RSSAC047 in Focus:

A closer look at reachability and publication delay

In collaboration with Verisign and ISC

Consortium:

Stichting Internet Domeinregistratie Nederland (SIDN)

NLnet Labs

Authors:

Moritz Müller (SIDN), Marco Davids (SIDN), Willem Toorop (NLnet Labs)

Last modified: Dec 10, 2024

Abstract

RSSAC047 and its updated version RSSAC047v2 “[d]efines measurements, metrics, and thresholds that root server operators (RSOs) meet to provide a minimum level of performance.” These measurements monitor the availability, response latency, correctness and publication delay of the services provided by the root server identifiers (RSI) and the root server system (RSS) as a whole.

RSSAC047v2 also defines requirements that a system performing these measurements should fulfill. ICANN is developing an initial implementation of the measurement software. Reports generated by this measurement software showed that for several months the RSS had failed to meet the availability threshold. Additionally, reports generated by the initial implementation from May 2024 did not mention larger zone publication delays at C-root in the same month.

In this study, we focus on validating the availability and publication delay measurements of the initial implementation of the RSSAC047 measurement software. The main goal is to understand whether the timeouts reported by the measurements were caused by problems at the RSI or whether the timeouts were caused locally at the measurement vantage point.

We perform a statistical analysis of the measurement data, compare them with independent measurements, and analyze the measurement software. We find that the measurement software overall achieves its goal. However, we find signs for problems at measurement vantage points and the network paths between the vantage points and the sites of the RSIs. This indicates that a large number of timeouts reported by the initial implementation might not be caused by the RSIs themselves. When taking this into account, the RSS meets the availability threshold in all but one occasion.

Lastly, we find that a bug in the measurement software led to underestimating the publication delay of the root zone but not to the extent that it would have influenced the outcome of the report. Instead, we find that the formula proposed in RSSAC047v2 to calculate the RSI publication delay metric is not able to pick up publication delays, even if they last for several days. We propose a different, also comparable simple approach, to calculate the RSI publication delay.

Finally, we propose measures to improve the reliability of the collected measurements.

Table of Contents

Introduction

RSSAC047 and its updated version RSSAC047v2 “[d]efines measurements, metrics, and thresholds that root server operators (RSOs) meet to provide a minimum level of performance.”[1] These measurements monitor the availability, response latency, correctness and publication delay of the services provided by the root server identifiers (RSI) and the root server system (RSS) as a whole.

RSSAC047v2 (from now on simply referred to as RSSAC047) also defines requirements which a system performing these measurements should fulfill. ICANN is developing an initial implementation of the measurement software.[2] At the time of writing this report, the software does not monitor the correctness of the provided service yet. All the other metrics have been implemented and are collected stably since 2023-07-01. Aside from running and collecting the measurements, the initial implementation also generates monthly reports that contain information on whether the RSIs and the RSS have met the thresholds of the different metrics.

The reports gained attention from the Root Server Operators (RSOs) due to the Root Server System (RSS) consistently failing to meet the availability threshold for several months. Consequently, the RSOs, specifically Verisign and ISCs, engaged with this consortium for a more in-depth analysis of the measurements and the software responsible for collecting these measurements. The primary objective was to discern whether the timeouts reported by the measurements were a result of issues at the Root Server Instances (RSIs), or if they were caused locally at the measurement vantage point.

While carrying out this study, another incident was brought to our attention. In May 2024,C-root had problems with publishing zones on time.[3] These delays did not appear in the reports generated by the initial implementation. We study whether the measurement platform reported on the publication delays and if not, why.

We focus exclusively on the metrics RSI Availability, RSI Publication Latency, RSS Availability and RSS Publication Latency. The other metrics are not in scope of this report. Also, the goal is not to measure the availability and publication latency of the RSIs and the RSS, but to validate whether the reports about decreased availability and delayed zone publication are plausible. More concretely, we want to understand whether the reported problems actually had their origin at the service provided by the RSOs or whether the problems were caused by the measurement vantage points themselves or the network path from the vantage points to the sites of the RSI.

---

[1] RSSAC027v2: RSSAC Advisory on Metrics for the DNS Root Servers and the Root Server System
[2] RSSAC047 Initial Implementation: https://github.com/icann/root-metrics
[3] https://c.root-servers.org/
“2024-05-23 - On May 21 at 15:30 UTC the c-root team at Cogent Communications was informed that the root zone as served by c-root had ceased to track changes from the root zone publication server after May 18. Analysis showed this to have been caused by an unrelated routing policy change whose side effect was to silence the relevant monitoring systems. No production DNS queries went unanswered by c-root as a result of this outage, and the only impact was on root zone freshness. Root zone freshness as served by c-root was fully restored on May 22 at 16:00 UTC.”

The remaining report is structured as follows: First, we start with the analysis of the initial implementation of the RSSAC047 measurement software. The main goal is to understand whether problems in the measurement and reporting software could have influenced the results. Then, we briefly describe the characteristics of the initial deployment of the measurement software. This section analyzes the coverage and location of the used vantage points. Then, we focus on the availability metrics of RSSAC047. The goal of this section is to find signs that could indicate problems at measurement vantage points or at the RSIs. Here, we rely on statistical analysis of the measurement data collected by the initial implementation and independent measurements collected by RIPE Atlas. Next, we study measurement data collected to monitor the publication delay. In the Section titled Independent measurements of the RSSAC047 root metrics, we deploy the measurement software on our nodes and also test if we can improve the precision of the measurement. In the recommendations, we discuss the tested changes to the measurement software and its deployment that could help to improve the reliability of the collected measurement results. We conclude this report with a short discussion.

Analysis of the initial implementation of the RSSAC047 measurement software

This section contains an analysis (i.e. a code review) of the initial implementation of the measurement system described in RSSAC047v2. This analysis is done to determine whether there are implementation specific aspects that impact reported results.

The purpose of the implementation is to report on metrics to evaluate whether or not the RSS and its individual RSOs are meeting minimal levels of performance. An additional purpose of this initial implementation, as mentioned in the first recommendation of section 8: Recommendations[1] of RSSAC047v2, is to inform future revisions of RSSAC047 with the insights learned from the implementation.

This analysis encompasses:

- Evaluation of ease to deploy and to use the implementation in section Usability.
- Examination and assessment of the source code for four general topics in section Manual code inspection. The four general topics are:
    - Organization (future proof design and the setup of the code base)
    - Quality (craftsmanship, consistency in coding practice and general tidiness)
    - Reliability (the mechanisms in place to assess and guarantee correctness)
    - Security and Privacy

    Each general topic consists of a series of measures which might be satisfied to a greater or lesser extent. Each measure will be reviewed and either marked as 🆗 and/or recommendations for improvement will be reported.

The initial implementation is a work in progress. Measuring Correctness, as described in section 5.3: RSI Correctness is not yet implemented and will not be subjected to analysis.

RSSAC047v2 further mentions that the *initial* implementation can transform into the *official* implementation, for which expectations are enumerated in the second recommendation of section 8: Recommendations[1]. The initial implementation will be evaluated if it already meets the expectations set out for the official implementation in section Expectations.

Finally, a conclusion of the analysis will be given in section Conclusion.

Usability

Repository

The code base of the initial implementation is provided through the web-based Git-repository hosting service: GitHub[4]. A full track record of the development process, including all

[4] https://github.com/icann/root-metrics

activities on branches in present and past is available for inspection at all times. GitHub makes the development process transparent by presenting the track record visually, and linking work items with Issues in the GitHub issue-tracker and with “formal” requests for inclusion of development branches (Pull Requests). Using GitHub makes inspecting the code, in case of issues or for other reasons, easy and also facilitates future collaborations and/or maintenance takeovers.

Organization of code base

The repository contains a `README.md` file describing the general organization of the code base. In addition, a version of excerpts of RSSAC047 is provided (as `rssac047.md`) where all requirements from RSSAC047 are marked with a unique three-letter code “tag”. The code is annotated with these tags to indicate how and where the requirements are implemented.

The `README.md` further describes the components for a running deployment of the implementation and how to deploy those components.

Deployment

The initial implementation needs to be deployed on one or more vantage points that perform the measurements, and a “collector” that collects the measurements from the vantage points, aggregates them and generates reports for the metrics from RSSAC047v2. A deployment is set up on machines set up for the purpose by Ansible playbooks.

Ansible is a popular configuration management software from which configuration can be orchestrated from a controlling machine. Ansible playbooks contain the recipe for configuration. Ansible playbooks can be idempotent (describing a desired configuration state) enabling changes in configuration state to be easily centrally administered and deployed without having to consider the existing state. The playbooks used to deploy the initial implementation were all idempotent.

The initial implementation’s playbooks, use deployment specific configuration, including the inventory of the machines used for vantage points and the collector, from a “Local” directory. Besides having the vantage points listed in the inventory, the vantage points for the initial deployment are also listed hardcoded in the repository in a file called `vp_list.txt`. This file is used by the collector to contact the vantage points and transfer the measurements. Alternative deployments, besides having to provision the list of vantage points into the inventory in the “Local” directory, also need to adapt the `vp_list.txt` file on the collector with their list of vantage points manually.

Running

After deployment, the cron scheduler on the vantage points has been set up to perform the measurements, as described in section 5 of RSSAC047v2, from those vantage points every 5 minutes. The tasks to perform the measurements, and also those to process them in the collector, are implemented in Python.

The collector is also equipped with a cron table, but the tasks therein are commented out and need to be enabled manually. The collector performs the following tasks, each task implemented in a separate Python script:

1. An archive of root zones by serial number is kept up to date since activation. These are currently unused, but will become necessary as part of the correctness measurements implementation.

2. New files containing measurement results are copied from the vantage points.

3. These vantage point measurement files are subsequently processed. Most of their content is transferred into a PostgreSQL database table. The responses to the correctness measurements are stored separately in files.

4. One final script aggregates the results from the database into the RSSAC047v2 defined metrics, and reports on how those values compare to the RSSAC047v2 defined thresholds.

   The script creates two reports for a reporting period: One public version containing only whether or not the thresholds are met (resulting in pass or fail), and one version not for public consumption which also mentions the value resulting from aggregating the measurements for each metric.

   Currently the reports also contain the aggregated results for the RSI and RSS Correctness metric, even though Correctness measurements still need to be implemented. As a result, the correctness metrics currently always show up as 100% correct even though they are not actually performed.

The handling of measurement results is organized in three stages:

1. The measurements are done on the vantage points and the results are stored on disk.

2. The results are copied from the vantage point to the collector.

3. The collector processes the results and stores them in the database.

In the current implementation, there was no indication that results were deleted in any of those stages. After each stage, the results from the previous stage can safely be deleted without loss of information. The organization in stages allows for tuning resources needed for the vantage point and collector, where the vantage points can work with limited disk space. However, this still needs to be implemented.

Manual code inspection

Organization

🆗 The code achieves its purpose

The primary purpose of the implementation is to calculate and report on metrics as set out in RSSAC047v2, and the initial implementation does precisely that for the already implemented metrics (all but the Correctness metrics).

An additional purpose of the initial implementation is to inform future revisions of RSSAC047 with the insights learned from the implementation. The initial implementation has already done so in the past. As is mentioned in RSSAC047v2 section 11: Revision History[5], the current version 2 of RSSAC047 "primarily corrects technical errors and ambiguities found during the initial implementation of these metrics." Furthermore, at the time of writing, the initial implementation was instrumental in finding an issue with the Publication Latency metric, triggered by the fact that the current system did not report C-root's failure to refresh in between May 18 and May 21 2024[6] and further discussed below (see Publication delay).

🆗 The code is readable, commented and easy to understand

The code base contains a straightforward implementation of the procedures that are set out in RSSAC047v2. The relation between requirements in the document and the exact place in the code base where it is implemented are indicated using tags. This was very helpful in the code review process.

The coding style is Pythonic (clear, concise and maintainable) and shows the quality and craftsmanship that only comes from experience. Comments are amply available and supportive of the code base.

🆗 Functions have an understandable role and are not too big to manage

The tasks are performed in individual Python scripts that only need a few supportive functions. Logging and other common tasks are done similarly in all scripts.

🆗 The architecture/design is consistent

The README.md supports understanding the overall organization and design of the code base. The design of vantage point systems performing the measurements, and a central "collector" that aggregates them into metrics, fits the measurement system described in RSSAC047v2 well.

🆗 The syntax is consistent

Variables and functions have descriptive names in all lowercase with underscores between the names.

---

[5] https://www.icann.org/en/system/files/files/rssac-047-03feb22-en.pdf#h.8nxzx38jrt59
[6] See News section on https://c.root-servers.org/

- 🆗 **The code is easy to manage, test and debug**

  All activity is logged with timestamps. All data between the different processing stages is available for inspection at all times. This enables the ability to investigate unexpected outcomes in the reports and reproduce them with corrected code if needed.

- 🆗 **The code base is easy to maintain**

  The combination of the code base on GitHub, a clear readable design outlined in the README.md and clear, readable and well documented code, makes the code easy to maintain and extend. Maintenance of the code base should be transferable without pain.

Quality

- 🆗 **Termination conditions and "out-of-bound" errors are handled**

  Measurement results are not deleted in any of the stages handling them. This could lead to full disk errors. We recommend deleting the measurement results, some time (for example 3 months) after they have been processed in each stage, so that disk usage of at least the vantage points is bounded to a fixed size.

  All measurement results, in the form of result files as well as database rows, do have an associated date attached, making regular cleanup straightforward.

- 🆗 **There is no duplication of code. Functions have a clear purpose and stimulate reuse in future extensions**

  There is a bit of repetition in setting up the boilerplate for logging and other tasks. The code base could benefit somewhat from consolidating that boilerplate into a separate module to be included with all scripts. However, we don't consider this severe enough for an official code analysis recommendation.

- 🆗 **The code contains no hard coded values and comment out code**

  There is one file which contains the names of the vantage points hardcoded: `vp_list.txt`. We recommend generating this list from the ansible inventory.

  The IP addresses of the individual root server instances are stored hardcoded in the Python script that is performing the measurements from the vantage points. We recommend to extract the IP addresses either from the actual root zone, or acquire them dynamically otherwise (for example by downloading the root hints file alongside the root zone).

Reliability

- 🆗 **Tests exist for all functionality of the code base**

  Tests do exist for the still to be implemented Correctness metrics.

We recommend implementing unit tests for the Availability, Response Latency and Publication Latency as well. Not everything can be foreseen, but if there would have been tests for the Publication Latency metric, it might have shown its failure to detect longer outages in an earlier stage (and not after C-root's failures to refresh).

🆗 **Tests are for both success and failure cases**

There are cases for correct and incorrect positive and negative responses.

🆗 **Tests are written to maximize code coverage**

The test for the Correctness metric evaluates the function that is used to measure Correctness for the report. Even though we did not analyze this function since Correctness measurements are still a work in progress, the tested cases cover all possible outcomes.

The other metrics do not have tests and can not be evaluated for coverage.

Security and Privacy

🆗 **The code is secure and privacy aware in terms of authentication, encryption, injections, roles, unauthorized access, etc**

Credentials for the collector to access the vantage point is provided in the Local directory during deployment.

Expectations

Recommendation 2 of section 8 of RSSAC047v2 enumerates a list of six expectations that the *official* implementation must meet. The *initial* (non official) implementation is not expected to meet these, but it is expected that the initial implementation is designed such that it can transform into the official implementation. Below those expectations are repeated with an evaluation of whether or not they are already met by the initial implementation and deployment.

a. **Meet the minimum requirements specified in Section 3 of this report regarding the number, location, connectivity, and other requirements for the vantage points.**

    3.1. The initial deployment does have 20 vantage points as required

    3.2. The vantage points are not evenly distributed among the five geographical regions. Specifically the African region is underrepresented with only a single vantage point.

    3.3. The vantage points are all virtual machines from a variety of reputable cloud providers. Reliable power and diverse connectivity providers may be assumed.

b. **Publish all software related to its operation under an open source license as defined by the Open Source Initiative.**

The implementation ships with a BSD 3-clause license[7] which is an Open Source Initiative approved license[8]

c. **Make the raw measurement data available to anyone in the interest of transparency. A third party should be able to use the raw data to verify the computation of these metrics.**

   We requested and received the raw data and were able to verify the computations.

d. **In its monthly reports, only publish threshold "pass or fail" indicators for each RSI, not the actual measurements or metrics used to determine the threshold pass or fail values.**

   The initial implementation facilitates reports for public consumption that do not include the actual measurements or metrics. At the time of writing no reports have been made publicly available.

e. **Publicly describe its methods for collecting measurements and aggregating metrics, including the topological location of each measurement vantage point. This description should be complete enough for RSOs and DNS researchers to create their own measurement collection systems similar to those used by the official implementation.**

   The initial implementation is a one-on-one literal implementation of the methods for collection and aggregating metrics as described in RSSAC047v2. The source code used by the initial implementation is publicly available and easily usable by third parties to create similar measurement systems. We obtained the location of the vantage points from the implementer of the initial implementation. At the time of writing the locations of the vantage points were not publicly available.

f. **Share with an RSO the underlying measurements and metrics that resulted in failure any time an RSI fails to pass a threshold test. The shared measurements and metrics must include all measurements from around the time of failure and must include all measured values for all transports and address types.**

   The initial implementation preserves all measurement data. We received from the implementer all measurement results for our research. Our research was requested by two RSOs. Us being able to perform the research fulfills this requirement.

Conclusion on the analysis of the initial implementation

The initial implementation of the measurement system described in RSSAC047v2 is thorough and of solid quality. It is a one-on-one literal implementation of the measurements and metric calculations described in RSSAC047. The code base is accessible, readable and can be used to easily set up similar measurement systems. The initial deployment of the initial implementation maintains all measurement data. It was made available upon request and usable to inspect the questions raised by the parties that requested this research. *We did not find any implementation specific aspects that would impact the reported results.*

---

[7] https://github.com/icann/root-metrics/blob/main/LICENSE
[8] https://opensource.org/license/BSD-3-clause

The purpose of the initial implementation is set out in the first recommendation in section 8 of RSSAC47v2:

1. To gather operational data and experience from actual monitoring of the RSS

   This research shows that this purpose is met.

2. To inform future revisions of RSSAC047 with insights learned from the implementation

   The implementation has already proven this purpose, as the second revision of RSSAC047 primarily corrects technical errors and ambiguities found during the initial implementation.

   During the period of this research, C-root failed to refresh the root zone for a period of time, however this did not show up in the reports generated by the initial implementation. We were asked to investigate and found that the way the Publication Latency is determined is suboptimal. This will be valuable input to re-evaluate the metric for a third revision of RSSAC047.

The initial implementation is a work in progres. The evaluation of the queries performed for the Correctness measurements are not implemented yet. We recognise that implementation is an iterative process. Just like the initial implementation provides feedback to future revisions of the RSSAC047v2 report, so does experience with the initial deployment provide valuable insight for future revisions of the initial implementation.

While analyzing the implementation we identified a few fields where improvements are possible. These suggestions can be considered as additional requirements for the initial and official implementation and considered to be added as such to a third revision of the RSSAC047 document.

1. The implementation could benefit from coordinated removal of measurement results to prevent disk space to run out. The vantage points could operate with predictable and stable disk size requirements.

2. The names of the vantage points are currently hardcoded in the code base for the initial implementation. We recommend separating the deployment data from the implementation data to facilitate deployments independent from the initial or official deployment.

3. The IP addresses of the root servers are currently hardcoded in the code base. We recommend using the addresses either from the root zone at the time of measurement, or from the root hints file at the time of measurement.

4. The code base currently has a limited number of tests to verify implementation of the metrics. We recommend providing test data for all the metrics defined in RSSAC047 to be used in verifying the correct implementation of the metrics. For example in an appendix to the RSSAC047 report.

Also the definition of the metrics themselves could have benefitted from such test data. We believe that the inadequacy of the Publication Latency metric would have been exposed in an earlier stage if its capability was verified with test data.

Characteristics of the initial deployment

ICANN operates 20 measurement vantage points located in five different geographic locations running the initial implementation of the measurement software. Additionally, ICANN operators 1 server to collect the measurement results from the vantage points and in order to generate reports summarizing the metrics defined in RSSAC047. The measurements relevant for this report are fully implemented, namely “RSI Availability”, “RSI Publication Latency”, “RSS Availability”, and “RSI Publication Latency”. In this section, we describe the characteristics of the deployment of the initial implementation. We have focussed on the implementation itself in Analysis of the initial implementation.

Coverage

First, we look at the coverage of the deployment of the initial implementation. The initial implementation sets the NSID option when querying for the SOA records but the collector does not process the returned ID. We extended the collector code to extract the NSID from the measurement data as well and store it together with the availability measurement data in the database. We strip the server identifier from the returned NSID such that we keep only the location of the city. E.g. *a5.nl-ams.root* becomes *nl-ams.root.* We refer to the latter as the “site” in the remainder of the document.

Of the 1,501 sites reported on root-servers.org, 253 were seen by the vantage points of the initial implementation at least once in January 2024 As shown in the table below, coverage varies largely between RSIs. For some RSIs, the vantage points reach less than 20 individual sites, which indicates that the location of the vantage point leads to insufficient and redundant coverage. For some RSIs, the vantage points reach more than 20 individual sites over the course of the measurement period. This could be caused by changes in catchment or the addition of new sites.

RSI	Sites	Sites seen by VP
A	59	18
B	6	6
C	12	10
D	209	26
E	328	30
F	345	26
G	6	6
H	12	12
I	82	29

RSI	Sites	Sites seen by VP
J	150	27
K	120	22
L	151	30
M	21	11

Table: Number of sites per RSI (all and global) and the number of sites seen by the vantage points of the initial implementation in January 2024.

The vantage points reach a diverse set of sites. In accordance with RSSAC047, the initial implementation initiates measurements every 5 minutes. For January 2024, we counted per RSI, IP version, site and 5 minute measurement interval how many vantage points have reached the site. The figure below shows the maximum number of vantage points that reached a site at the same time. 69% of all sites were always reached by only one vantage point at the same time this month.

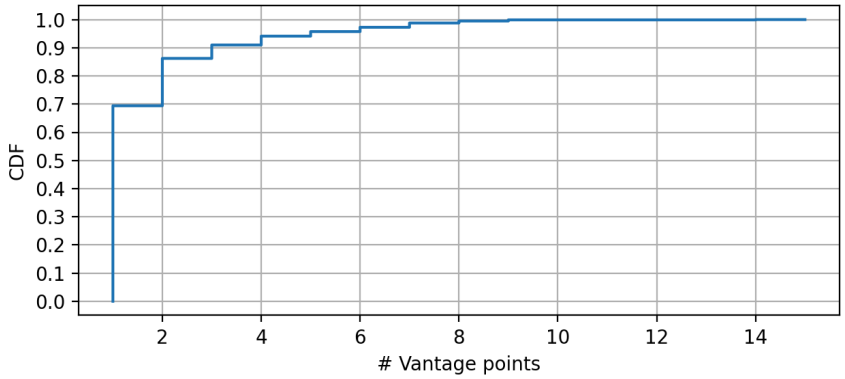


Figure: Maximum number of vantage points that reach the same site at the same time in January 2024.

Traceroute measurements

In each measurement interval, the vantage point initiates a traceroute towards each IP of all 13 RSIs. For the period 2023-07-01 until 2024-03-31, 38,991,700 traceroutes were performed out of which 61.14% reached their target on IPv4 and 51.25% on IPv6. Reasons for not reaching their target vary, but for the majority the traceroute terminated after it encountered 5 hops that were not responding. The distance from the vantage points towards

the RSIs vary. On both IPv4 and IPv6, the median hop count per vantage point ranges from 7 to 13 hops (median 9 hops).

Since a large number of traceroute measurements do not reach their target regardless, their added value is limited. However, as we will show later, they reveal problems on the network path like routing loops in a few cases. Also, they might help to explain anomalies, like an increase in response times.

Availability

Initial implementation measurement results

For this report we focussed on measurement data collected between 2023-07-01 and 2024-03-31. We relied on data and reports provided to us by the operator of the initial deployment of the measurement software.

For the measurement period from July 2023 to March 2024 the initial implementation reported 13 times that the RSS as a whole did not meet the threshold of 99.999% availability (see table below).
The individual RSIs always met their individual threshold of 96% availability.

Month	IP Version	Transport	Availability
2023 July	4	UDP	99.997%
2023 August	4	UDP	99.992%
	6	UDP	99.992%
2023 September	4	UDP	99.997%
	6	UDP	99,996%
2023 October	4	UDP	99.993%
	6	UDP	99,987%
2023 November	4	UDP	99.986%
	6	UDP	99,984%
2023 December	4	UDP	99,996%
	6	UDP	99,996%
2024 January	6	UDP	99.996%
2024 March	6	UDP	99.844%

Table: RSS availability not meeting the threshold between July 2023 and March 2024.

Note that in August 2023, the RSS also did not meet the response rate latency threshold of 0.150 seconds via IPv6 and UDP (0.164 seconds median). We studied this further and found that this was caused by a bug in the initial implementation. With the bug being fixed, the latency decreased to 0.021 seconds (median). Our fix has been integrated.[9]

Timeout overview

Throughout this period, the measurement platform carried out in total 81,370,764 availability measurements directed towards each RSI, via IPv4 and IPv6 and via UDP and TCP. Because the RSS has only failed to meet the availability threshold via UDP, we focussed on measurements using this transport protocol only.

The table below shows the availability measurements and the recorded timeouts. Measurements by the initial implementation timed out 1.7 times more often via IPv6 than via IPv4 (85,721 timeouts on IPv6 vs 50,474 timeouts on IPv4).

Month	IP Version	Transport	Total Measurements	Timeouts	Timeouts %
2023 July	4	UDP	2,312,180	3,712	0.16%
	6	UDP	2,312,179	3,315	0.14%
2023 August	4	UDP	2,321,085	8,736	0.38%
	6	UDP	2,321,085	8,105	0.35%
2023 September	4	UDP	2,215,031	5,303	0.24%
	6	UDP	2,215,030	5,288	0.24%
2023 October	4	UDP	2,312,245	4,528	0.20%
	6	UDP	2,312,244	5,974	0.26%
2023 November	4	UDP	2,211,378	8,882	0.40%
	6	UDP	2,211,378	9,839	0.44%
2023 December	4	UDP	2,270,213	3,145	0.14%
	6	UDP	2,270,214	9,050	0.40%

[9] https://github.com/icann/root-metrics/pull/17

Month	IP Version	Transport	Total Measurements	Timeouts	Timeouts %
2024 January	4	UDP	2,320,838	6,138	0.26%
	6	UDP	2,320,838	15,030	0.65%
2024 February	4	UDP	2,135,874	4,854	0.23%
	6	UDP	2,135,874	13,426	0.63%
2024 March	4	UDP	2,243,969	5,176	0.23%
	6	UDP	2,243,969	15,694	0.70%

Table: Availability measurements overall and measurements resulted in a timeout between July 2023 and March 2024.

In order to understand how long timeouts lasted we summarized timeouts that were observed by the same vantage point for the same RSI, IP version, transport protocol and site consecutively as a *timeout event*. To count as a consecutive timeout event, no successful DNS query for the SOA record between the start and end of the timeout event must have been observed.

We extracted the site from the returned NSID. Obviously, measurements that resulted in a timeout do not contain a NSID. To understand at which site the query likely would have ended up, we extract the NSID of the last successful availability measurement received at the same vantage point from the same RSI, via the same IP version and transport protocol. We treated all servers at a site the same and stripped any information from the NSID identifying individual servers.

In total, we observed 94,388 timeout events (compared to 136,195 individual timeouts). 84.65% of all timeout events lasted only for one measurement interval - meaning that after observing one timeout in one measurement interval the next measurement attempt was successful. As shown in the figure below, 15.23% of timeout events lasted longer than 5 minutes (one measurement interval) but less than an hour (12 consecutive measurement intervals). Only 0.12% of observed timeouts lasted for one hour or longer.

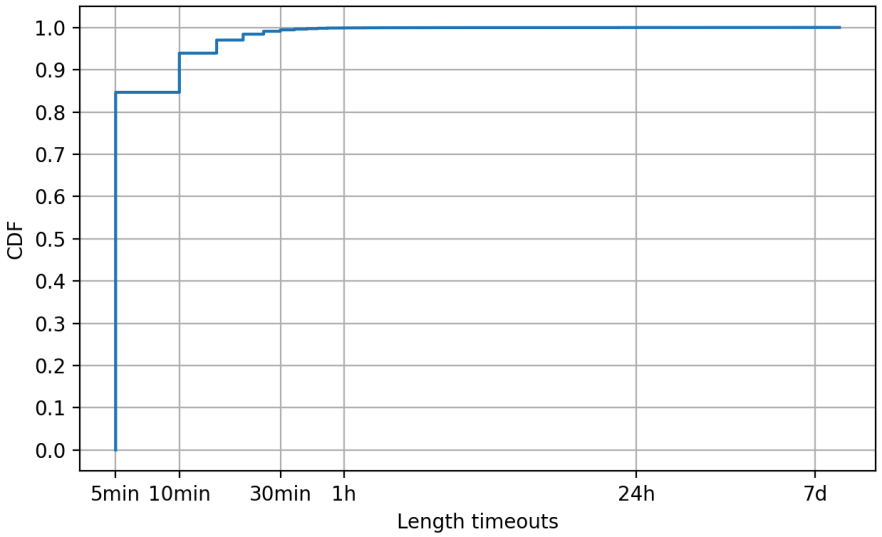


Figure: Length of timeout events from July 2023 to March 2024. X-axis is in log scale.

Timeouts per RSI

The figure below shows the timeouts observed per RSI and IP version. Timeouts are not spread across RSIs evenly.

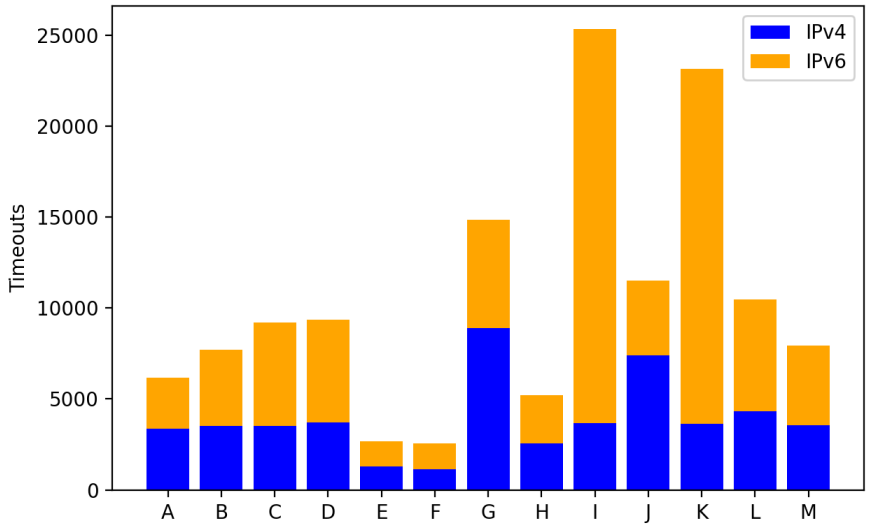


Figure: Timeouts via UDP per RSI and IP version from July 2023 to March 2024.

We also wanted to understand at which *sites* the timeouts were observed. The figure below shows the number of timeouts per RSI, site and IP version. For each RSI only the 5 sites with the most timeouts are shown. The number of received timeouts varied between the different sites. For four sites, timeouts were exclusively reported for queries via IPv6, compared to two sites for which timeouts were only reported for queries via IPv4.

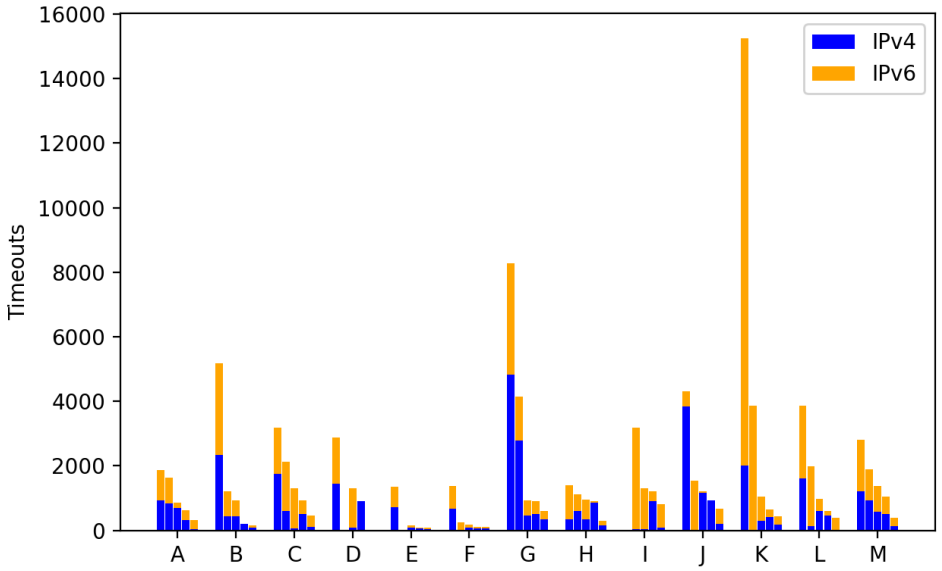


Figure: Timeouts via UDP per RSI, site and IP version from July 2023 to March 2024. Top 5 sites only.

Timeouts per vantage point

Now we look at timeouts from the perspective of a vantage point. The number of timeouts observed by each vantage point varies. The table below shows the number of timeouts observed by each vantage point order by the number of timeouts.

Vantage point	**All Timeouts**
202	18,871
405	17,640
507	13,230
502	7,543
305	3,659
203	3,141
205	2,537
306	2,324
403	2,317
404	2,212
307	2,124
206	2,048

Vantage point	All Timeouts
201	2,031
102	1,659
501	1,353
304	1,333
505	934
503	920
504	912
303	616

Table: Timeouts observed per vantage point between July 2023 and March 2024.

Timeouts per vantage point and RSI

The figure below shows the number of timeouts as measured by each vantage point separated by IP version. Additionally, each bar shows for which RSI the timeout was observed. Especially vantage point 507 stands out, since it observes timeouts across both IP versions and all RSIs almost evenly. Vantage point 502 shows a similar pattern over IPv6.

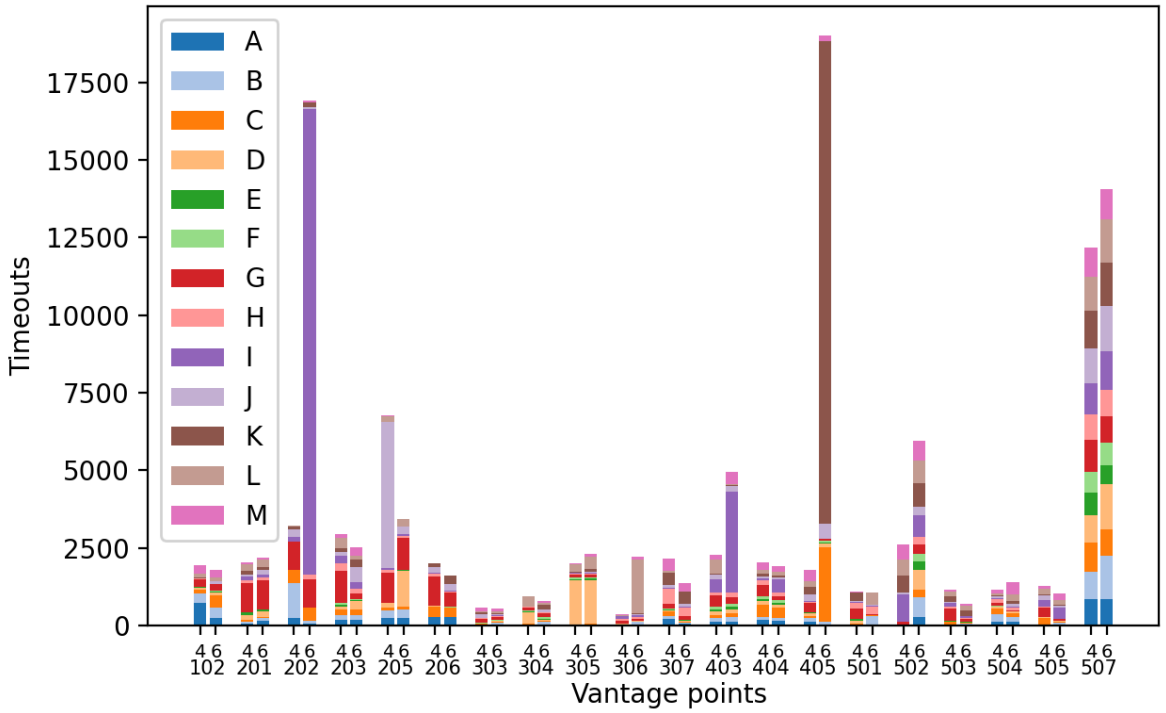


Figure: Timeouts via UDP per vantage point, IP version and RSI from July 2023 to March 2024.

Overlapping timeouts

98.35% of all timeouts were only observed by one single vantage point. This is expected, since only a small number of sites are reached by more than one vantage point (see Characteristics of the initial deployment).

Often, single vantage points observed multiple timeouts at the same time. For 37.32% of all timeouts, the vantage point observed another timeout in the same measurement interval. If we only take timeouts into account that were observed at another RSI but on the same IP version, then still 26.16% of all timeouts were measured by vantage points that observed another timeout at the same time. If this is the case, then this is for us a strong sign that the timeout was caused by the measurement vantage point.

Here, especially vantage points 404, 507 and 307 stand out. Vantage point 404 observed for 56.14% of all the timeouts at least one other timeout simultaneously. For vantage points 507 and 307, this is the case for 77.14% and respectively 77.56% of the timeouts.

Correlating timeouts with traceroute measurements

Aside from the DNS measurements, RSSAC047 requires the measurement vantage points to send traceroutes simultaneously. The initial implementation uses the tool scamper[10] to

[10] https://www.caida.org/catalog/software/scamper/

send UDP based paris-traceroutes. We analyzed the traceroutes collected 5 minutes before, during and 5 minutes after a vantage point observed a timeout.

By default, scamper stops probing after it encounters 5 unresponsive hops. This was the case for 53.47% of traceroutes collected during a timeout (compared to 53.12% before the timeout). This limits the information we can gain from the traceroutes.

For 3.39% of the timeouts the traceroute terminated because it encountered a routing loop, while it had not encountered a loop in the measurement interval before the timeout. 89.1% of these timeouts occurred on IPv6. This indicates problems on the network path towards the affected RSI.

As discussed in Characteristics of the initial deployment, the distance from the vantage points towards the sites of the RSIs varies. However, we could not find signs that a higher hop count correlates with a higher number of timeouts.

Summary

The measured availability of individual sites varies strongly.

Two vantage points showed behavior that indicate problems with the vantage points itself. They experienced a roughly equal number of timeouts for each RSI. We consider it unlikely that the sites of the RSIs reached by those vantage points had the same level of unavailability.

Also, vantage points observed multiple timeouts for different RSIs at the same time in more than a quarter of all observed timeouts. We see this as an indication that these timeouts were not caused by the individual RSIs either.

RIPE Atlas Measurements

From the analysis of the measurements carried out by the initial implementation it is not possible to draw strong conclusions about whether the observed timeouts were caused by problems at a RSI or whether the timeouts were independent of the functioning of the RSIs.

For this reason, we use the measurement platform RIPE Atlas to verify whether the timeouts observed by the initial implementation were also observed by RIPE Atlas probes. The RIPE Atlas measurement platform consists of more than 12,000 vantage points.[11] Aside from measurements run by users, these probes continuously send a number of ICMP and DNS based measurements to RSIs, both via IPv4 and IPv6.

Every four minutes, they send a DNS “CHAOS” query to the root servers with “hostname.bind” as a query name. The hostname identifies the site of the RSI. This and other built-in measurements towards the root do not ask for the NSID. For this reason we

[11] https://atlas.ripe.net/coverage/

rely on those measurements for our further analysis and map the returned hostname to the NSID reported by the initial implementation.

Methodology

In this analysis we relied on the RIPE Atlas measurement results stored on Google Big Query (GBQ). Because the RIPE NCC only stores results on GBQ for roughly six months, we had to limit our measurement study to the time period between 2023-11-09 and 2024-04-03.

For each timeout observed by the initial implementation, we collected all RIPE Atlas measurements for the affected RSI and IP version 12 hours before and 12 after the timeout occurred.

Filters

In order to improve the reliability of our approach we applied filters before processing the measurement results further.

Because older probes are known for being more unreliable, we did not consider measurement results collected by probes of version 1 and 2.[12] Additionally, we only took measurement results into account from probes that were automatically tagged by the RIPE Atlas measurement platform as resolving A and AAAA records correctly, indiciating reliable DNS resolution.

Multiple studies in the past have observed DNS manipulations at RIPE Atlas probes. In order to take only measurements into account that likely reached the real RSIs, we only relied on measurement results that included a site that was also seen by RIPE Atlas anchors the same day. Anchors are more reliable and potentially more trustworthy vantage points on the RIPE Atlas platform, and we believe less likely prone to DNS manipulations.
After applying these filters, our RIPE Atlas measurements observed 987 unique sites (65.76% coverage of all sites).

Identifying timeouts in RIPE Atlas

Per observed site, we grouped the 24 hours worth of measurements into buckets of 5 minutes. For each bucket, we counted the number of RIPE Atlas probes that reached that site.

In this time series, we tried to identify if and when the number of probes reaching a certain site dropped significantly. Our assumption is that when the number of probes reaching a site drops under a certain threshold, the affected site has become largely unavailable. By aggregating measurements from independent probes, we assume that the experienced drop in availability was not caused by problems at the individual probes, but by problems at the RSI.

---

[12] Lessons Learned From Using the RIPE Atlas Platform for Measurement Research, Bajpai et al., https://dl.acm.org/doi/abs/10.1145/2805789.2805796

In order to identify drops in reachability, we calculated the average number of probes reaching a site over the 24 hour time frame. Additionally, we calculated the standard deviation across the 5 minute buckets. Since identifying drops reliably is not trivial we apply a liberal approach and a conservative approach. The conservative approach has a low false positive rate meaning that it identifies drops with higher confidence but might miss less significant drops. The liberal approach is able to identify more drops, but might falsely classify noise as a drop. Thus, the conservative approach provides a lower bound of identified drops and the liberal approach an upper bound.

In the conservative approach, we consider a drop if the observed number of probes *X* reaching a certain site is lower than the mean (μ) minus 3 times the standard deviation (σ) ($X < \mu - 3\sigma$). In the liberal approach, the value only needs to be lower than the mean minus 2 times the standard deviation ($X < \mu - 2\sigma$). We have manually looked at a sample of classified drops and found the method to work reliably.

Correlate timeouts with RIPE Atlas measurements

After identifying a drop in the RIPE Atlas measurements we tested if the drop occurs roughly at the same time the vantage point of the initial implementation has received a timeout for the same site. We considered that a drop observed by RIPE Atlas has correlated with a timeout observed in the initial implementation if it has occurred not more than 5 minutes before or 5 minutes after the timeout. We allow for this 10 minute bandwidth because of the 5 minute granularity of both measurements.

As an example, the figure below shows the number of probes that have reached a certain site over time. The vertical line marks the measurement interval at which point the initial implementation has observed a timeout. The drop is clearly visible: the number of probes reaching the sites decreases from roughly 100 per 5 minute time slot to almost 0. This lasts for roughly 2 hours during which also the initial implementation observed a timeout.

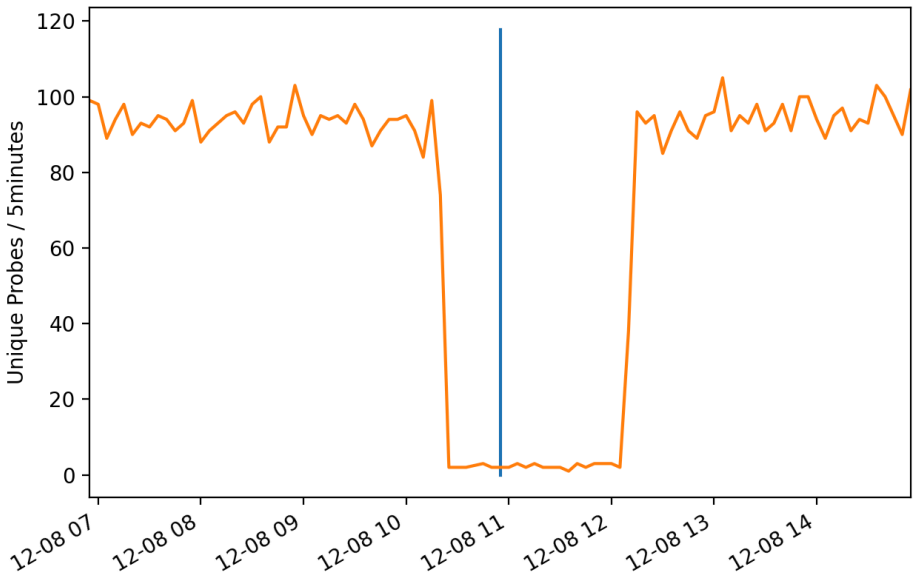


Figure: Example of a timeout observed by the initial implementation (vertical line) automatically correlated with drop in reachability observed by RIPE Atlas.

Limitations

This approach has a number of limitations. First, the response to the CHAOS query of RIPE Atlas did not always match with the NSID returned by the SOA query, e.g. because there were occasions where the hostname.bind value also contained the country code, while the NSID value did not. Also, the NSID of one root server only contained a general identifier for a period of time.

Second, we only took RIPE Atlas measurements into account that returned values for hostname.bind that were also seen by RIPE Atlas anchors. Despite the fact that almost 800 anchors were available it was not guaranteed that we were able to reach all available sites. As a consequence, we were not able to find RIPE Atlas measurements for all timeouts reported by the initial implementation.

Third, it is well known that the probes of RIPE Atlas are not evenly distributed across countries and networks. As a consequence, we were not able to correlate some timeouts.

Correlating timeouts with RIPE Atlas measurements

Between 2023-11-09 and 2024-04-03 the initial implementation observed 87,404 timeouts. For 87.48% of these timeouts, we were able to find measurements in RIPE Atlas that were reaching the same site of the queried RSI.

For 1,604 timeouts we could identify a drop in reachability in RIPE Atlas with high confidence (conservative threshold). This accounts for 1.84% of all timeouts and 2.10% timeouts for which we could find corresponding RIPE Atlas measurements. When applying the liberal threshold, this number increases to 10,464 timeouts (11.97% of all timeouts and 13.69% of matched timeouts). For the vast majority of timeouts we could not identify a significant drop in reachability at RIPE Atlas.

Classify timeouts

To classify whether a timeout observed by the initial implementation was likely caused by problems at the RSI, by problems at the measurement vantage point, or by problems on the path between the vantage point and the site, we combined data collected by RIPE Atlas with the analysis in Initial implementation measurement results.

For a timeout to be caused at the vantage point, we defined the following criteria:

- A timeout is **very likely** caused by problems with the measurement vantage point when all four criteria hold:
    1) The timeout has *not* been observed by another vantage point at the same time, and
    2) the vantage point has observed another timeout at the same time, and

    3) the timeout does not coincide with a traceroute measurement that resulted in a loop, and
    4) the timeout could not be correlated with a drop in RIPE Atlas with *low confidence*.
- A timeout is **likely** caused by problems with the measurement vantage point when all the above criteria hold, except from *2).*

For a timeout to be caused at the site of the RSI we defined the following criteria:

- A timeout is **very likely** caused by problems at a site of a RSI when all three criteria hold:
    1) The vantage point has *not* observed another timeout at the same time, and
    2) the timeout does not coincide with a traceroute measurement that resulted in a loop, and
    3) the timeout could be correlated with a drop in RIPE Atlas with *high confidence*.
- A timeout is **likely** caused by problems at a site of a RSI when:
    1) The vantage point has *not* observed another timeout at the same time. and
    2) the timeout does not coincide with a traceroute measurement that resulted in a loop, and
    3) the timeout could be correlated with a drop in RIPE Atlas with *low confidence*.

For a timeout to be caused on the path between the vantage point and the site of the RSI we defined the following criteria:

- A timeout is **likely** caused by problems on the network path when:
    1) There exists a traceroute measurement towards a site of an RSI that resulted in a routing loop at the time of the timeout, and
    2) there does not exist a traceroute measurement towards a site of an RSI that resulted in a routing loop one measurement interval before the timeout, and
    3) the timeout could not be correlated with a drop in RIPE Atlas with *low confidence*.

The figure below shows a classification tree of the individual timeouts.

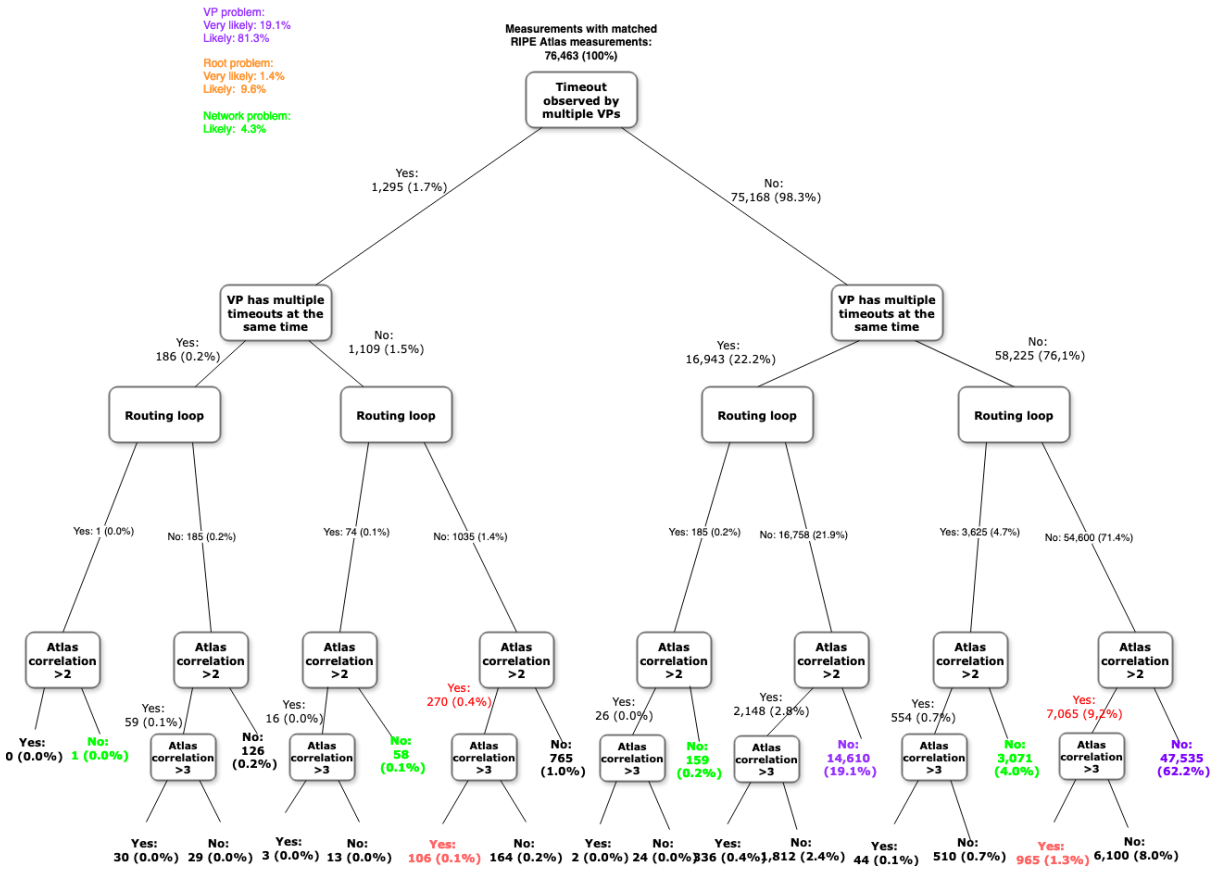


Figure: Classification of individual timeouts into the cause likely being a problem of measurement vantage points or of an RSI (larger version in the Appendix: Figure: (Large) Classification of individual timeouts into the…).

We consider it *very likely* that 19.1% of the observed timeouts were caused by problems at the measurement vantage point or its connectivity. In 81.3% of the cases we consider the chance that the timeout had its root cause *not* at the RSI as *likely*. In comparison, we consider between 1.4% and 9.6% of the timeouts as caused by the RSI itself. 4.3% of the timeouts are likely caused by problems on the path between the measurement vantage point and the site of the RSI.

Problem location	**Likely**	**Very likely**
Measurement vantage point	81.3%	19.1%
Network path	4.3%	-
RSI	9.6%	1.4%
Unknown	4.8%	79.5%

Table: Timeouts classified by location, observed between July 2023 and March 2024.

Timeouts, likely caused by problems with the vantage points, had a similar duration as timeouts likely caused by problems at the RSI, as shown in the figure below.

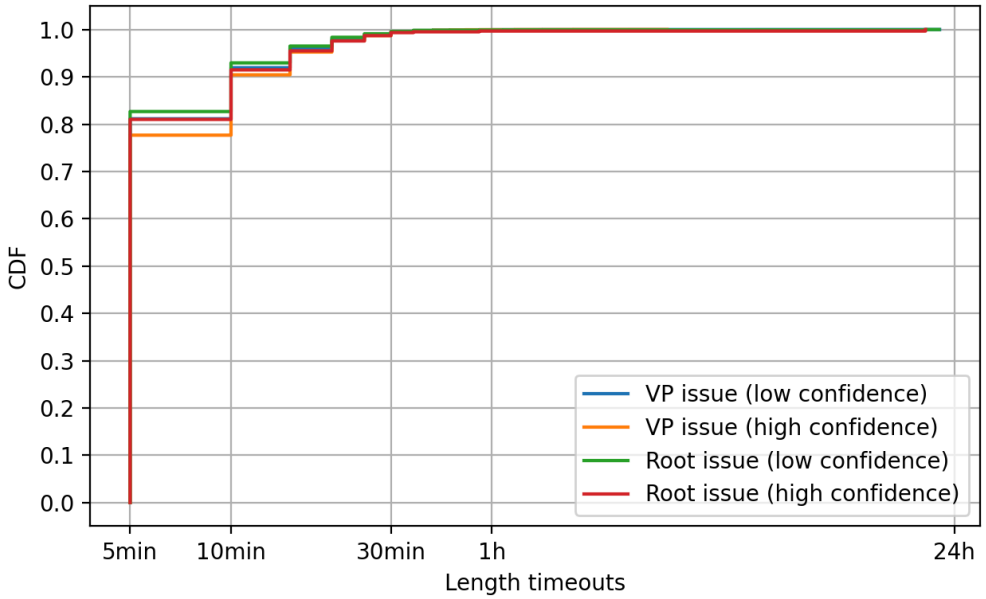


Figure: Length of timeout events correlated with RIPE Atlas measurements and classified. X-axis is in log scale.

Possible problems with vantage points

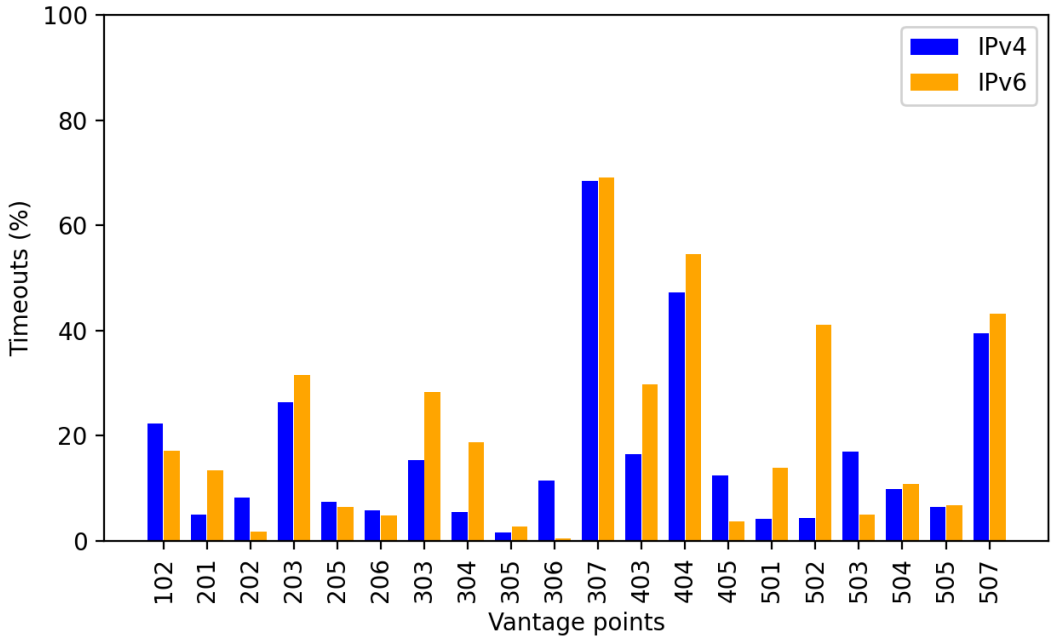
We looked closer at the timeouts we suspect were caused by problems at the vantage points. The table below shows the timeouts observed by each vantage point and the share of timeouts we suspect were caused by problems at the vantage point, ordered by the share of timeouts that we classified as vantage point problems with high confidence.

Vantage point	Confidence (high)	Share (%)	Confidence (low & high)	Share (%)
307	1460	68,7	1727	81,3
404	1123	50,8	1720	77,8
507	5508	41,6	8879	67,1
502	2325	30,8	5809	77,0
203	937	29,8	2110	67,2
303	136	22,1	360	58,4
403	510	22,0	1548	66,8
102	330	19,9	1156	69,7

Vantage point	Confidence (high)	Share (%)	Confidence (low & high)	Share (%)
503	115	12,5	543	59,0
304	144	10,8	707	53,0
504	96	10,5	671	73,6
201	201	9,9	863	42,5
501	116	8,6	1021	75,5
205	175	6,9	1151	45,4
505	62	6,6	524	56,1
206	109	5,3	1208	59,0
405	813	4,6	14179	80,4
202	493	2,6	16621	88,1
305	81	2,2	2460	67,2
306	35	1,5	2118	91,1

Table: Least reliable vantage points (IPv4 and IPv6) sorted by share of timeouts classified with high confidence as caused by local problems, starting from 2023-11-09 until 2024-04-03

Of the timeouts categorized as VP problem, 5362 (36.7%) were on IPv4 and 9248 on IPv6 (63.3%). The figure below shows the timeouts classified as vantage point problems separated by vantage point and IP version. Especially vantage points 307, 404 and 507 seem least reliable, regardless of the measured IP version.



IPv4 and IPv6 timeouts classified with high confidence as vantage point problem 2023-11-09 - 2024-04-03

Adjusted root server metrics

Taking the classification from the previous section into account, we re-calculated the RSSAC047 RSI and RSS availability metrics. Because we only had RIPE Atlas data available for the full months of December 2023 until March 2024, we only took these months into account.

RSI availability

First, we focussed on the RSI availability metrics. We also took timeouts into account for which we could not find corresponding RIPE Atlas measurements.

Then, we calculated the availability for each RSI three times:

- Not treating timeouts likely or very likely caused by vantage point problems as a successful query (default).
- Treating timeouts likely caused by vantage point problems and likely caused by network problems as a successful query (liberal RSI availability).
- Treating timeouts very likely caused by vantage point problems as a successful query (conservative RSI availability).

When treating timeouts likely caused by problems with the vantage point as a successful query, the lowest measured availability for any RSI is *99.457,* compared to 97.058, as shown in the table below.

	All (default)	***Conservative***	***Liberal***
Mean	*99.596*	*99.659 (+0.063)*	*99.909 (+0.313)*
Median	*99.780*	*99.853 (+0.073)*	*99.960 (+0.180)*
Min	*97.058*	*97.287 (+0.229)*	*99.419 (+2.361)*
Max	*99.980*	*99.992 (+0.012)*	*99.997 (+0.017)*

Table: RSI availability for the months December 2023 until March 2024, corrected by timeouts that were classified as VP problems.

RSS availability

The availability of individual RSIs also met the threshold defined in RSSAC047 before we corrected the measurements by timeouts likely caused by problems with the vantage points. However, the availability of the RSS failed the threshold of 99.999% four times between December 2023 and March 2024.

When calculating the RSS availability for those months and considering timeouts likely caused by vantage points as successful measurements, the RSS meets the threshold in each month, except for March 2024 and IPv6. Still, for this month and IP version, the availability metric increased from 99.855 to 99.998.

One likely reason that even the corrected availability metric failed in March 2024 is the fact that we could not find corresponding RIPE Atlas measurements for a relatively high number of timeouts (88.88% covered timeouts vs 93.9% - 95.5%). In March 2024, the initial implementation reported timeouts for two sites not covered by RIPE Atlas.

		Availability		
Month	**IP version**	**Default**	**Low Confidence**	**High Confidence**
2023-12	**v4**	99.995848	100.000000	100.000000
	v6	99.995920	100.000000	100.000000
2024-01	**v4**	99.999160	100.000000	100.000000
	v6	99.995519	99.999650	99.999650
2024-02	**v4**	99.999467	100.000000	100.000000
	v6	99.999163	99.999848	99.999848
2024-03	**v4**	99.999638	100.000000	100.000000
	v6	99.844015	99.998117	99.998117

Table: RSS availability for the months December 2023 until March 2024, corrected by timeouts that were classified as VP problems. Red font marks measurements that did not meet the availability threshold of 99.999%

Takeaways

We consider that at least 19.1% of all timeouts are not caused by problems at RSIs. For these timeouts, we could not find signs of availability issues in RIPE Atlas since. Additionally, vantage points that observed these timeouts had also problems reaching other RSIs at the same time. This number even rises to 81.3% if we loosen the latter requirement. If we additionally take potential problems on the network path into account, then 85,6% of all

timeouts are caused by problems at the vantage point or on the path towards the site of the RSI. In contrast, we could only find in 1.4% of all timeouts strong signs for problems at the RSI.

We discuss what these findings mean for measuring and calculating the availability metrics in more detail in the Discussion and recommendations.

Publication delay

On 21 May 2024, the DNS community noticed that the servers of C-Root had not published the latest root zone.[13] While C-Root appeared to be serving the root zone with serial 2024051801 all other root name servers were serving the root zone with serial 2024052101. One day later, members of the mailing list were reporting that C-Root was serving the zone with serial 2024052101 while the other root servers were serving the root zone with serial 2024052200. On 23 May it was reported that the C-Root was back in sync with the other root servers. The RSO of C-Root has confirmed this incident.[14]

The initial implementation of the measurement software reported for C-Root a publication delay of 0.0 seconds for May 2024. Considering the duration of the incident, this came to a surprise to many. For this reason, we study whether the initial implementation measured and reported the publication delay correctly in this section. We come to the conclusion that the initial implementation had a software bug, but otherwise calculated the publication latency as described in RSSAC047. The metric itself, however, should be improved.

Initial implementation measurements and reporting

Measuring the publication delay

The initial implementation contains code to measure the publication delay. We tested, if the initial implementation had observed the increased delay and the missing zone files.

In May 2024, the vantage points observed that 74 out of 85 zones published by C-Root were on time (not more than 65 minutes late). 3 zones were late for more than 65 minutes and 8 zones were not observed by the vantage points of the initial implementation. The late zones are the zones that were reported by the DNS community and the missing zones fall into the time window of the incident.

This shows that the initial implementation identified the reported problems with zone publication at C-Root correctly.

[13] As first reported on the mailing list “dns-operations” by DNS OARC on 21 May 2024.
[14] https://c.root-servers.org/

Calculating the publication delay

We found that the input for calculating the publication delay metric was very likely correct. We therefore put our focus on calculating and reporting the publication delay.

Here, we observed one bug in the initial implementation that in some cases could lead to underestimating the publication delay of individual RSIs. However, in the case of the incident at C-Root, this bug did not affect the reported publication delay. We have submitted a fix for this bug, which has been merged to the main branch of the initial implementation.[15]

Adjusted publication delay metric

We have shown that the vantage points of the initial implementation have picked up on the increase in publication delay at C-Root in May 2024 and also the missing zone files. The code to calculate the publication delay of the individual RSIs did contain a bug, however this bug was not significant enough to have led to a different publication delay.

We therefore come to the conclusion that the formula to calculate the publication delay of individual RSIs is not sufficient to pick up on even large delays as the ones observed in May 2024 at C-Root. For this reason, we propose a slightly different approach for calculating the publication delay.

First, we recommend taking missing zone files into account as well. If a new serial number is not seen because of failure to refresh, the amount of time elapsed until a given vantage point observes the first observed next new serial number is calculated. Or, if there is no first observed next new serial, the time elapsed until the current time is taken.

Second, even when taking missing zone files into account, relying on the median number of publication delays is probably not sensitive enough. In the most extreme case, an RSI could have several hours of publication delay for around half of the zone files and a publication delay below 65 minutes for the other half and would still meet the threshold. In contrast, taking the mean would have enabled us to identify the increase in publication delays at C-Root. The table below shows the calculating publication delay when applying our proposed method, compared to the method described in RSSAC047.

RSI	Measurements	Median publication delay (s)	Adjusted publication delay (s)
A	85	0.0	21.2
B	85	0.0	17.6
C	85	0.0	11,523.5
D	85	0.0	17.7
E	85	0.0	28.2

[15] https://github.com/icann/root-metrics/pull/3

RSI	Measurements	Median publication delay (s)	Adjusted publication delay (s)
F	85	0.0	24.7
G	85	0.0	14.1
H	85	0.0	14.1
I	85	0.0	17.6
J	85	0.0	3.5
K	85	0.0	17.6
L	85	0.0	35.3
M	85	0.0	17.6

Table: Publication delay for May 2024 for each RSI, calculated according to RSSAC047 (median) and by calculating the mean.

When applying the threshold defined in RSSAC047 of 65 minutes (3900 seconds), C-Root would not have passed the test when taking missing zone files into account and when calculating the publication delay using the mean instead of the median.

When calculating the publication delay for the RSS as a whole we could either calculate the median across the individual publication delays or use again the mean. The median of the RSI's publication delays is 17.6 seconds. Using the mean the publication delay of the RSS is 904.1 seconds (15.1 minutes). Both delays are below the threshold defined in RSSAC047 of 35 minutes (2100 seconds).

Independent measurements of the RSSAC047 root metrics

We deployed the code of the initial implementation on 17 vantage points. We collected the metrics for several weeks and in this section we discuss the results from August 2024. The goal of this exercise was to test the deployment of the software on a larger scale and in a setup that resembles the deployment by ICANN to some extent. This also enabled us to compare metrics reported by ICANN's deployment with the metrics reported by our study to find possible discrepancies. Finally, it allowed us to evaluate the effectiveness of extensions to monitor the reliability of VP measurements.

Setup

We deployed the measurement software on 17 vantage points located at data centers by Vultr. Our vantage points had the following specifications and were deployed at the following locations.

Vantage point specifications:

- CPU: 2 vCPUs (Vultr category "High Performance")
- Memory: 2 GB

Vantage point locations:

Location	**Hostname**
Amsterdam, Netherlands	vp-ams1.sidnlabs.nl
New Delhi, India	vp-del1.sidnlabs.nl
São Paulo, Brazil	vp-gru1.sidnlabs.nl
Honolulu, United States	vp-hnl1.sidnlabs.nl
Seoul, South Korea	vp-icn1.sidnlabs.nl
Osaka, Japan	vp-itm1.sidnlabs.nl
Johannesburg, South Africa	vp-jnb1.sidnlabs.nl
Madrid, Spain	vp-mad1.sidnlabs.nl
Melbourne, Australia	vp-mel1.sidnlabs.nl
Mexico City, Mexico	vp-mex1.sidnlabs.nl
Miami, United States	vp-mia1.sidnlabs.nl
Santiago, Chile	vp-scl1.sidnlabs.nl
Seattle, United States	vp-sea1.sidnlabs.nl
Singapore, Singapore	vp-sin1.sidnlabs.nl

Location	Hostname
San Jose, United States	vp-sjc1.sidnlabs.nl
Tel Aviv, Israel	vp-tlv1.sidnlabs.nl
Warsaw, Poland	vp-waw1.sidnlabs.nl

Table: Vantage points and hostnames

Note that also the deployment of ICANN relies on nodes at Vultr. Our measurements thus possibly overlap with ICANN's measurements to some extent.

Collector:
The collector was located in our own network, with the following specifications:

- Platform: Virtual machine
- Location: AS 215088
- CPU: 16 cores
- Memory: 24GB

Extensions

We implemented and tested two extensions to the initial implementation of the measurement software and its deployment.

First, we hypothesized that additional measurements from the vantage points towards services on the Internet that are not reliant on the DNS could give us additional insights into the availability of the vantage points. Thereby, we would be able to filter out timeouts not caused by the RSS.

For this reason, we implemented additional traceroute measurements towards the IPv4 and IPv6 addresses of the Public DNS services by Cloudflare and Google. Every 5 minutes the vantage points performed a traceroute measurement towards these four IP addresses and we discuss their impact on classifying timeouts later in this section. The traceroute measurements are also part of this fork of the initial implementation.

Second, we performed continuous ICMP ping measurements from an independent VM located in our own network towards each of our vantage points using smokeping[16]. Also here, we hypothesized that this would give us more insights into the availability of a vantage point, but as with the original availability measurements towards the RSS classifying an ICMP ping timeout remains challenging.

Additionally, our software contained a fix for calculating the publication delay (as discussed in Analysis of the initial implementation of the RSSAC047 measurement software).

[16] https://oss.oetiker.ch/smokeping/

Results

First, we briefly discuss the availability measurements as reported by our deployment of the extended initial implementation and compare it with the metrics reported by the deployment of ICANN. Then, we attempt to classify the reported timeouts by our deployment, as previously described in the Section Availability while also taking the measurements from our extensions into account.

RSS availability comparison

The table below shows the availability measurement results as created by the reporting script of our and ICANN's deployment for August 2024. Note that the table also reports the pass/fail calculations which are not part of the public part of the report.

Our measurement platform sent 148,512 measurements, compared to 150,808 measurements by ICANN. The difference can be explained by the fact that ICANN has three more measurement vantage points.

Letter	Test		Deployment	Pass/Fail	Pass/Fail Rating
a	IPv4	UDP	Consortium	Pass	99.17%
	IPv4	UDP	ICANN	Pass	99.83%
	IPv6	UDP	Consortium	Pass	99.34%
	IPv6	UDP	ICANN	Pass	99.88%
b	IPv4	UDP	Consortium	Pass	99.25%
	IPv4	UDP	ICANN	Pass	99.88%
	IPv6	UDP	Consortium	Pass	99.39%
	IPv6	UDP	ICANN	Pass	99.88%
c	IPv4	UDP	Consortium	Pass	99.22%
	IPv4	UDP	ICANN	Pass	99.85%
	IPv6	UDP	Consortium	Pass	99.41%
	IPv6	UDP	ICANN	Pass	99.85%
d	IPv4	UDP	Consortium	Pass	98.03%
	IPv4	UDP	ICANN	Pass	99.94%
	IPv6	UDP	Consortium	Pass	96.36%
	IPv6	UDP	ICANN	Pass	98.72%
e	IPv4	UDP	Consortium	Pass	99.32%
	IPv4	UDP	ICANN	Pass	99.98%
	IPv6	UDP	Consortium	Pass	98.88%
	IPv6	UDP	ICANN	Pass	99.97%
f	IPv4	UDP	Consortium	Pass	99.28%
	IPv4	UDP	ICANN	Pass	99.94%
	IPv6	UDP	Consortium	Pass	99.82%
	IPv6	UDP	ICANN	Pass	99.93%

Letter	Test		Deployment	Pass/Fail	Pass/Fail Rating
g	IPv4	UDP	Consortium	Pass	98.70%
	IPv4	UDP	ICANN	Pass	99.62%
	IPv6	***UDP***	***Consortium***	***Fail***	***86.42%***
	IPv6	***UDP***	***ICANN***	***Fail***	***84.62%***
h	IPv4	UDP	Consortium	Pass	99.26%
	IPv4	UDP	ICANN	Pass	99.93%
	IPv6	UDP	Consortium	Pass	99.23%
	IPv6	UDP	ICANN	Pass	99.83%
i	IPv4	UDP	Consortium	Pass	99.23%
	IPv4	UDP	ICANN	Pass	99.64%
	IPv6	***UDP***	***Consortium***	***Fail***	***94.94%***
	IPv6	UDP	ICANN	Pass	96.78%
j	IPv4	UDP	Consortium	Pass	99.26%
	IPv4	UDP	ICANN	Pass	99.90%
	IPv6	UDP	Consortium	Pass	99.41%
	IPv6	UDP	ICANN	Pass	99.91%
k	IPv4	UDP	Consortium	Pass	99.26%
	IPv4	UDP	ICANN	Pass	99.86%
	IPv6	UDP	Consortium	Pass	99.46%
	IPv6	UDP	ICANN	Pass	99.83%
l	IPv4	UDP	Consortium	Pass	99.25%
	IPv4	UDP	ICANN	Pass	99.80%
	IPv6	UDP	Consortium	Pass	99.44%
	IPv6	UDP	ICANN	Pass	99.84%
m	IPv4	UDP	Consortium	Pass	99.22%
	IPv4	UDP	ICANN	Pass	99.80%
	IPv6	UDP	Consortium	Pass	99.70%
	IPv6	UDP	ICANN	Pass	99.77%

Table: RSI availability metrics 2024-08, measured and reported by ICANN and this consortium.

Our implementation reports a lower availability than ICANN's implementation for 25 of 26 measurements. In two cases, our implementation reported a failure rate below 96%, one of which was also reported by ICANN's implementation.

Both deployments report a RSS availability below the 99.999% threshold, with lower a pass/fail rating reported by our deployment.

	Test	Pass/Fail Rating	Pass/Fail	Measured
Deployment				
Consortium	IPv4 UDP	99.515%	Fail	1,913,489
ICANN	IPv4 UDP	99.994%	Fail	1,957,421
Consortium	IPv6 UDP	99.960%	Fail	1,888,789
ICANN	IPv6 UDP	99.998%	Fail	1,928,542

Table: RSS availability metrics 2024-08, measured by ICANN and this consortium

Classifying timeouts

Also with timeouts reported by our deployment it is not clear whether they were caused by problems at the RSI, at the measurement vantage point or the network in between.

We apply a similar method as in the previous sections to differentiate between the timeouts, but do not include RIPE Atlas measurements. Instead, we monitor the availability of the vantage points using traceroute measurements towards non-DNS related targets and smokeping.

Timeout overview

69.9% of the timeouts reported by our deployment occurred via IPv6. 66% of the timeouts were reported for only three RSIs, with 37.1% of the timeouts reported for one single RSI. Also across our vantage points timeouts are spread unevenly: 3 vantage points report 48.2% of the timeouts. Additionally, 3 of our vantage points report almost an equal number of timeouts for all RSIs, indicating problems with the vantage point or the network instead of the RSIs.

For 50.6% of the timeouts, the vantage point observing the timeout observed at least one other timeout at another RSI. In contrast, 2.2% of the timeouts were observed by two vantage points independently at the same site.

Improving measurement accuracy

We study whether the additional measurements introduced in our implementation can help to differentiate between problems at the vantage point, the network or the RSI.

Additional traceroute measurements

Of the 16,902 timeouts reported on IPv4, 9,221 (54.5%) also had traceroute measurements that failed to reach the services by Google and Cloudflare in the same measurement interval. All of those timeouts were measured by one single vantage point, indicating connectivity issues on IPv4 at the vantage point, instead of problems at the RSI.

Only 18 of the 39,278 timeouts measured on IPv6 occurred at the same time as failed traceroute measurements towards Google and Cloudflare.

Smokeping measurements

For our ICMP ping measurements, we aggregate the measurement results towards each vantage point every 5 minutes and calculate the average packet loss rate across this time window. For 98.9% of all measurement intervals the average loss rate is 10% or lower.

We count how many observed timeouts correlate with smokeping measurements that report an average loss rate above 10%. This is the case for 6,922 (41.0%) of the timeouts on IPv4 and 3,683 (9.4%) of the timeouts on IPv6. Again, the vantage point from the previous section stands out, confirming that the vantage point or the network of the vantage point had problems.

Adjusted root server metrics

We classify timeouts as likely a vantage point issue if

- the timeout was NOT observed by multiple VPs at the same time, and
- the vantage point has NOT observed a routing loop towards the RSI, and
- the vantage point observed multiple timeouts at the same time, and
- either the vantage point observed reachability problems towards Google and Cloudflare,
- we observed reachability problems from the collector towards the vantage point.

20.82% of the observed timeouts fulfill these requirements.

We classify timeouts as likely a network issue if

- the timeout was NOT observed by multiple VPs at the same time, and
- they are NOT classified as a vantage point issue, and
- we have observed a routing loop towards the RSI.

0.03% of the observed timeouts fulfill these requirements.

If we treat timeouts that fall into one of these classes as successful queries and recalculate the RSS availability metrics then the availability improves from 99.515% to 99.993% (IPv4) and from 99.960% to 99.962% (IPv6).

Takeaways

Also our deployment reports a sub-par RSS availability, but also with our measurements it remains unclear whether this was actually caused by problems at the RSS or by problems at the vantage point or the network. For example, we found with high confidence that at least 20.85% of the timeouts observed by our deployment are not caused by the servers of the RSS. By adding additional measures to monitor the availability of the vantage points themselves, we can increase the confidence in our measurements.

Recommendations

For measuring the RSSAC047 metrics in general we recommend discussing the following extensions and improvements.

Vantage points

Monitoring

When going forward with the current form of the measurement platform, we recommend adding additional measures to monitor whether the vantage point itself works flawlessly. In case that there are signs that the quality of the measurements by a vantage point might be impacted in any way, the measurement results from this vantage point should be ignored.

First, we recommend measuring the vantage points externally. As we have shown in Independent measurements of the RSSAC047 root metrics, tools like smokeping could already give insights into the availability and performance of individual vantage points.

Second, measurement vantage points should initiate additional measurements to test whether they are stable and well connected. Currently, the vantage points only initiate traceroute measurements towards the root servers. We recommend adding non-DNS based measurements to services unrelated to the RSS. In our own deployment of the initial implementation we performed traceroutes to the public DNS services by Cloudflare and Google.

Footprint and location

We believe that adding more vantage points to the measurement platform could increase the confidence in the collected measurements. A timeout observed from multiple vantage points is less likely caused by local problems. Additionally, a larger footprint would increase the chance that vantage points would pick up on publication delays that do not affect all sites from a RSI but only a subset.
However, we acknowledge the fact that finding a reliable and diverse set of vantage points can be challenging. For this reason, RSSAC could consider relying on existing measurement platforms like RIPE Atlas. However, RIPE Atlas has its own limitations, some of which we discussed in this document.

Furthermore, it might be desirable to select vantage points that are in close vicinity to the sites of the RSIs. This could remove the impact of routing and latency problems on the collected metrics. That being said, we could not find a link between a large number of hops towards the sites and the number of timeouts observed by a vantage point (see Correlating timeouts with traceroute measurements).

Publication delay

As described in Adjusted publication delay metric, we recommend taking missing zone files into consideration and to reevaluate the metric to calculate the publication delay.

Similar to Section 6.1 - RSS Availability in the RSSAC047v2 document, different failure scenarios should be described. Those not only could act as test cases for developers of the implementation of the measurement software, but also show how different failure scenarios would affect the metric.

Implementation

Besides the recommendations listed in Analysis of the initial implementation, we recommend the following changes and additions to the implementation of the RSSAC047 measurement software.

First, the software has the IP addresses of the RSIs hard-coded. As a consequence, it cannot react to address changes automatically. This issues is also currently being discussed by a work party of the RSSAC Caucus[17]. The measurement software could perform priming queries or could fetch an up to date list of root servers from IANA. The work party currently still debates how the measurement platform should treat decommissioned IP addresses.

Second, the initial implementation performs traceroute measurements towards each RSI every 5 minutes. In case the trace observes 5 unresponsive hops, the trace terminates. This could decrease the usefulness of the traceroute measurements and we recommend increasing the number of unresponsive hops before the trace terminates.

---

17 https://community.icann.org/display/RSSAC/RSSAC+Caucus+Guidelines+for+Changing+Root+Server+Addresses+Work+Party

Discussion and conclusion

Measuring the availability of a highly distributed system like the RSS externally is challenging. Many factors can influence the measurement results like the availability of the measurement vantage point and the network path in between. It is probably even more challenging to judge the quality of the measurements after the fact.

We believe that by combining the measurements of tens or even hundreds of vantage points, we can draw stronger conclusions about the availability of the RSIs and the RSS as a whole. We conclude that the availability of the RSIs and RSS reported by initial implementation is likely too low. However, it is not possible to calculate the availability more accurately retrospectively.

Appendix

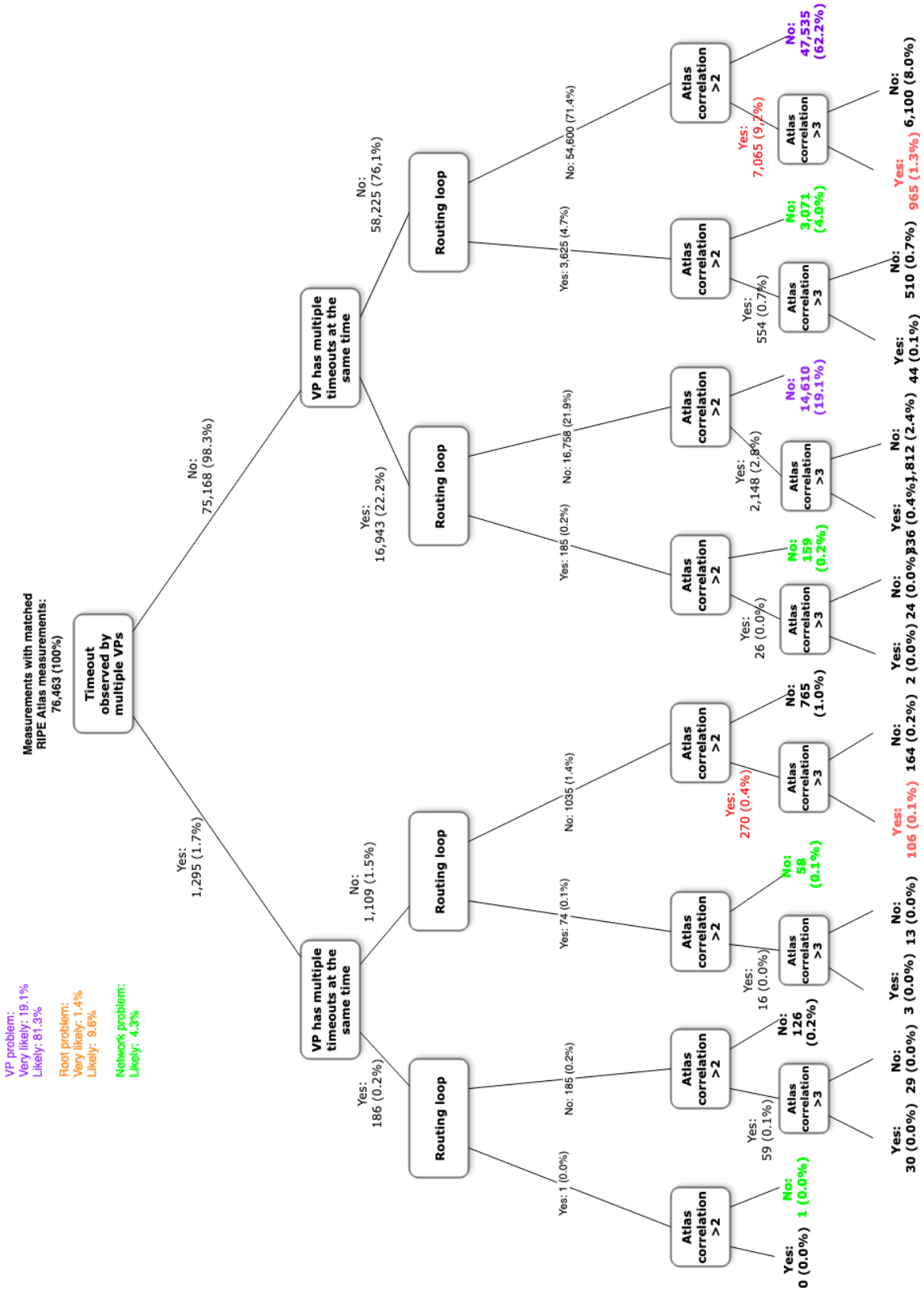


Figure: (Large) Classification of individual timeouts into the cause likely being a problem of measurement vantage points or of an RSI.