RSSAC057: Requirements for Measurements of the Local Perspective on the Root Server System

A Report from the ICANN Root Server System Advisory Committee (RSSAC)
9 September 2021

Preface

This is an Advisory to the Internet Corporation for Assigned Names and Numbers (ICANN) Board of Directors and the Internet community more broadly from the ICANN Root Server System Advisory Committee (RSSAC). In this Advisory, the RSSAC defines requirements for measurements of the local perspective on the root server system.

The RSSAC seeks to advise the ICANN community and Board on matters relating to the operation, administration, security and integrity of the Internet's root server system. This includes communicating on matters relating to the operation of the root servers and their multiple instances with the technical and ICANN community, gathering and articulating requirements to offer to those engaged in technical revisions of the protocols and best common practices related to the operational of DNS servers, engaging in ongoing threat assessment and risk analysis of the root server system and recommend any necessary audit activity to assess the current status of root servers and root zone. The RSSAC has no authority to regulate, enforce, or adjudicate. Those functions belong to others, and the advice offered here should be evaluated on its merits.

A list of the contributors to this Advisory, references to RSSAC Caucus members' statement of interest, and RSSAC members' objections to the findings or recommendations in this Report are at the end of this document.

Table of Contents

1 Introduction

1.1 Goals / Purpose

The DNS Root Server System (RSS) has over 1000 instances deployed all over the world in an effort to provide fast, reliable service to the Internet. There may, however, be certain locations or points on the Internet where the level of service does not seem as good as others due to one reason or another. The RSSAC wishes to have a tool or set of tools that can easily measure the local perspective of the RSS at various locations, or points, on the Internet. This allows Internet users to share measured data from their network perspective and help inform root server operators (RSOs) where to deploy new instances for greater global coverage. The tool(s) should collect enough information to identify some of the reasons why the location is performing at the measured level. Analysis of the collected data is open-ended and not described in this document.

1.2 Scope

This document:

- Defines a set of measurements that can determine the level of service provided by the RSS at a location (e.g., latency, reliability). This is complementary to the work described in RSSAC047 which measures the performance of the RSS versus users' perspectives.[1]
- Identifies a target audience of these tools such that data gathered helps identify topological strengths and weaknesses in the distribution of root server instances.
- Identifies a method to collect data measured by tools for research or reporting to RSOs so that appropriate actions can be considered, including privacy concerns and protections.
- Describes benefits of these tools to users to encourage them to run the tools and participate in the data collection process. Such benefits could include seeing their own data compared to global statistics, and providing valuable information that could improve root zone service in their network location.

2 Use Cases

In this section, some use cases are described for a measurement tool that focuses on the operation and use of the RSS. There may be many more potential use cases. It is expected that the tool would be run several times over a finite period of study to answer questions or characterize a particular vantage point. Each use case describes what types of analyses are desired and a set of supporting measurements. The details and procedures of such analyses are open-ended and are not described in this document.

2.1 Informing the Determination of Underserved Areas

The RSS has relatively good global coverage, but the RSOs are still interested in deploying additional instances, particularly to areas that might be considered underserved. An underserved area has a reasonably-sized user base that perceives poor performance from the RSS due to its closest root server instances being topologically distant, resulting in a perceived higher latency or lower availability of the service. This is a subjective designation and it is not reasonable to

[1] See RSSAC047: RSSAC Advisory on Metrics for the DNS Root Servers and the Root Server System

expect that a tool could directly measure it against some users' perceptions. The metric described in this section only informs a decision on identifying an underserved area which could, in turn, inform decisions in placing new root server instances. Multiple measurements run in diverse topological/geographical locations will be required to inform those decisions.

The goal of these measurements is to assess the performance of the RSS at a local measurement point and compare it to control measurement points. A local measurement that is worse than others may be an indicator of an underserved area. Availability and latency are the primary measurements. Since a resolver only needs a small set of well-performing RSIs for sufficient service, only the lowest few (maybe three) latency measurements should be considered in determining an area to be underserved. Measurements should be direct queries to RSIs (not via a recursive resolver) from a network location that is typical of a recursive resolver placement and should be repeated periodically (e.g., hourly).

The measurements of interest are:

1. Latency - High latencies are potential signs of an underserved area.
2. Availability - This should be measured such that, even with higher timeout values, availability for lower timeouts can be inferred.
3. RSI instance identification - Identifying which instance
4. of an RSI can help identify routing inefficiencies or explain latency results. It is recommended to query for the hostname.bind record to determine this identification.
   Example: dig @<RSI> CHAOS TXT hostname.bind
5. Reference latency measurements - To establish a baseline point of reference, DNS queries should be sent to the well-known open recursive services specified in Section 3.2. DNS queries should be for the root NS RRset ("./IN/NS"). Reference latency measurements are intended to be aggregated to characterize the "last mile" connectivity of the measurement source, and are not intended for direct comparison to individual root server identities. Interpretation of any reference latency measurements is left to the party performing the data analysis.
6. Path information - Traceroutes should be conducted for each RSI to get an idea of path and hop count. This can help identify slow, unreliable, or inefficient links or routes.

2.2 Evaluating Third Party Requests to Host an Anycast Instance

RSOs periodically receive offers to host, or suggestions to place, anycast instances at new locations. Such offers could come from an ISP, IXP, or other network operator. The requestor states they would like faster or more redundant connectivity to the RSS, but may be unable to elaborate further on precisely what that means. While each RSO will have different criteria for whether to accept such a proposal, including many non-technical factors, it is desired to have a set of measurements that can inform such a decision. The RSO may ask the requestor to run this tool to collect data that can be used in the evaluation.

If the proposed location already experiences comparatively good service (latency and availability) from the RSS, then there could be less value in placing a new instance there. The

size of the anycast catchment at the location could inform the ability of a location to sink a denial of service (DoS) attack or to shift load to other anycast instances.

The measurements of interest are:

1. Latency - Low latencies to existing anycast instances for all RSIs might lead to a lower perceived need for an instance at this location. Interpretation of latency values is subjective.
2. Availability - As with latency, high availability of a sufficient set of RSIs may lead to lower perceived need for placement at this location.
3. RSI instance identification - Identifying the particular instances that respond to queries from this location can help identify routing inefficiencies or explain latency results. It is recommended to query for the hostname.bind record.
   Example: dig @<RSI> CHAOS TXT hostname.bind
4. Anycast catchment - A large number of potential users of a proposed anycast instance would increase the value of such a placement. It is out of scope for the proposed tool to implement such a measurement. However, VerfPloeter is well-suited to measure this independently. (See Section 4.5)

2.3 Recursive Operator

An enterprise, ISP, or other organization that operates one or more recursive resolvers may want to better understand the performance characteristics of the RSS from their perspective. This could help inform them of network optimizations, such as routing changes, that could improve the performance of the RSS from their perspective. While performance of the RSS is only one consideration for the recursive operator, it is the hope that this data could be useful in their decision process.

The measurements of interest for a recursive are:

1. Availability - Measurements should be made such that availability of individual RSIs can be determined, given a sufficient number of attempted measurements. The recursive operator may use this data to find locations where availability of at least some RSIs are above some threshold of availability.
2. Latency - Latency is a relative low concern for recursive operators due to high TTLs and relatively low number of queries that result in positive answers. However, queries for non-existent TLDs at volume have the potential to indicate problems that could be mitigated with faster response times. It has also been observed that DNS response times are useful in determining routing changes both locally and globally. This data does not directly impact the root infrastructure but has been helpful in correlating events across multiple networks.

3 Measurements

This section contains the list of the measurements the tool will capture. Each subsection contains an explanation of the measurement, followed by pseudocode using *dig*[2] or *traceroute*.[3] The specific formatting of reported data is left to implementers, however the chosen format should be easily machine parsable. The tool must be capable of publishing results to a repository (see Section 5).

3.1 Timestamp

The tool must record a timestamp when the tool begins operation and again when it finishes operation. This timestamp must use the format described in RFC3339 with a precision of seconds.[4] The separator between *full-date* and *full-time* must be the ASCII upper-case character *T*. The time reported must be in Coordinated Universal Time (UTC) and a trailing ASCII *Z* must be present.

The example below shows a timestamp for September 16, 2030, 2 hours, 18 minutes, and 12 seconds in UTC.

```
2030-09-16T02:18:12Z
```

3.2 DNS Query Latency

The tool must perform DNS queries and measure the time it takes to get a response. The tool must send these queries using IPv4 UDP, IPv4 TCP, IPv6 UDP and IPv6 TCP.

For UDP, the time measured will be the difference in timestamps between the single packet sent and the single packet received. For TCP, the time measured will be the difference in timestamps between the DNS question being sent and receiving all DNS data. TCP setup and tear down time will be excluded from these measurements.

Ten queries must be sent to each destination for each of the four protocol combinations with individual response times recorded for each query. The timeout for each query is one second. The queries may be performed asynchronously. However, the latency measurements must not be influenced by multiple queries executing in parallel.

For each query both the latency measurement in milliseconds and the response data must be recorded. Unexpected responses must register as failures due to bad data, measurements that timeout must register as failures due to timeout, and measurements that return unexpected RCODEs must register as failures due to a bad RCODE.

Three different kinds of queries are to be sent to all RSIs:

1. A single question for *hostname.bind* with a resource record of *TXT,* class *CH,* without

[2] See dig — DNS lookup utility , https://bind.isc.org/doc/arm/9.11/man.dig.html
[3] See traceroute(8) - Linux man page, https://linux.die.net/man/8/traceroute
[4] See RFC 3339: Date and Time on the Internet: Timestamps, https://datatracker.ietf.org/doc/rfc3339/

any EDNS0 OPT RR. The expected response is an instance name and the expected RCODE is NoError.

2. A single question for the *.COM* nameservers with a query type of *NS,* query class *IN,* with the *checking disabled (CD)* flag set, and EDNS0 enabled. The expected response is a list of nameservers and the expected RCODE is NoError.
3. A single question for the *.COM DS* resource record, query class *IN*, with the *checking disabled (CD)* and *DNSSEC OK (DO)* flags set. The expected response is a *DS* record and the expected RCODE is NoError.

For each transport, a set of queries should be sent directly to the IP address of each open resolver listed below for the set of NS records for the root zone. The expected RCODE is NoError. The query times for each query should be reported. The exact list of open resolvers is left open to implementors. The examples given below are merely suggestions.

```
foreach [1 .. 10]:
  foreach [ipv4 ipv6]:
    foreach [udp tcp]:
      foreach <RSI IP address> as $RSI:
        dig @$RSI +noedns CHAOS TXT hostname.bind
        dig @$RSI +edns +cd com ns
        dig @$RSI +cd +dnssec DS com

    foreach [CloudFlare Google OpenDNS Quad9] as $REF:
        dig @$REF +noedns IN NS .
```

3.3 Routing and Connectivity

The tool must initiate UDP and TCP traceroutes to each RSI over both IPv4 and IPv6. The UDP packets will be sent to port 53, and the TCP packets will be sent to port 53. Traceroute probe packets should have a timeout value of five seconds. Three probes should be sent for each incrementing time-to-live (TTL) value with a maximum TTL of 32 hops. In addition to a maximum TTL, the tool will halt sending probes if no response is received after five successive TTLs. All intermediate IP addresses and their associated probe packet TTL should be recorded along with the delay in milliseconds for each response.

```
foreach [ipv4 ipv6]:
  foreach <RSI IP address> as $RSI:
    traceroute -n $RSI -m 32 -U 53
    traceroute -n $RSI -m 32 -T -p 53
```

3.4 IP addresses of measurement source

For analyzing and comparing different vantage points, it will be necessary to have a public IP address of the host running the measurements. Several public, DNS-based “whoami” services can be queried to obtain that address. For each of IPv4 and IPv6, query one of these services until a response with RCODE=NoError is returned. Report the IP address returned in the answer.

```
while RCODE != NoError :
  for ns in $(dig +short akamai.net NS) ; do dig -6 +short @$ns
```

```
whoami.akamai.net AAAA 2>/dev/null && break ; done
  for ns in $(dig +short google.com NS) ; do dig -6 +short @$ns
o-o.myaddr.l.google.com TXT 2>/dev/null && break ; done
  for ns in $(dig +short v6.powerdns.org NS) ; do dig -6 +short @$ns
whoami.v6.powerdns.org AAAA 2>/dev/null && break ; done


While RCODE != NoError :
  for ns in $(dig +short akamai.net NS) ; do dig -4 +short @$ns
whoami.akamai.net A 2>/dev/null && break ; done
  for ns in $(dig +short google.com NS) ; do dig -4 +short @$ns
o-o.myaddr.l.google.com TXT 2>/dev/null && break ; done
  for ns in $(dig +short v4.powerdns.org NS) ; do dig -4 +short @$ns
whoami.v4.powerdns.org A 2>/dev/null && break ; done
```

4 Existing Tools

This section contains short descriptions of existing tools in this space.

4.1 Traceroute

Traceroute is a standard UNIX command-line tool.[5] It is commonly found on UNIX derived operating systems. On Microsoft Windows operating systems it is instead named *tracert*. Its primary use is to determine intermediate gateways between two networked hosts. It accomplishes this by sending packets with an incrementing time-to-live (TTL) and receiving the ICMP time exceeded packets sent by the intermediate gateways.

4.2 Dig

Dig is a DNS diagnostic command-line tool developed and maintained by Internet Systems Consortium (ISC).[6] It is commonly found on many operating systems connected to the Internet. Dig can be used to perform many different types of DNS queries, and supports many options that provide it great flexibility.

4.3 Perf_root

Perf_root is a Python script that can gather connectivity data on the RSS from a local perspective.[7] When started perf_root crawls the DNS root zone for a number of TLDs. It then issues timed queries to each RSI over UDP and TCP, IPv4 and IPv6. Traceroutes are also performed to each root server identity over IPv4 and IPv6. Results of these tests are then output in JSON locally.

[5] See traceroute(8) - Linux man page, https://linux.die.net/man/8/traceroute
[6] See dig — DNS lookup utility, https://bind.isc.org/doc/arm/9.11/man.dig.html
[7] See perf_root, https://github.com/rssac-caucus/perf_root

4.4 ISC's RSS Visualizer

ISC's RSS Visualizer takes IPv4 UDP DNS latency measurements made from the extensive network of RIPE Atlas probes to the closest instance of each RSI,[8] as chosen by the BGP routing algorithm and displays them on a map of the world. The visualisation shows latency by colour, and hovering over a probe's location shows the latency to each RSI and the RSI's geo-code for the associated anycast instance. A slider allows the user to colour the plot based on the N fastest responses, thus showing how many anycast instances are within a nominally acceptable latency performance range.

4.5 Verfploeter

Verploeter is a measuring system that uses IPv4 ICMP packets to determine which IP addresses are using which anycast instance.[9] The system generates ICMP echo request packets with forged source IP addresses to many destination IP addresses. The source IP address in the ICMP echo request is set to that of an anycast node. When the host receiving this ICMP packet generates an ICMP echo response it will be received at one of many anycast nodes. In this manner, Verfploeter is able to determine the IP addresses which are using a specific anycast node. In 2021, Verfploeter was modified to additionally measure latency from each anycast catchment.

5 Public Data Repository

Measurements from the local perspective tool are most useful when they can be aggregated, archived, and later analyzed. The RSSAC Caucus envisions a public data repository for this purpose, to which the tool can automatically submit results. Users of the tool should explicitly decide whether to publish the results to the repository. The resulting archive should be available to researchers, because the RSS will benefit most by maximizing researcher access to the data in order to perform analysis and support improvements to the RSS. The repository shall make reasonable efforts to protect the privacy of data collected from users. The tool documentation shall list what data is published to the repository, how the data will be used, and the privacy measures employed by the repository. The tool should encourage users to publish machine-readable results to the repository. The repository should allow point of contact (POC) information to be associated with measurement results from a vantage point to facilitate potential improvements for the reporting site or to the tool. It is recommended that users provide this POC information when publishing.

It is important that the data repository maintain a high level of data quality. Steps should be taken in both the tool and the data repository to protect against abuse, deception, and data pollution. It is recommended that the tool be regularly run by a set of volunteer organizations that publish their results to establish a baseline of data for comparison.

[8] See ISC's RSS Visualizer, https://atlas-vis.isc.org/
[9] See Verfploeter: Broad and Load-Aware Anycast Mapping, https://www.isi.edu/~johnh/PAPERS/Vries17a.pdf

6 Recommendations

Recommendation 1: The RSSAC recommends that a tool, or set of tools, be built based on the requirements articulated in Section 3 of this document. The tools described in Section 4 of this document could be used as building blocks. The tools should be made available for the Internet community.

Recommendation 2: The RSSAC recommends that the ICANN Board identify a person or group to collaborate with the RSSAC Caucus on further development of a data repository as described in Section 5 of this document. The purpose of such collaboration is to make a specific proposals for a data repository, including:

A. Implementation of the data publication mechanism
B. Whether or not access to measurement results should be public or limited due to privacy concerns
C. How to ensure data quality and prevent abuse
D. A proposed database schema and model
E. A proposed data exchange format (e.g., JSON)
F. Cost estimates for the initial development and ongoing operation
G. Identification of groups or parties that could operate the data repository

7 Acknowledgments, Disclosures of Interest, Dissents, and Withdrawals

In the interest of transparency, these sections provide the reader with information about four aspects of the RSSAC process. The Acknowledgements section lists the RSSAC caucus members, outside experts, and ICANN staff who contributed directly to this particular document. The Statement of Interest section points to the biographies of all RSSAC caucus members. The Dissents section provides a place for individual members to describe any disagreement that they may have with the content of this document or the process for preparing it. The Withdrawals section identifies individual members who have recused themselves from discussion of the topic with which this Advisory is concerned. Except for members listed in the Dissents and Withdrawals sections, this document has the consensus approval of the RSSAC.

7.1 Acknowledgments

RSSAC thanks the following members of the Caucus and external experts for their time, contributions, and review in producing this Report.

RSSAC Caucus members
Jaap Akkerhuis
Fred Baker
Ray Bellis
Marc Blanchet
Kazunori Fujiwara
Shailesh Gupta

Wes Hardaker
Hiro Hotta
Chris Ishisoko
Warren Kumari
Abdulkarim Oloyede (Work Party Leader)
Jeff Osborn
Ken Renard (Work Party Leader)
Joey Salazar
Shinta Sato
Barbara Schleckser
Robert Story
Brad Verd
Duane Wessels
Dessalegn Yehuala

ICANN Staff
Andrew McConachie (editor)
Danielle Rutherford
Ozan Sahin
Steve Sheng

7.2 Statements of Interest

RSSAC Caucus member biographical information and Statements of Interests are available at: https://community.icann.org/display/RSI/RSSAC+Caucus+Statements+of+Interest

7.3 Dissents

There were no dissents.

7.4 Withdrawals

There were no withdrawals.