

Report on the Conclusion of the RSSAC Modern Resolvers Work Party

3 January 2020

The Root Server System Advisory Committee (RSSAC) requested that the RSSAC Caucus investigate how modern recursive resolvers act when deployed on the Internet, particularly with respect to how they interact with the Root Server System (RSS). The request focused on two primary areas; popular resolver implementations in code bases, and deployed resolvers. This document describes the outcome of that work.

Statement of Work

On 07 August 2018, RSSAC issued a Statement of Work for “Studying Modern Resolver Behaviours.”¹ The RSSAC Modern Resolvers Work Party was formed soon thereafter, composed of members of the RSSAC Caucus.

The four tasks given to the Work Party were:

1. Analyze DNS resolver network traffic and behavior to better understand how they operate as they interact with authoritative servers generally and the RSS specifically in terms of preferred root server selection.
2. Analyze DNS resolver and authoritative server code bases and perform (ideally repeatable) simulations to further extract a model of how modern resolvers implement caching and priming of the root name servers.
3. Analyze DNS resolver code bases and perform (ideally repeatable) resolution simulations to further extract a model of how modern resolvers choose which authoritative server for a given zone to query.
4. Analyze DNS resolver systems using multiple resolution instances (with potentially individual or shared caching systems) to understand how they interact with the RSS. (e.g. Google, Cloudflare, quad9 type systems)

Results

Tasks #1 and #4 are related to observing how recursive resolvers on the Internet currently work by sending queries through them and watching how they send queries to authoritative servers. APNIC, who has been collecting such data for many years, gave a presentation on how their test setup works. The Work Party discussed the two tasks but there was insufficient interest to pursue them.

Tasks #2 and #3 required a collection of resolver software implementations and a testbed in which to use them. Caucus member Paul Hoffman created a testbed that would show how

¹ See <https://mm.icann.org/pipermail/rssac-caucus/2018-August/000601.html>

various versions of resolver software do priming and caching of root server information (task #2). The same testbed can also show how those resolvers do server selection (task #3).

The testbed is open source and fully documented and is available through a GitHub repository.² There was limited use of the testbed after it was completed.

The testbed is expandable in many ways:

- Researchers can write their own tests to run in the testbed. For example, writing new tests could simulate what would happen if half of the RSS went offline or whether the RSS would be better off with more root server instances.
- Resolver software interacts with the RSS differently depending on configuration. Researchers can test how different configurations change the ways resolvers behave.
- The testbed can evaluate how different contents of the root zone (such as with different TTLs, different DNSSEC signatures, more or fewer root server operators, and so on) affect caching. A separate software project for creating custom root zones is used with the testbed.³
- The simulated root zones can have variable delays in responding to queries, and those delays can be changed.
- New resolver software appears over time, and that software can be added to the testbed.

The testbed can continue to be developed in the future when there is additional interest from DNS researchers or from the RSSAC.

² See <https://github.com/icann/resolver-testbed>

³ See <https://github.com/icann/test-root>