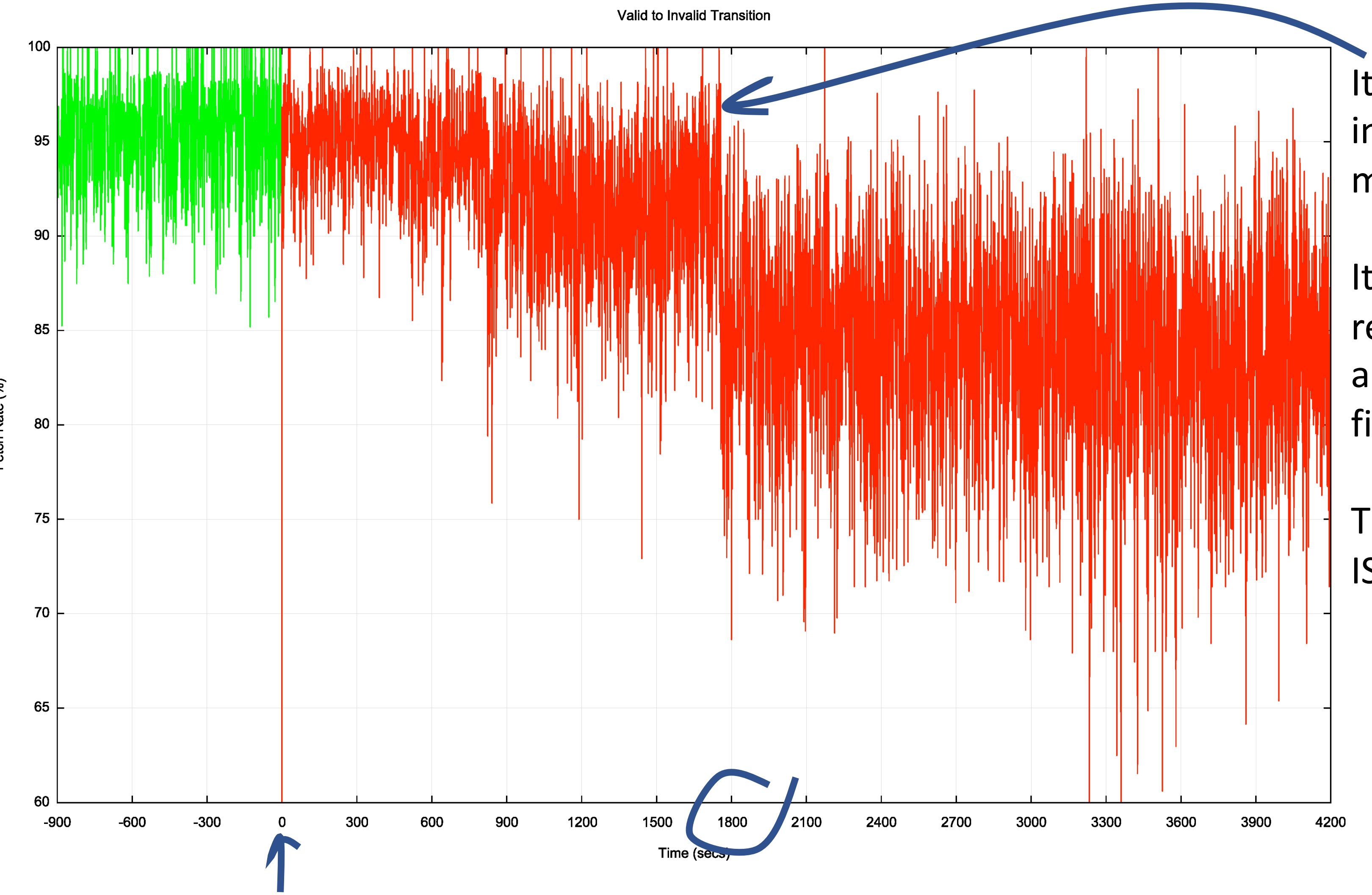
We used 12 and 36 hour states



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



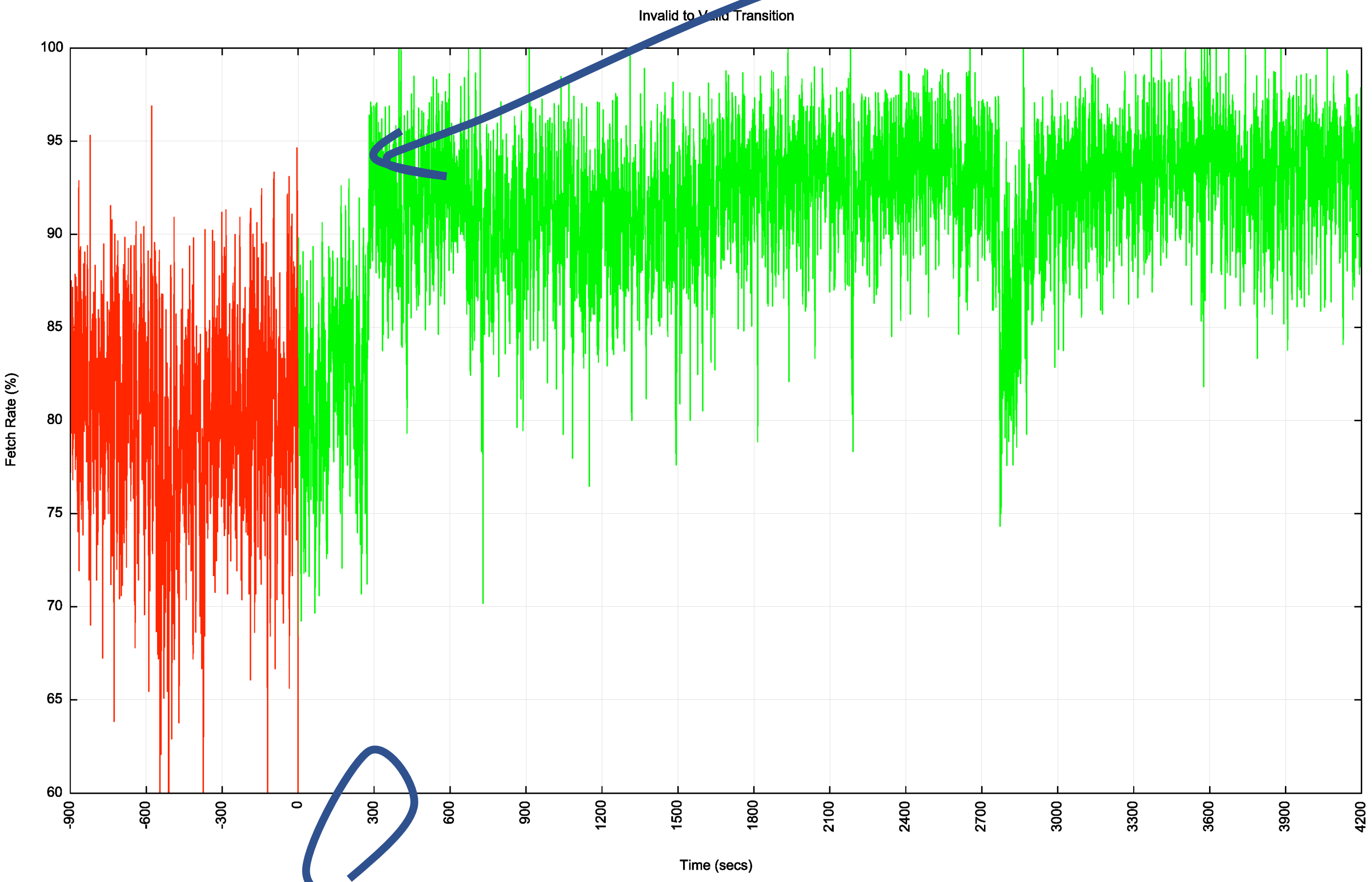
Time of ROA change at the RPKi repository

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



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

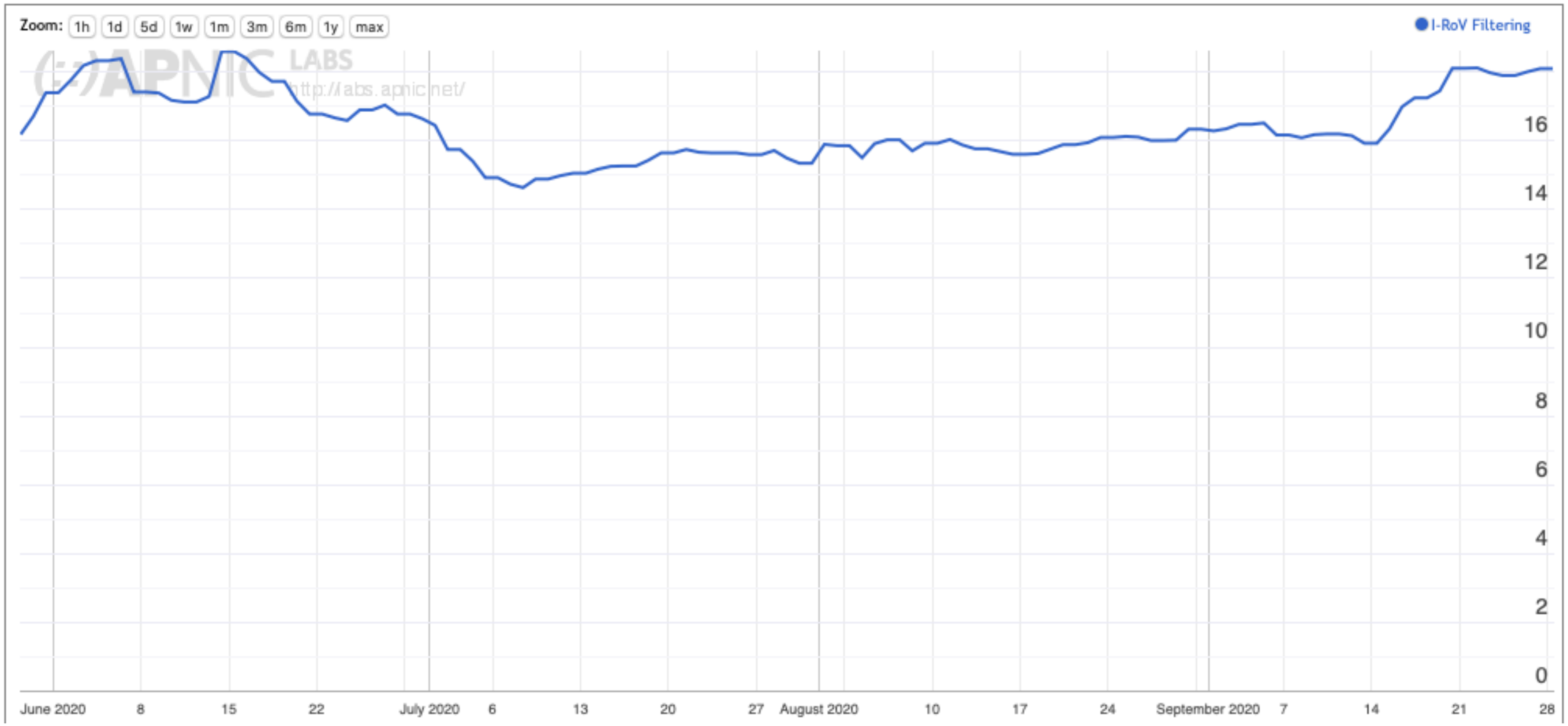
Time of ROA change at the RPKi repository

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

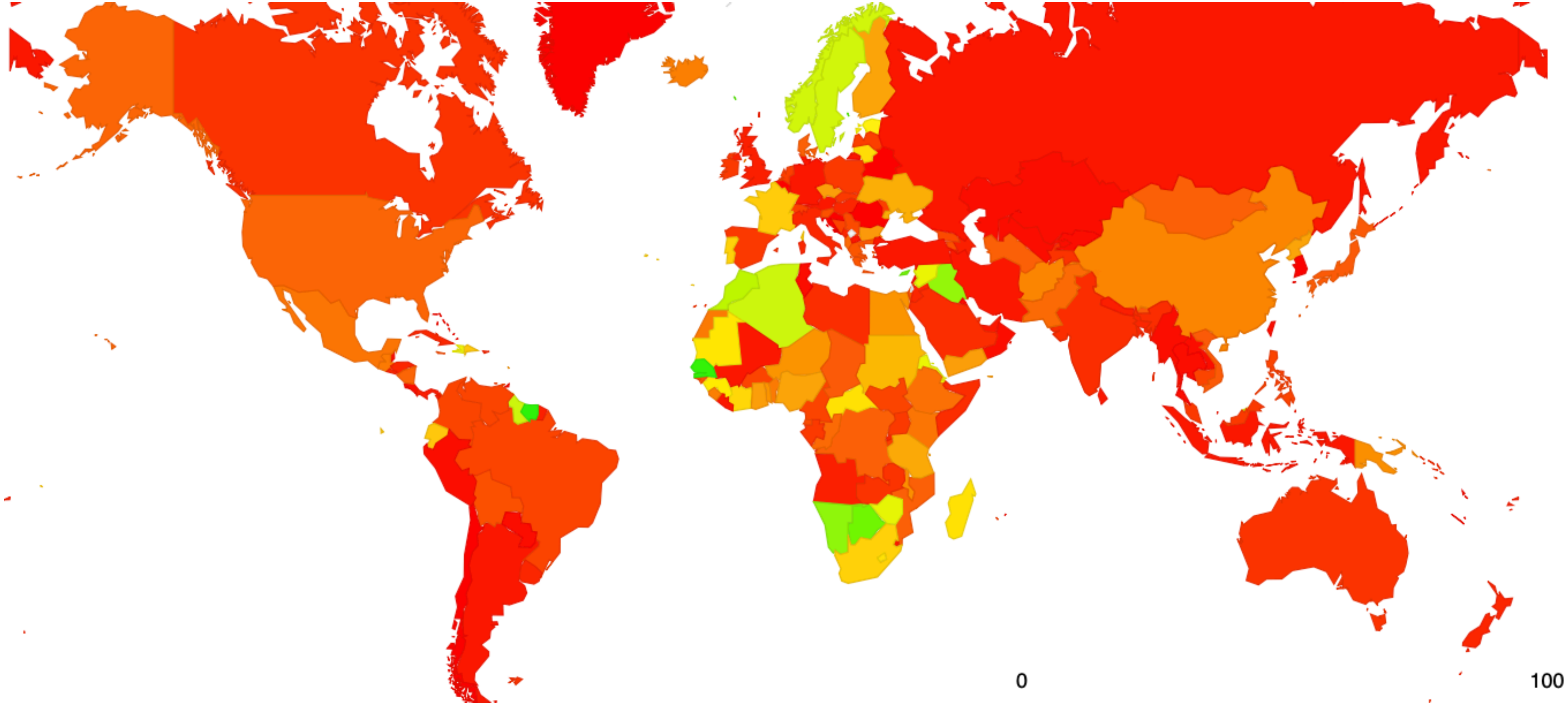
Use of RPKI Validation for World (XA)



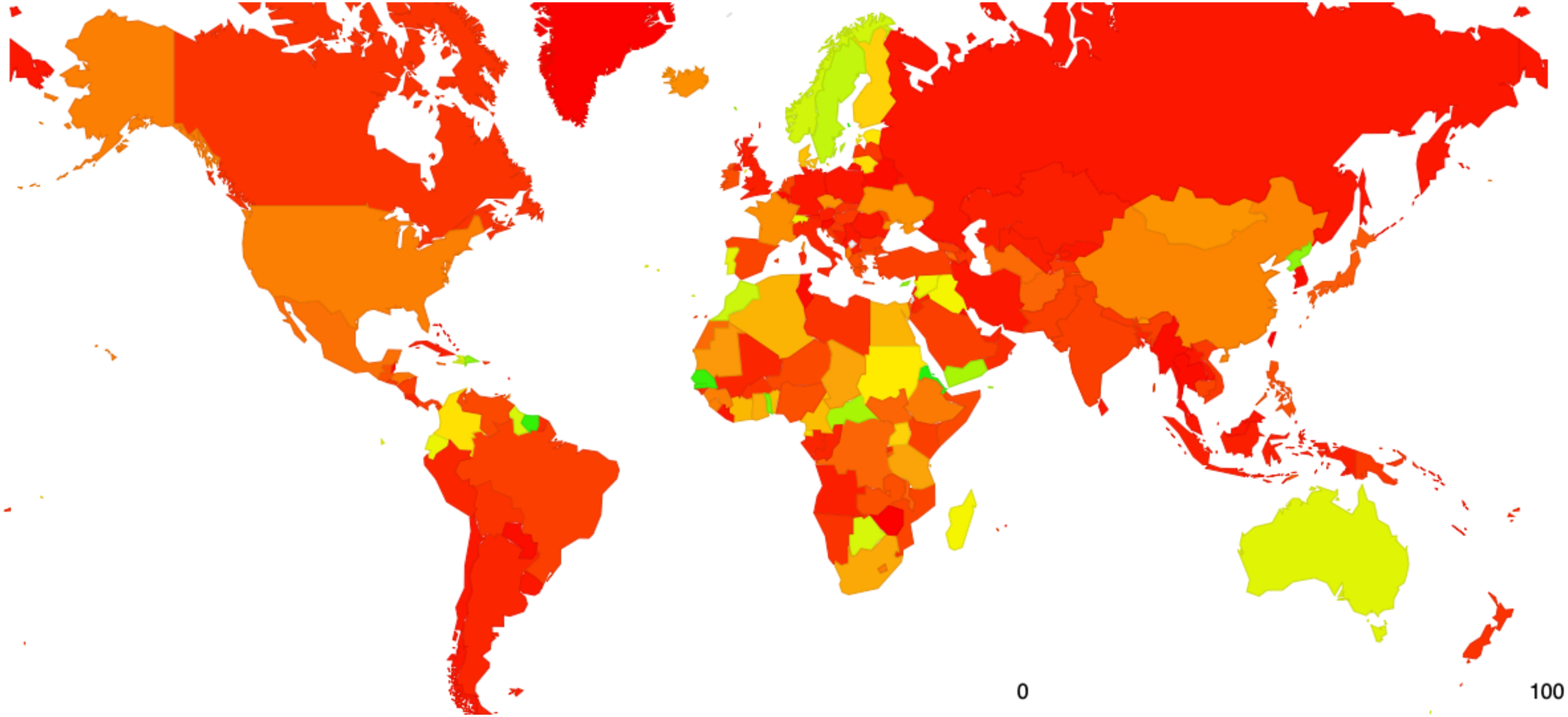
At 17% of users that’s a surprisingly large impact for a very recent technology

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

Results: User Impact of RPKI filtering - Jul 2020



Results: User Impact of RPKI filtering - Oct 2020

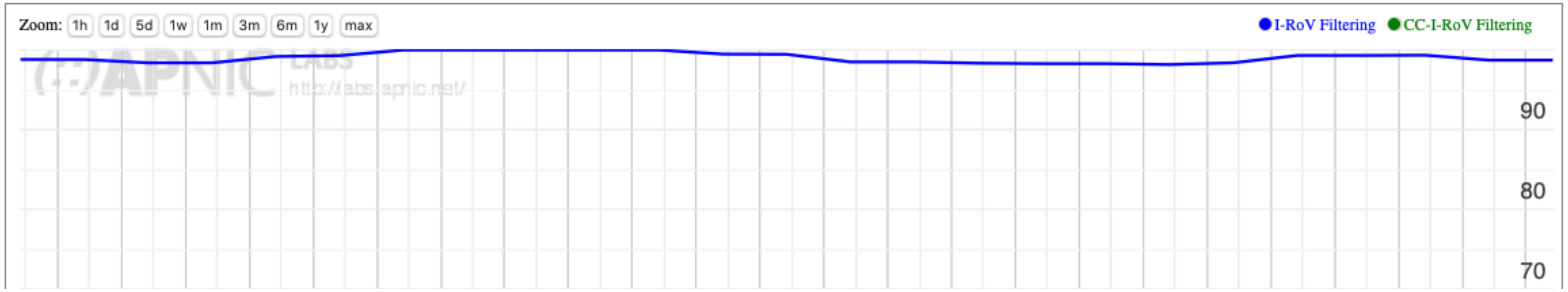


Why?

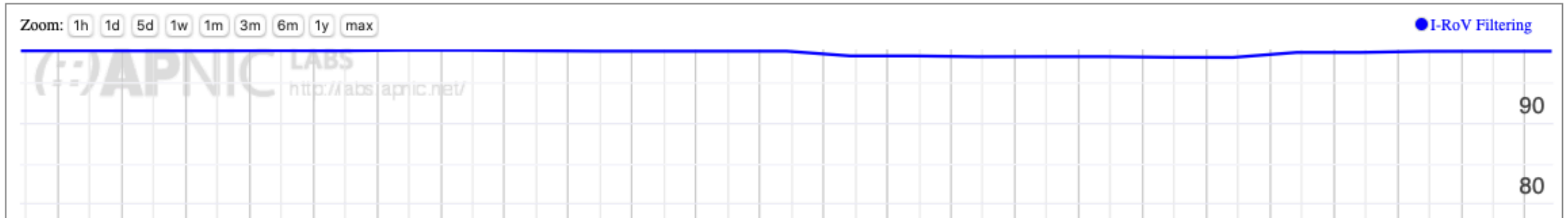
This map is a mix of two factors

- 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

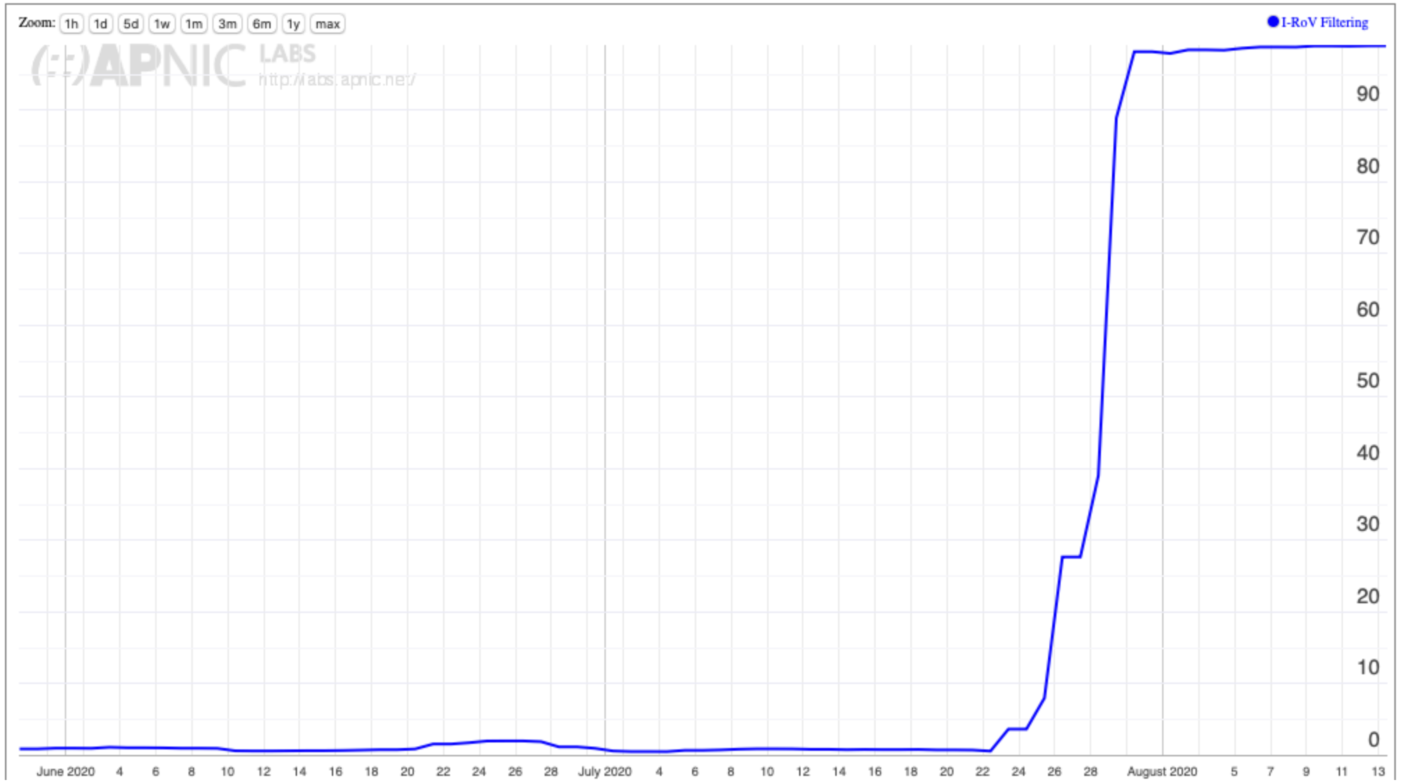
and

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

In either case the basic RPKI RoV objective is achieved, in that end users and their networks are not exposed to invalid route objects

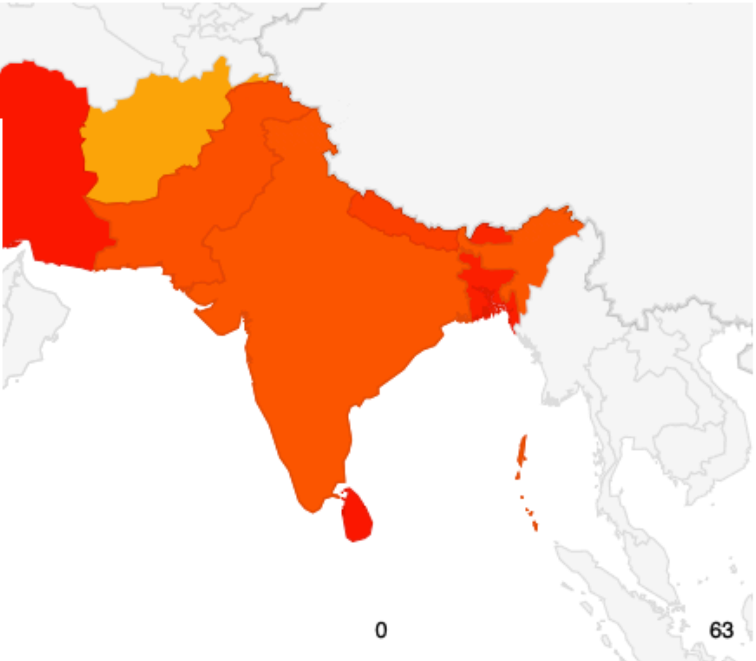
Recent Deployment of RPKI Drop Invalids in AS1221

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

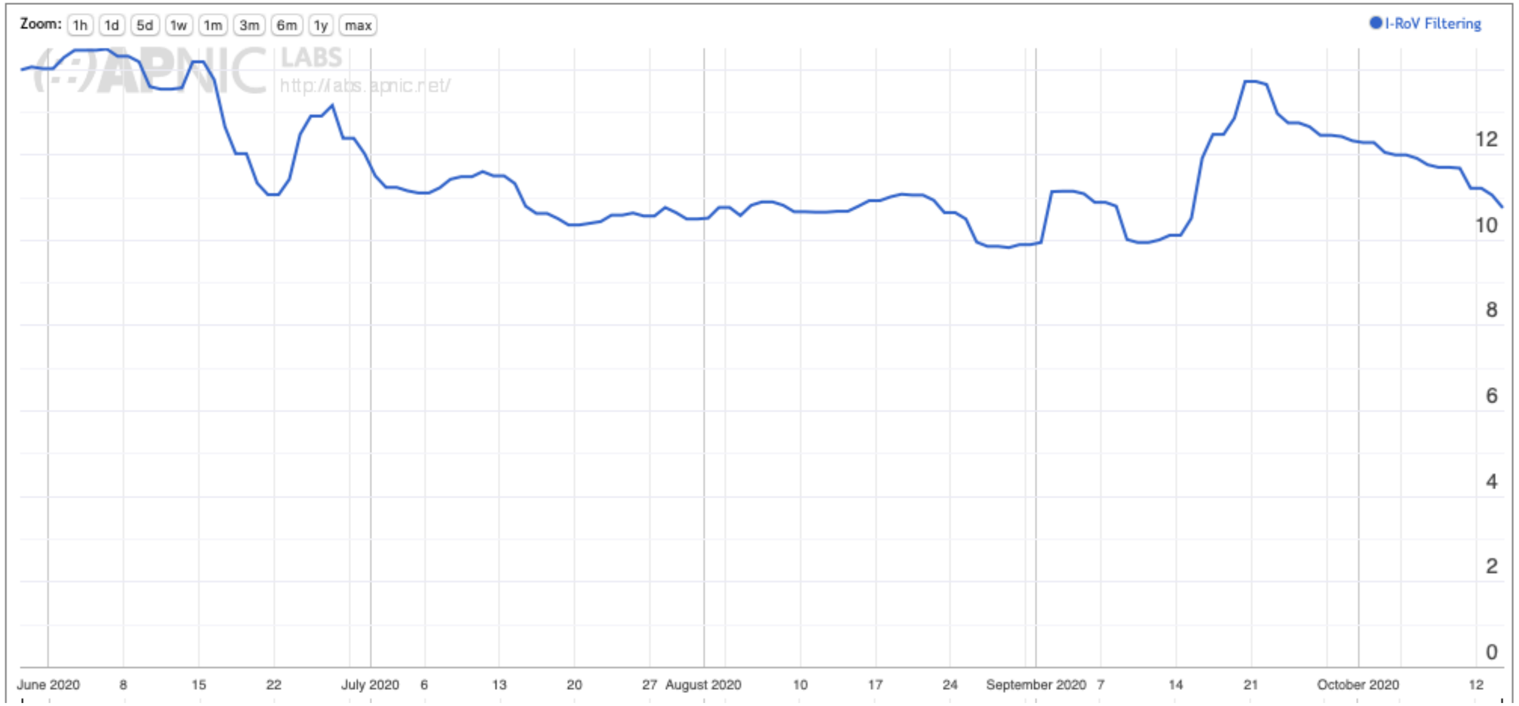


What about the Local Region?

Region Map for Southern Asia (034)

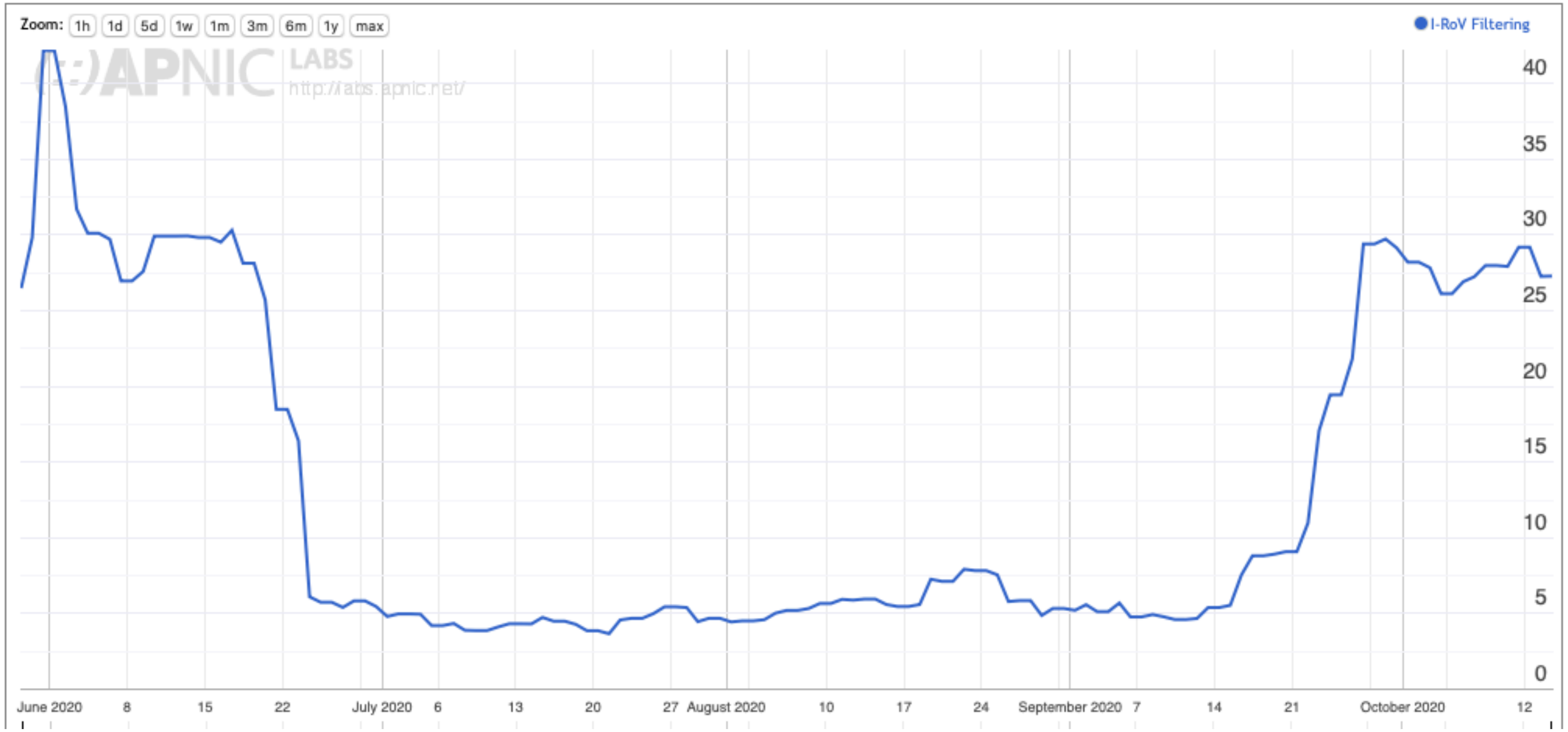


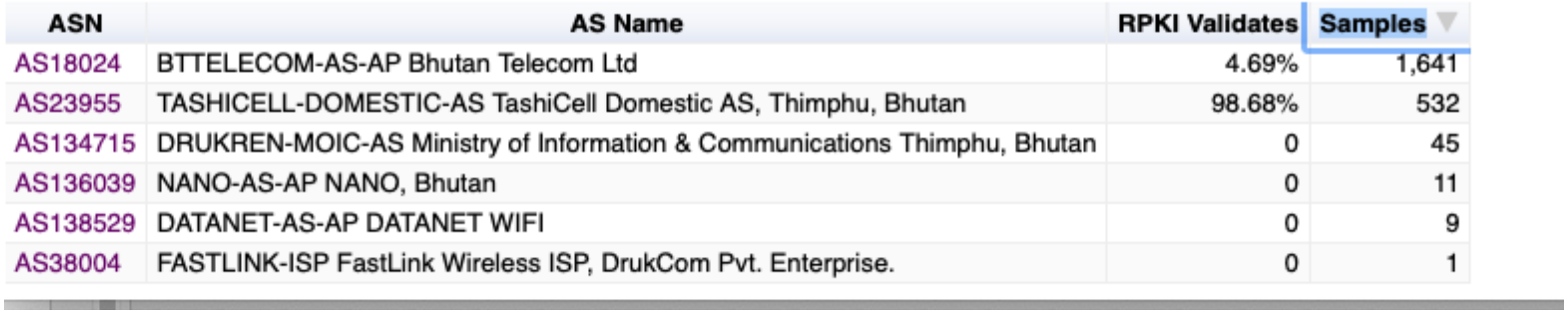
Use of RPKI Validation for Southern Asia (XT)



Bhutan

Use of RPKI Validation for Bhutan (BT)





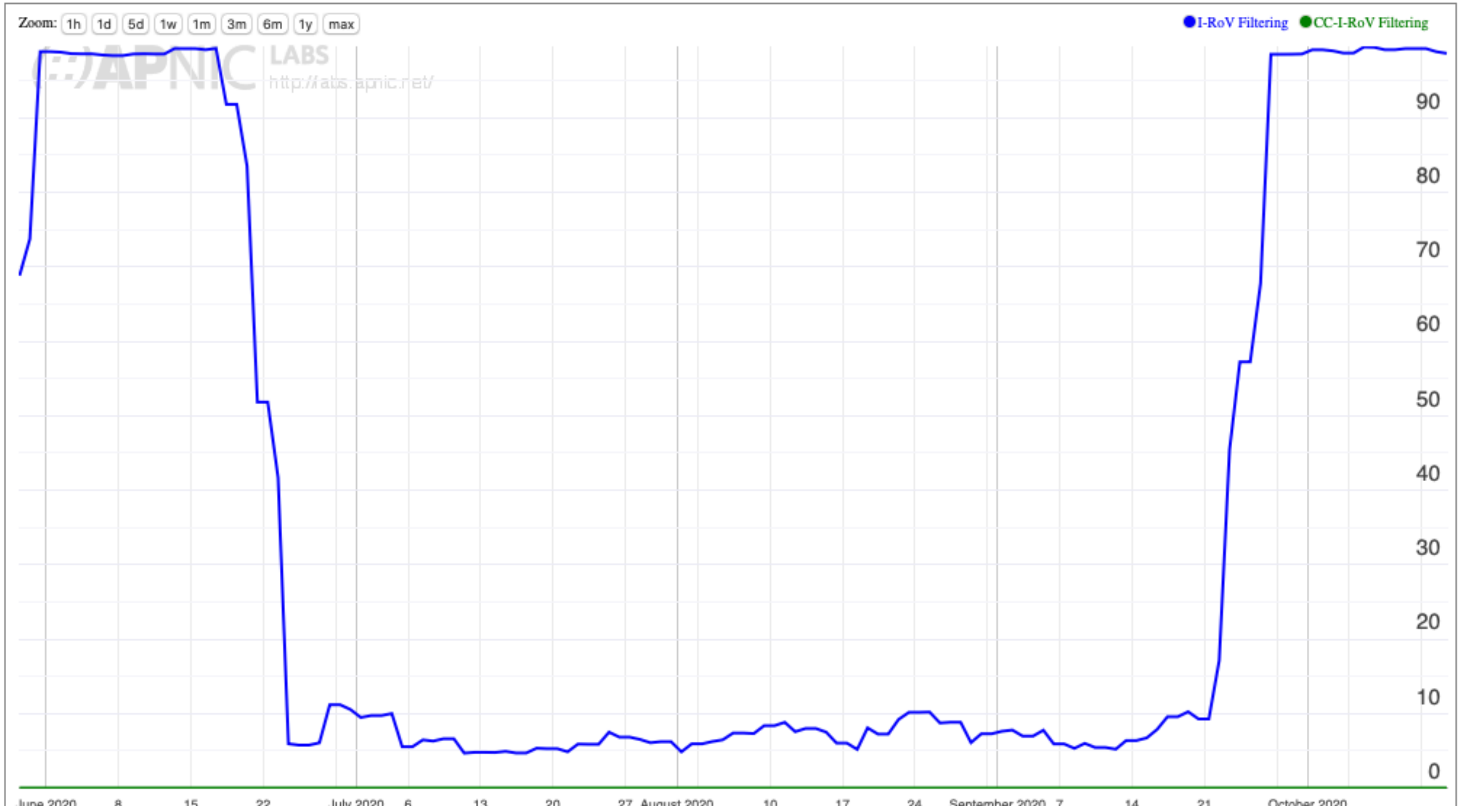
ASN	AS Name	RPKI Validates	Samples
AS18024	BTTELECOM-AS-AP Bhutan Telecom Ltd	4.69%	1,641
AS23955	TASHICELL-DOMESTIC-AS TashiCell Domestic AS, Thimphu, Bhutan	98.68%	532
AS134715	DRUKREN-MOIC-AS Ministry of Information & Communications Thimphu, Bhutan	0	45
AS136039	NANO-AS-AP NANO, Bhutan	0	11
AS138529	DATANET-AS-AP DATANET WIFI	0	9
AS38004	FASTLINK-ISP FastLink Wireless ISP, DrukCom Pvt. Enterprise.	0	1

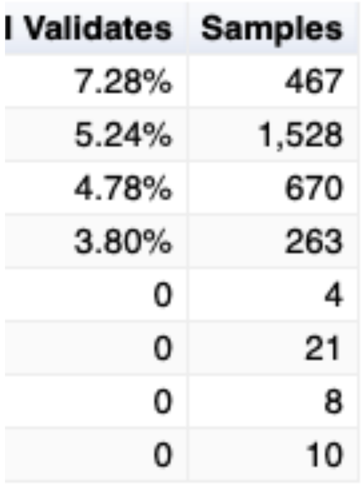
Bhutan?

Use of RPKI Validation for Bhutan (BT)

Zoom: 1h 1d 5d 1w 1m 3m 6m 1y max
I-RoV Filtering : 4.33 | 10:00 July 12, 2020
40
APNIC LABS
http://labs.apnic.net/

RPKI I-ROV Per-Country filtering for AS23955: TASHICELL-DOMESTIC-AS TashiCell Domestic AS, Thimphu, Bhutan, Bhutan (BT)





l Validates	Samples
7.28%	467
5.24%	1,528
4.78%	670
3.80%	263
0	4
0	21
0	8
0	10

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!