A Few Years In The Life Of An RPKI Validator

Rob Austein <sra@hactrn.net>
Randy Bush <randy@psg.com>
Michael Elkins <michael.elkins@parsons.com>
. . . and a lot of help from our friends

IETF 94
Yokohama
November 2015

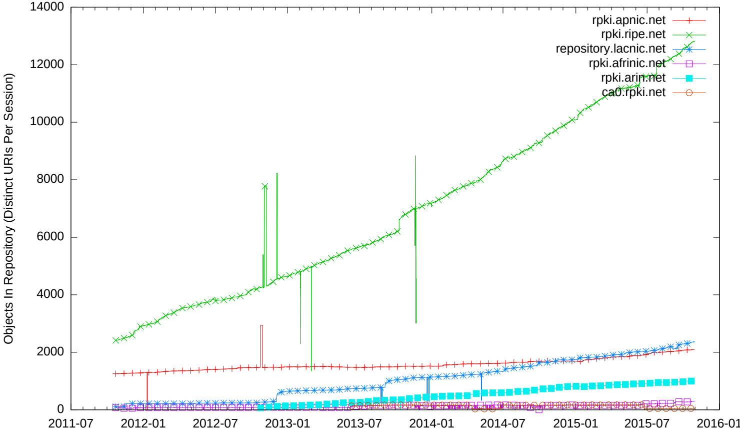
The World As Seen By One RPKI Validator

- Data as logged by one validator in Seattle.
- Data collection started late October 2011.
- Guilty parties are good people, all friends here.
- Expect updated report(s) at later date(s).

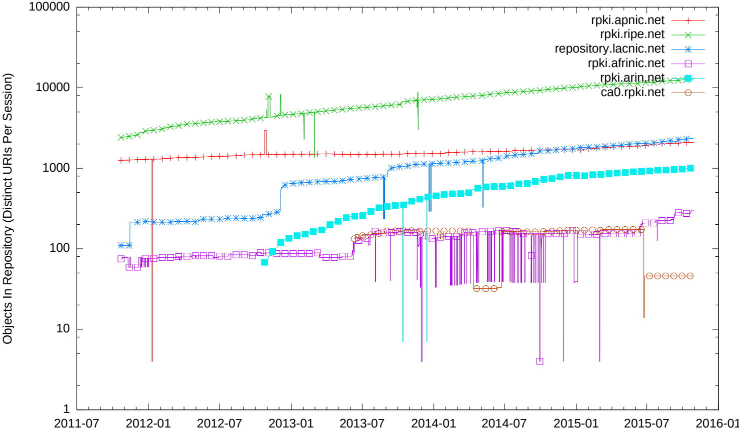
A Brief Overview of RPKI Validation

- Distributed global database of X.509 certificates and dependent objects.
- The X.509 certificates contain `rsync://` URIs.
- Validation starts at trust anchor(s).
- Validator walks certificate tree, following URIs.
- rcynic is one such validator.
- rcynic is session-oriented (cron job).
- Measurements to date are only for rsync, not RRDP.

Object Counts (Linear)



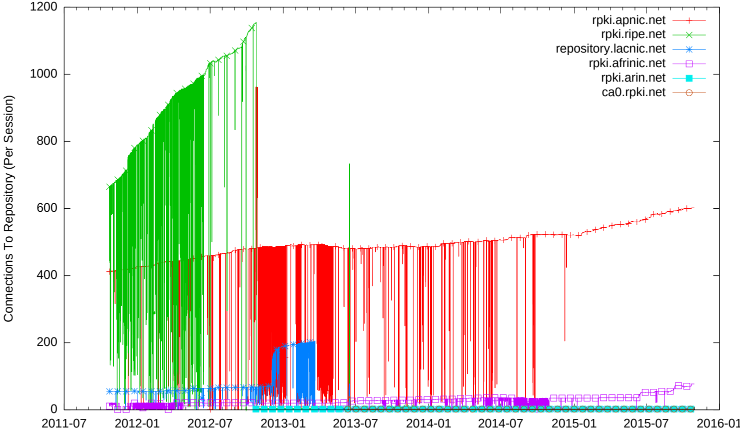
Object Counts (Logarithmic)



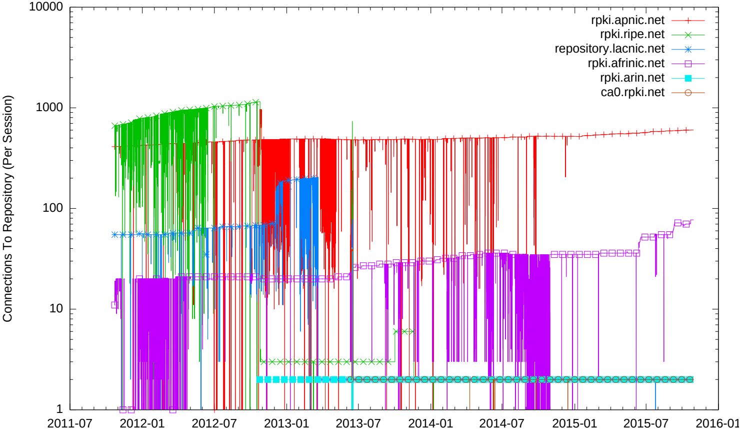
Object Counts: Observations

- Large downward spikes are either genuine mass extinction events or, more likely, validation failure of a high-level certificate causing a large subtree to go invalid. Either way, these usually indicate Something Very Bad.
- . . . Or a mess being cleaned up.

Connection Counts (Linear)



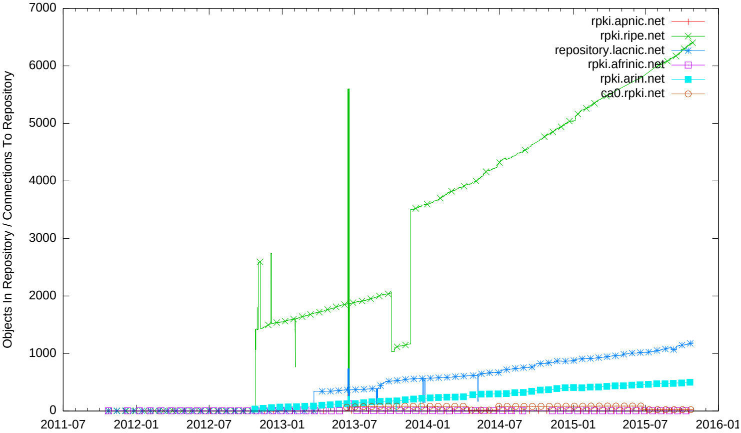
Connection Counts (Logarithmic)



Connection Counts: Observations

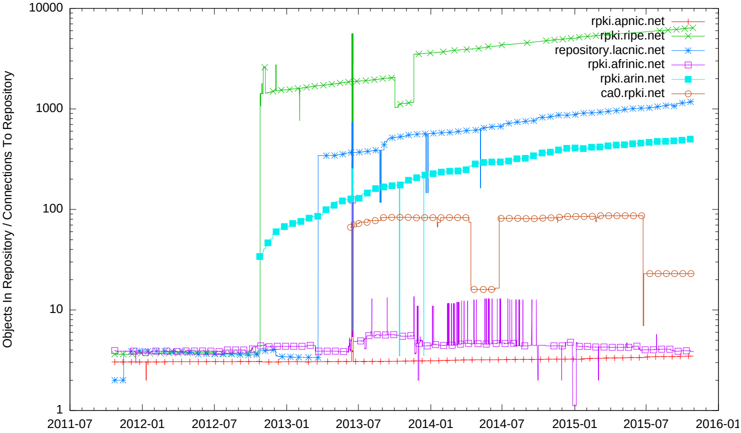
- Repeated downward spikes are connection failures or misinterpreted rsync exit codes.
- “Connection failures” may be server problems, *e.g.*, the grouping of `rpki.ripe.net` failures in early 2012 turned out to be a mis-configured HA cluster.
- Note massive drop in connection count when RIPE reconfigured from flat to hierarchical publication in late 2012.
- LACNIC appears to have made the same transition to hierarchical publication in early 2013.
- Entire hierarchical publication issue (probably) becomes irrelevant if and when we all move to RRDP.

Objects/Connection (Linear)



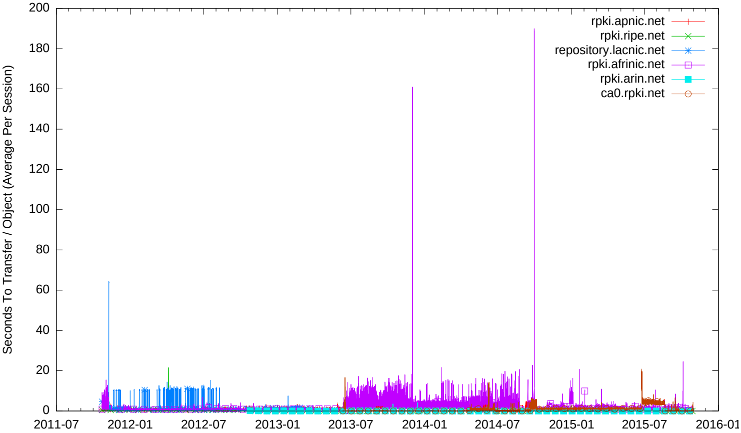
(Sessions with connection failures not shown)

Objects/Connection (Logarithmic)



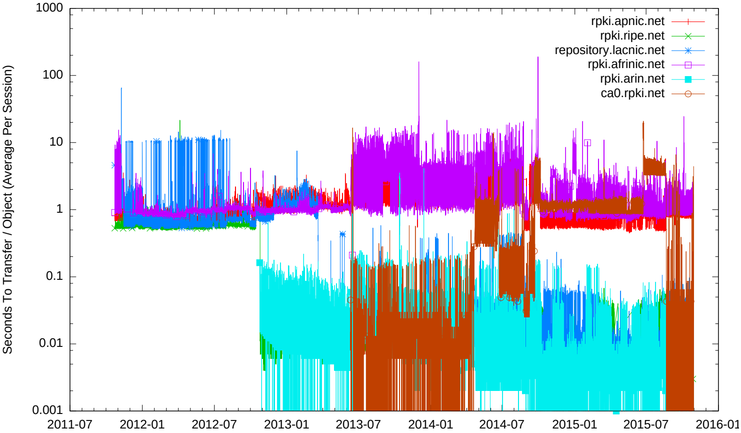
(Sessions with connection failures not shown)

Seconds/Object (Linear)



(Sessions with connection failures not shown)

Seconds/Object (Logarithmic)

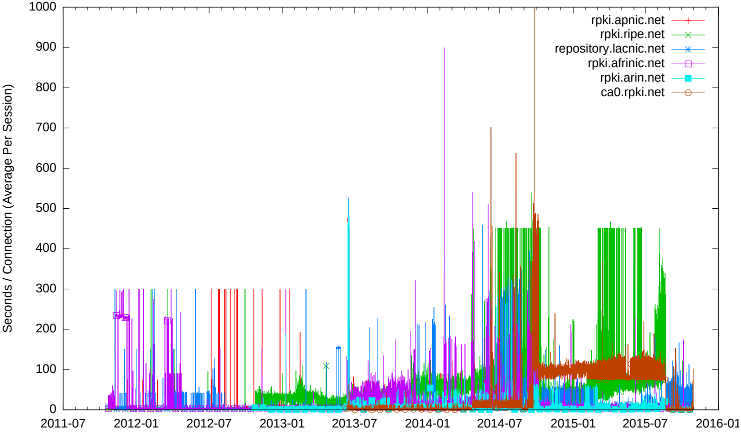


(Sessions with connection failures not shown)

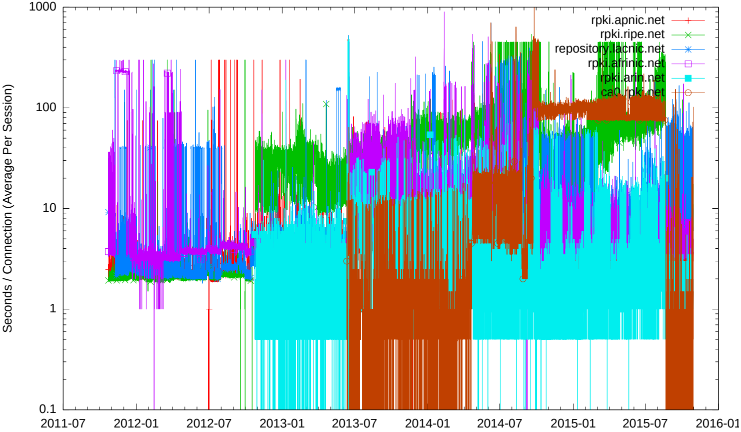
Seconds/Object: Observations

- "Elapsed time" as reported here is sum of parallel connection times—five parallel connections of four minutes each counts as twenty minutes.
- We can speed up in terms of wall time by running more connections in parallel, but that puts more load on the repository servers and risks rate limiting.
- Spikes here are slow repository servers; whether it's the network path or the server itself that's slow, we don't know.
- Note drop in seconds/object when RIPE and LACNIC go to hierarchical publication.
- RRDP with caching infrastructure would (probably) be a very different picture (but that's prediction, not measurement).

Average Connection Duration (Linear)



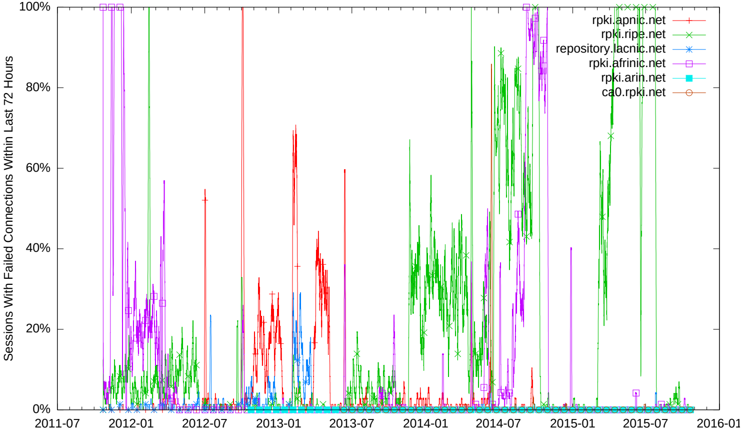
Average Connection Duration (Logarithmic)



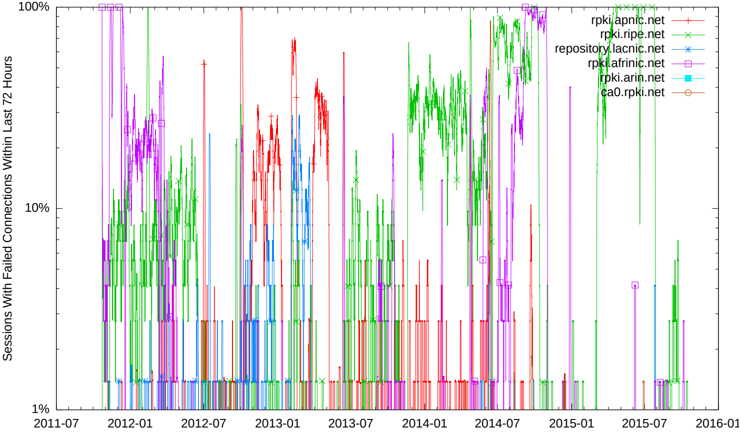
Average Connection Duration: Observations

- Early modeling and testing said rsync setup/teardown cost of about 500ms tends to dominate for large numbers of rsync connections. This analysis still seems to hold up.
- Average connection times go up with transition to hierarchical publication (smaller number of connections, but more happening per connection). More efficient, but at a cost.
- Spikes used to top out at 300 seconds because that's when rcynic whacks stalled rsync.
- With large numbers of objects per connection, it's common to see longer connection times and rsync processes still doing useful work after 300 seconds, particularly with RIPE.
- Don't really know why we see such wide range of connection times for RIPE, might be network issues, might be some interaction between our hourly polling cycle and their data refresh cycle.

Failure Rate (Linear)



Failure Rate (Logarithmic)



Failure Rate: Observations

- Most interesting thing to note here is just that the recent wide range of connection times with RIPE seem to corollate with frequent connection failures.
- Failure rate is a bit hard to measure because:
  - We give up on a repository host for the duration of that session after the first failure.
  - rsync exit codes often don't tell us much we can use, so we can't really tell the difference between TCP reset, incorrect SIA caRepository causing us to poll a nonexistant URI, and NFS failure within server's HA cluster: all look like "rsync exit code #23: Partial transfer due to error."
- So shape of the curve is significant: a brief spike from 0% to 100% is probably a data error, while a failure rate that stays high or wanders all over the map is probably a network or server issue.

Things We're Not Measuring Yet?

Freshness: Some kind of measure of whether we're keeping up with what's being published, regardless of how we do it or how much pain is involved. One could make a case that this is the critical measurement and that all else is just dickering over the price.

RRDP: Too early in development and deployment cycle for any useful RRDP measurements, but clearly we'll want to track this.

What else?

Questions?