RSSAXXX
RSSAC Report on Root Zone TTLs

A Report from the ICANN Root Server System Advisory Committee (RSSAC)
8 June 2015

Preface

This is a Report 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 Report, the RSSAC studies the TTLs for the root zone.

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 merit.

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

Table of Contents

Executive Summary 4
1. Introduction 5
2. Terminology 5
2.1. Common Units Conversion 6
3. Current Root Zone TTLs 7
3.1. The SOA Record 7
3.2. DNSSEC Signature Validity Periods 8
4. Reasons to Consider Changing Root Zone TTLs 8
5. History of TTLs in the Root Zone 9
5.1. DNS-OARC Archive (1999) 10
5.2. BIND Source Code 10
5.3. Individual Testimonial – Mark Kosters 11
6. Empirical Analysis 11
6.1. Survey of TTLs of TLDs 12
6.1.1. Delegated vs. Authoritative NS TTLs 12
6.1.2. DS vs. DNSKEY TTLs 14
6.2. Survey of “max-cache-ttl” parameters in popular recursive resolver implementations 16
6.3. DITL Analysis 18
6.4. Interactions between SOA refresh and DNSSEC Signature Validity 23
6.4.1. Validity Period for ZSK Signatures 24
6.4.2. Validity Period for KSK Signatures 25
6.4.3. Analysis of Proposed Remedies 26
7. Findings 27
8. Recommendation 29
9. Future Work 29
10. Acknowledgments, Disclosures of Interest, Dissents, and Withdrawals 31
10.1. Acknowledgments 31
10.2. Statements of Interest 31
10.3. Dissents 32
10.4. Withdrawals 32

Executive Summary

Root zone TTLs have not changed since 1999. In this report, the RSSAC caucus studies the extent to which the current root zone TTLs are still appropriate for today's internet environment.

Selecting a TTL for a given resource record involves finding the right balance between a few tradeoffs. Intuitively, shorter TTLs are beneficial for data that changes frequently, whereas longer TTLs are beneficial for data that is relatively stable. Related to this, longer TTLs provide robustness in the event of operational failures. All other things being equal, shorter TTLs generally result in higher query rates, and longer TTLs result in lower query rates.

Through a series of empirical data analysis, the RSSAC Caucus finds that:

- The root zone delegation TTLs are appropriate for today's environment.
- Root zone TTLs values could be reduced to 1 day without any significant impact on the amount of traffic to root servers.
- Increasing root zone TTLs should only be done with careful consideration of DNSSEC-related implications.
- Root zone TTLs appear to not matter to most clients.
- Few reasons exist today to consider changes to root zone TTLs.
- Two potential problems related to the interaction between the SOA Expire value and the root zone's signature validity periods exist. These need to be addressed by the Internet Community.

The RSSAC Caucus makes the following recommendations:

- To address the DNSSEC problems identified in Section 6.4, the RSSAC should choose one or a combination of mitigation options (*e.g.* reducing the SOA Expire Time, reducing NS RRSet TTL, increase the KSK-generated Signature Validity Time, etc.), after carefully considering their operational implications.
- No other changes to Root Zone TTLs should be made at this time.

1. Introduction

Resource Records (RRs) in the DNS are given a Time to Live (TTL) value, which specifies the amount of time the data may be stored in a cache. When the TTL for a cached resource record expires, the caching name server must contact the authoritative name server again to receive up-to-date data. TTL values are generally given in seconds. In this report we may also refer to TTLs in hours or days for convenience.

Selecting a TTL for a given resource record involves finding the right balance between a few straightforward tradeoffs. Intuitively, shorter TTLs are beneficial for data that changes frequently, whereas longer TTLs are beneficial for data that is relatively stable. Related to this, longer TTLs provide robustness in the event of operational failures. All other things being equal, shorter TTLs generally result in higher query rates, and longer TTLs result in lower query rates.

In addition to straight TTL values of resource records, this report discusses a number of other TTL-related parameters, such as those in the Start of Authority (SOA) record, and the signature validity periods found in DNSSEC Resource Record Signature (RRSIG) records.

The remainder of this document is specific to TTLs in the root zone. It is organized in the following manner: Section 2 provides terminology; section 3 talks about current TTLs and other parameters; section 4 presents reasons to consider changing root zone TTLs; section 5 explores the history of root zone TTLs; section 6 discusses a set of four empirical analyses to answer the study questions; section 7 presents the findings; section 8 details the recommendations; section 9 covers future work items; and section 10 includes acknowledgements, disclosures, etc.

2. Terminology

Authoritative server -- A system that responds to DNS queries with information about zones for which it has been configured to answer with the AA flag in the response header set to 1. It is a server that has authority over one or more DNS zones.

Authoritative data -- All of the RRs attached to all of the nodes from the top node of the zone down to leaf nodes or nodes above cuts around the bottom edge of the zone. (Quoted from Section 4.2.1 of RFC 1034) It is noted that this definition might inadvertently also include any NS records that appear in the zone, even those that might not truly be authoritative because there are identical NS RRs below the zone cut. This reveals the ambiguity in the notion of authoritative data, because the parent-size NS records authoritatively indicate the delegation, even though they are not themselves authoritative data.

Delegation -- The process by which a separate zone is created in the name space beneath the apex of a given domain. Delegation happens when an NS RRset is added in the parent zone for the child origin, and a corresponding zone apex is created at the child origin.

Glue records -- Resource records which are not part of the authoritative data, and are address resource records for the servers listed in the message. They contain data that allows access to name servers for subzones. (Definition from RFC 1034, section 4.2.1)

Resource Record Set (RRset) -- A set of resource records with the same label, class and type, but with different data. (RFC 2181)

Resource Records Signatures (RRSIG) -- DNSSEC uses public key cryptography to sign and authenticate DNS resource record sets (RRsets). Digital signatures are stored in RRSIG resource records and are used in the DNSSEC authentication process described in (RFC4035).

Round Trip Time (RTT) – The round trip time of a DNS query is the measurement of time between a DNS query being issued and the time the answer is received.

Time to Live (TTL) -- The maximum "time to live" of a resource record. A TTL value is an unsigned number, with a minimum value of 0, and a maximum value of 2,147,483,647. The TTL "specifies the time interval that the resource record may be cached before the source of the information should again be consulted". (RFC 1035)

Key signing key (KSK) -- DNSSEC keys that only sign the apex DNSKEY RRset in a zone. (RFC 6781)

Zone signing key (ZSK) -- DNSSEC keys that can be used to sign all the RRsets in a zone that require signatures, other than the apex DNSKEY RRset. (RFC 6781)

2.1. Common Units Conversion

Table 1: Common units conversion

60 seconds	1 minute
300 seconds	5 minutes
900 seconds	15 minutes
1800 seconds	30 minutes
3600 seconds	1 hour
86400 seconds	1 day
172800 seconds	2 days

518400 seconds	6 days
604800 seconds	7 days

3. Current Root Zone TTLs

Today, records in the root zone have the following TTL values:

Table 2: TTL values for Resource Records in the Root Zone. Highlighted text (*) represents resource records new to the DNSSEC signed root zone.

Resource Record	Type	TTL
Root SOA	authoritative	1 day
Root DNSKEY*	**authoritative**	**2 days**
Root NS	authoritative	6 days
Root Glue (A, AAAA)	glue	6 days
Root NSEC*	**authoritative**	**1 day**
TLD NS	delegation	2 days
TLD Glue (A, AAAA)	glue	2 days
TLD DS*	**authoritative**	**1 day**

Highlighted(*) rows represent records specific to DNSSEC. Not listed, however, are RRSIG records, which always have a TTL matching the record type they cover. Note that glue TTLs match their associated NS TTLs. Most authoritative data in the root zone has 1 day TTLs, except for the root zone NS and DNSKEY RRsets.

In these document, we use the term “root zone TTLs” to collectively refer to the TTL values for Resource Records in the root zone.

3.1. The SOA Record

The Start-Of-Authority (SOA) record includes a number of time-related fields as well. The root zone SOA record has these values:

Table 3: Time related values for root zone Start-of-Authority Record

Field	**Value**

Refresh	30 minutes
Retry	15 minutes
Expire	7 days
Minimum	1 day

The SOA Expire field specifies how long a secondary authoritative name server may serve its data after losing contact with the primary server.

The SOA record also determines a zone's negative cache TTL. A negative response happens either when the queried name does not exist, or when there is no data of the requested type for a name. RFC 2308 clarifies that the negative cache TTL is computed as the minimum of two values: the SOA TTL, and the SOA Minimum field. For the root zone, both of these values are set to 1 day, which means it has a negative caching TTL of 1 day. However, as a practical matter, some popular implementations enforce a lower limit on negative caching TTLs by default. See Section 6.2 ("Survey of "max-cache-ttl" parameters in popular recursive resolver implementations").

3.2. DNSSEC Signature Validity Periods

In DNSSEC, the signatures stored in RRSIG records are given a certain validity period. That is, the time over which the signature can be considered valid. Whereas TTL values are relative (i.e., from the time a record enters a cache), the signature validity is expressed as absolute start and end times. In this document, however, we'll refer to the signature validity period as the time between the start and end values.

Table 4: DNSSEC Signature Validity Period for the Root Zone

Signatures Generated By	Covering Record Types	Validity Period
KSK	DNSKEY	15 days
ZSK	All others	10 days

4. Reasons to Consider Changing Root Zone TTLs

In general, a DNS zone operator selects particular TTLs to strike a balance amongst a number of tradeoffs. The most obvious tradeoff is the desire to have changes propagate quickly, versus the desire to let data remain in resolver caches.

Caching is beneficial in three important ways: (1) it improves performance by reducing latency for the end user; (2) it can reduce load on authoritative servers, and; (3) it can help users survive certain network partitions (i.e., when some name servers may be unreachable).

Unlike many DNS zones, the root zone rarely has a need for the quick propagation of changes. Rather, changes to the root zone are made slowly and deliberately. Delegations (TLDs) are added well in advance of queries from end users. Root name servers themselves are renumbered infrequently and with great care and planning.

Nonetheless, here we enumerate some of the reasons that parties to the root zone system may give for changing TTLs:

1. To affect a change--increase or decrease--in traffic between root servers and recursive name servers.
2. To have root zone changes take effect more quickly.
3. To change the amount of time that a recursive name server can function without communicating with a root name server. Scenarios include denial-of-service attacks and outages due to natural disasters.
4. To tie the TTLs of particular RRsets to other RRsets (see example below).
5. To work around software bugs in resolver implementations (see example below).
6. To alter timing interactions related to DNSSEC records (see example below).
7. To remain compliant with IETF standards documents that may change from time-to-time.
8. To remain compliant with IANA policies that may change from time-to-time.

In 2014, Verisign identified a potential issue with DNSSEC signatures in the root zone. At the time, the signature validity period over the NS RRset was seven days, while the NS RRset TTL was six days. This did not provide a sufficient “buffer” in the event a root server instance failed to refresh the zone. A DNSSEC validator forwarding queries through a non-validator might receive only stale signatures in this situation. At Verisign’s request, and with the approval of RSSAC and other stakeholders, the signature validity period for all signatures generated by the ZSK was increased from seven to ten days.[1] Please see section 6.4.1 below for a discussion of why even ten days may not be sufficient.

5. History of TTLs in the Root Zone

One of the questions the study team was asked to explore is: Are the TTLs currently in the root zone appropriate for today’s root server system and today’s overall Internet?

[1] https://www.icann.org/en/system/files/files/rssac-dnssec-validity-root-zone-17dec14-en.pdf

To assist in answering this question, it may be helpful to take a look back in time. For example, it may be helpful to know if the TTLs were different in the past and then changed for some reason? Or have they always been the same, since the very start?

5.1. DNS-OARC Archive (1999)

DNS-OARC has perhaps the best archive of historical root zone files, thanks to contributions from a number of its members and others in the DNS community.[2] The DNS-OARC archive dates back to May 31, 1999.

In that zone published with serial number 1999053100, the TTLs and the time-related SOA fields are the same as today's values. In other words, root zone TTLs have not changed since May 31, 1999.

5.2. BIND Source Code

Internet Systems Consortium still makes old versions of the BIND software available for download via their FTP server.[3] The BIND software, as well as others, includes a copy of the root zone "hints" file to bootstrap the resolution process. By looking at old copies of the source code, we can look back a little farther in time.

The oldest BIND source code available is for version 4.9.2, dating back to around December 1993. In this package is found the following "conf/root.cache" file:

```
;; QUESTIONS:
;;   ., type = NS, class = IN

;; ANSWERS:
.    518400    NS   NS.INTERNIC.NET.
.    518400    NS   AOS.ARL.ARMY.MIL.
.    518400    NS   KAVA.NISC.SRI.COM.
.    518400    NS   C.NYSER.NET.
.    518400    NS   TERP.UMD.EDU.
.    518400    NS   NS.NASA.GOV.
.    518400    NS   NIC.NORDU.NET.
.    518400    NS   NS.NIC.DDN.MIL.

;; ADDITIONAL RECORDS:
NS.INTERNIC.NET. 518400  A    198.41.0.4
AOS.ARL.ARMY.MIL.   518400    A    128.63.4.82
AOS.ARL.ARMY.MIL.   518400    A    192.5.25.82
KAVA.NISC.SRI.COM.  518400    A    192.33.33.24
```

[2] https://www.dns-oarc.net/oarc/data/zfr/root

[3] ftp://ftp.isc.org/isc/

```
C.NYSER.NET.   518400    A    192.33.4.12
TERP.UMD.EDU.  518400    A    128.8.10.90
NS.NASA.GOV.   86400 A   128.102.16.10
NS.NASA.GOV.   86400 A   192.52.195.10
NIC.NORDU.NET.   518400  A    192.36.148.17
NS.NIC.DDN.MIL.  518400  A    192.112.36.4

;; FROM: gw.home.vix.com to SERVER: ns.nasa.gov
128.102.16.10
;; WHEN: Sun Dec 19 13:42:51 1993
;; MSG SIZE  sent: 17  rcvd: 402
```

Unfortunately, the root.cache file can't tell us anything about the delegations or the authoritative SOA record; but at least we can see that the root zone glue records had 6 day TTLs as far back as 1993.

Note that the A records for NS.NASA.GOV have a TTL of 1 day, which is different from all the others. The root.cache file was generated by sending a query to NS.NASA.GOV (as shown in the comments), which was almost certainly authoritative for both the root and nasa.gov zones, where the record had the lower TTL.

5.3. Individual Testimonial – Mark Kosters

Mark Kosters worked at Network Solutions from 1991 to 1993 in support of the NIC-DDN contract awarded by DISA. Among his duties was to maintain and publish the root zone file, maintain A and J root-servers, and create/maintain source code that generated the root zone. He continued those responsibilities working for NSI as he moved to the Principle Investigator to NSF for the InterNIC project in 1993. Mark continued to work for Network Solutions as it was acquired first by SAIC, then later VeriSign until 2007.

Upon our request, Mr. Kosters was able to check his archived documents, messages, and source code. He confirms that, as far back as 1991, TTLs in the root zone were: 6 days for authoritative data, 2 days for delegations, and 2 days for glue.

6. Empirical Analysis

In order to improve our understanding of how root zone TTLs affect (or do not affect) the wider DNS as a whole, we undertook a number of surveys and conducted a few experiments:

1. **Survey of TTLs used in TLD zones.** While there is no requirement that delegation and authoritative TTLs should match, a comparison of delegated vs. authoritative NS TTL provides some insight on what TLD operators believe are appropriate values for NS TTLs.

2. **Max-cache parameters of common recursive implementations.** Software such as BIND and Unbound place limits on the amount of time that data may remain cached. If the majority of recursive software has limits that are lower than most root zone TTLs, we might expect that lowering TTLs close to those limits would have very little impact on root server traffic.
3. **An analysis of Day-In-The-Life (DITL) of the Internet data.** Here we analyze the 2014 DITL data to answer the question, "How often do root servers see queries for the same TLD from the same IP address?" If caching is working well, the time between queries for the same TLD should be relatively large.
4. **Interactions between SOA refresh and DNSSEC signature validity.**

6.1. Survey of TTLs of TLDs

The survey is divided into two parts. First, study team compared the TTLs of NS delegations in the root zone to the TTLs in the authoritative TLD zones. Second, we compared the TTLs of DS records in the root zone to the corresponding DNSKEY records in the authoritative TLD zones.

6.1.1. Delegated vs. Authoritative NS TTLs

As stated earlier, delegation NS records in the root zone are all given a 2-day TTL. However, NS records also appear in the delegated zone (i.e., TLD), where they are authoritative, and are set by each TLD per its registry policy. There is no requirement for the TTLs to match.

RFC 2181 (section 5.4.1) describes ranking of data from different sources. Data from the authority section of an authoritative answer (the TLD) ranks higher than the data from the authority section of a non-authoritative answer (root zone referral). We generally expect implementations to replace lower-ranking data with higher-ranking data in their caches (although RFC 2181 could be clearer on this subject).

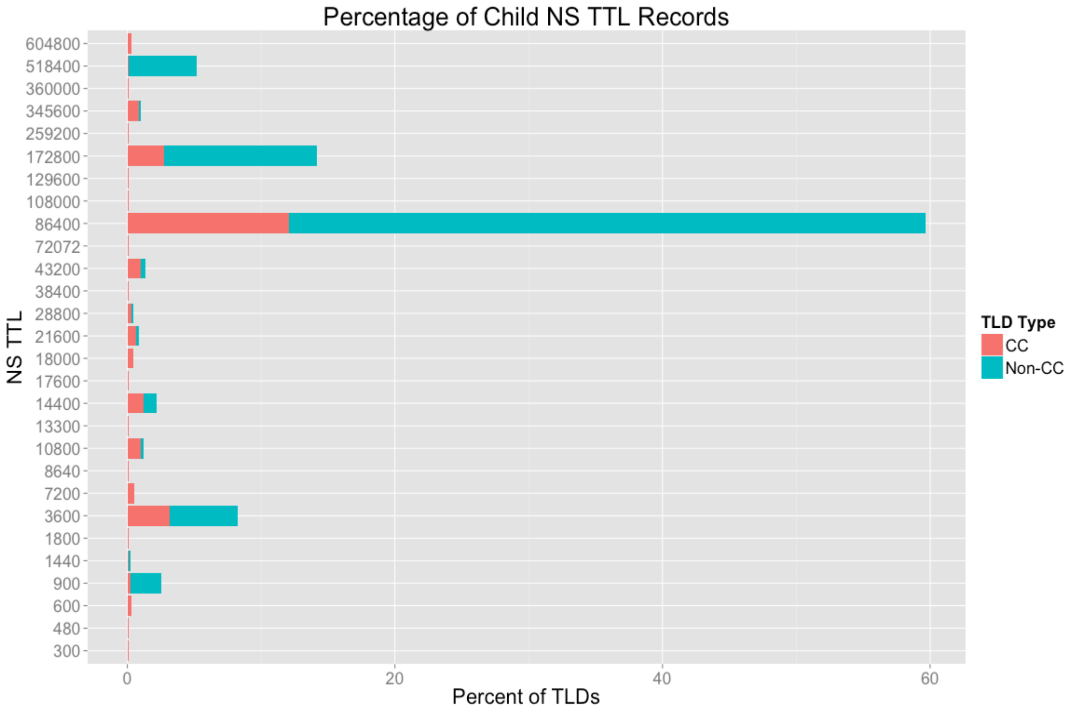

Figure 1: Percentage of Child NS TTL Records

On May 5, 2015, the study team surveyed all TLDs (923 in total) present in the root zone, and analyzed the NS RRset TTL in each zone. The results are presented in Figure 1 above.

From Figure 1 we can see that 1 day (86400 seconds) is the most common authoritative TTL value. Nearly 60% of TLDs use this value. The next most common value is two days (172800 seconds), used by roughly 16% of TLDs. The third most common value is one hour (3600 seconds), used by roughly 8% of the TLDs. Overall, 80% of TLDs have authoritative NS TTLs greater than or equal to 1 day. During our study, 13 TLDs (8 ccTLDs and 5 IDN TLDs) were unreachable due to timeouts or other errors.

If all caching name servers implement the data ranking and cache replacement methods described in RFC 2181, it is reasonable to expect that only these authoritative NS TTLs would affect cache expiration and, therefore, the rate at which clients query root name servers.

The graph below shows each TLD’s authoritative NS TTL (y-axis) and its query count in the 2014 DITL data (x-axis). While there are certainly a large number of less-popular TLDs with large RTTs, by looking at the lower right section of the graph we can see that the more-popular TLDs tend to have larger RTTs (Round Trip Times). In other words, there are no popular TLDs with small RTTs. Overall, 90% of queries in 2014 DITL data were to TLDs having NS TTLs greater than or equal to 1 day.

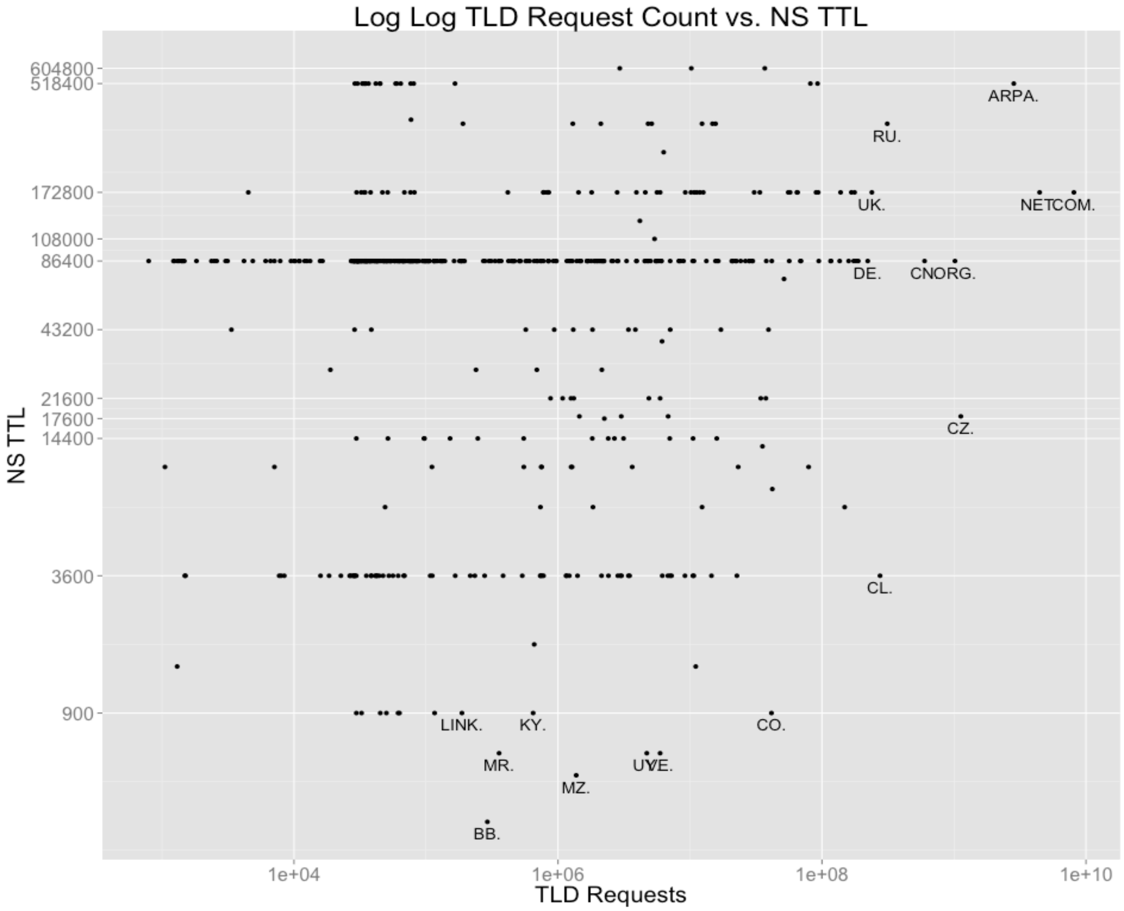

Figure 2: TLD query counts vs NS TTL (2014 DITL data)

6.1.2. DS vs. DNSKEY TTLs

In DNSSEC, the DS record is a cryptographic hash of a DNSKEY record. Unlike NS records, which exist both in the parent and child zones, DS records exists only in the parent zone and DNSKEY records exist only in the child zone.

Two RFCs describing DNSSEC mention TTLs for DS records. However, they are not in agreement. RFC 4034 (section 5) says:

> The DS RR has no special TTL requirements.

RFC 4035 (section 2.4) says:

> The TTL of a DS RRset SHOULD match the TTL of the delegating NS RRset.

In the root zone, delegating NS records have a 2 day TTL. However, the DS records have a 1 day TTL, against the advice of RFC 4035. This is not particularly surprising since a mistake with a DS record can deny resolution for all names under a TLD. Given the way that DS records are currently used in the root zone (e.g., usually matching just one TLD KSK) it is better for them to have a lower TTL in the event of an emergency change.

We also studied the relationship between TTLs of DS and DNSKEY records. While there is no requirement that DS and DNSKEY TTLs match, the choice of DNSKEY TTLs provides some insight in what values the TLD operators consider appropriate. Thus, the study team conducted a similar survey on all TLDs (923 in total) present in the root zone, and analyzed the DNSKEY TTL in each zone. At the time of the survey (May 5, 2015), 744 TLDs have a DS record in the root zone and 179 do not (mostly ccTLDs). The results of DNSKEY TTLs are shown in Figure 3 below.

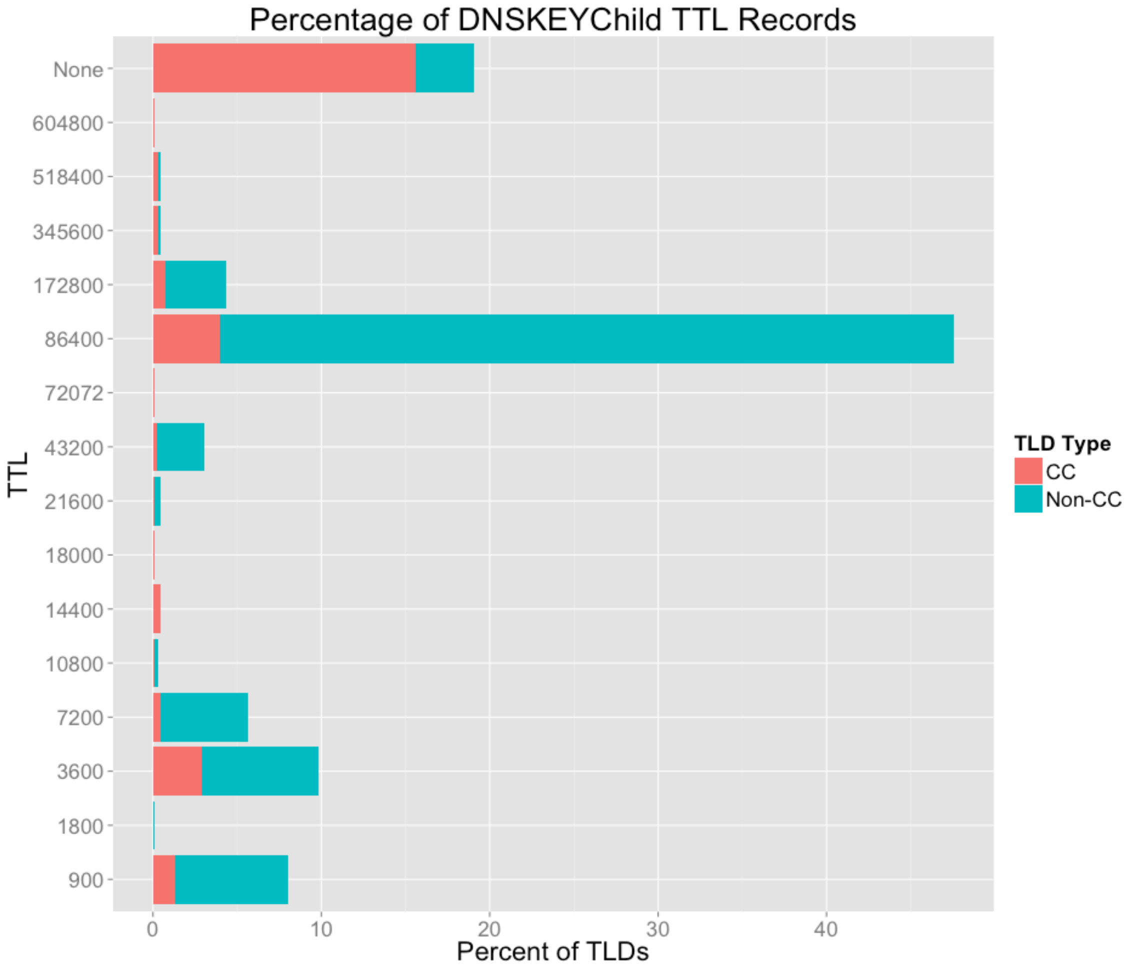

Figure 3: Distribution of TTL values for TLD DNSKEY records. The "None" category shown here on the y-axis represents TLDs that are not DNSSEC-signed.

All DS records in the root zone are given a one day TTL. Figure 3 shows that about 47% of TLDs have DNSKEY TTLs matching the root zone DS TTL. Among those that don't match, most of them are set lower (900, 3600, 7200 seconds) and just a few are set higher

(2 days). Overall, 65% of signed TLDs have DNSKEY TTLs greater than or equal to 1 day.

The key takeaways of these two studies are:

- 80% of TLDs have authoritative NS TTLs greater than or equal to 1 day.
- 90% of queries in 2014 DITL data were to TLDs having NS TTLs greater than or equal to 1 day.
- 65% of signed TLDs have DNSKEY TTLs greater than or equal to 1 day.

This means that most TLDs, and especially those with the most traffic, use authoritative NS TTLs match with those in the root zone. Thus, the root zone NS TTL and DS TTL are still appropriate for today's Internet environment.

6.2. Survey of "max-cache-ttl" parameters in popular recursive resolver implementations

The Domains Operation Guide (RFC 1033) advises using a TTL value between one day to one week. The DNS Standard (STD 13, RFC 1035) recommends that, as an optional step, responses with excessive TTL "greater than 1 week" should be discarded. This led to the practice that (recursive) caching resolvers have a limit on the amount of time a response is cached.

Numerous recursive name server implementations have a max-cache-ttl parameter that sets the maximum caching time (in seconds). Thus, if the recursive server receives a record with a TTL value larger than max-cache-ttl, it will be removed from the cache after max-cache-ttl seconds. Some implementations reduce the stored TTL value so that it gets removed when it reaches zero. Others use the original value and remove it when it reaches "original minus max-cache-ttl."

A negative response is one in which the query resolution either encountered an error, or data of the requested type was not available. The amount of time that a negative response may be cached is determined by values in the SOA record (see section 3.1 "The SOA Record"). Some implementations also have a separate max-ncache-ttl parameter for negative responses.

We surveyed these two parameters for a number of popular products (their latest versions) and report the findings in the table below.

Table 5: Default max-cache-ttl and max-ncache-ttl for popular recursive resolvers

DNS Recursive Resolver Software	max-cache-ttl	max-ncache-ttl
BIND	7 days	3 hours
Djbdns	7 days	N/A
MaraDNS	730 days	N/A
Microsoft DNS	1 day	15 min
Nominum Vantio	7 days	3 hours
PowerDNS (pdns recursor)	1 day	1 hour
Unbound	1 day	N/A

The table below lists maximum caching time of some popular public recursive DNS services. The data was collected by issuing queries for a name that returns a Unix timestamp in a TXT record. The authoritative server returns the TXT record without a 10-day TTL. Queries are repeated (from a single location) every 60 seconds and the results analyzed to see how long each service returns cached records.

The raw data shows that some services return a single (or small number of) cached records for a long time, while others exhibit more complex caching behavior. Since the authoritative server returns unique records, we can easily see how many different records a particular service returns from its cache. There are a number of reasons that a particular service might return different cached records over a short period of time, including: IP anycast routing changes, numerous load-balanced backend systems, cache replacement, and application restarts.

For each service we calculate the amount of time that it returns each unique timestamped record. From these we calculate the average (median) and maximum "time in cache" values.

Table 6: maximum caching time of public recursive DNS services

		Time In Cache	
Name	IP	Average	Maximum
Censurfridns	89.233.43.71	24h	24h
Comodo Secure DNS	8.26.56.26	21h 53m	22h 8m

Dyn	216.146.35.35	24h	24h
Google Public DNS	8.8.8.8	5h 35m	6h 38m
Level 3	4.2.2.2	19m	24h
Norton ConnectSafe	198.153.194.50	24h	24h
OpenDNS	208.67.222.222	5h 40m	4d 4h
UltraDNS/Neustar	156.154.71.1	24h	24h
Verio / NTT	129.250.35.250	3d 20h	6d 16h

The key takeaways from this analysis are:

First, while BIND has a relatively large max-cache-ttl default value (1 week), most of the other popular implementations enforce a limit of 1 day. Second, most of the popular public recursive name servers limit caching to 1 day, with OpenDNS and Verio / NTT as notable exceptions (4 day 4 hours and 6 day 16 hours respectively). The average caching time for the public recursive resolvers is much lower, from 19 minutes to 24 hours.

6.3. DITL Analysis

To understand the extent to which root server clients "honor" root zone TTLs, the team analyzed DNS-OARC's 2014 Day In The Life of the Internet (DITL) data.[4]

Study Data: DITL-2014 includes a 48-hour data collection (pcap files) from a number of root name servers, as well as TLDs and other DNS services. Nine root operators (A, C, E, F, H, I, J, K, M) provided data. Data from I-root was not usable because its source IP addresses are anonymized. Thus data from 8 root servers are used. A data cleansing was performed to filter out IP addresses from obviously spoofed source addresses (e.g., 10/8, 0/8, 127/8), leaving 1.2 Terabyte of compressed pcap files to study.

The study team notes that the DITL data collection period is 48 hours, matching exactly delegation NS TTL. Thus if a client was doing "perfect" caching based on the delegation NS TTL, we would only see one query from it (per TLD) during the entire data period.

The study team did find 20% of the IPs made only one request for delegated TLDs during collection. However, 90% of those IPs issued 10 or fewer queries – which is not a behavior typically exhibited by a recursive name server. A typical recursive server would send a much higher number of queries – especially queries for non-existent domains. Thus, the study team concludes that a majority of these IPs are in fact non-recursive resolvers.

Study Methodology: To simplify the analysis, we first sorted the pcap files by source IP address. We generated a list of delegated TLDs from an archived root zone file. Most of the analysis is performed on queries for delegated TLDs only, as queries for undelegated

[4] https://www.dns-oarc.net/node/341

TLDs (aka “NXDomains”) are subject to negative caching and will not be affected by changes to root zone TTLs.

Since the root name servers are also authoritative for the root-servers.net zone, we treat root-servers.net as a separate pseudo-TLD, rather than have these queries fall under the .net TLD.

For each (client IP, TLD) pair we calculate the time delta between queries from that client IP for names under that TLD. From there we calculate:

- Minimum Time Delta
- Maximum Time Delta
- Mean Time Delta
- Median Time Delta
- Number of Time Delta Measurements
- Number of Requests from IP regardless of qName

This results in 13.47B time delta measurements from 9.85MM unique IPs. We observed that the vast majority of these IPs issue relatively few requests, 50% made less than 10 queries; most likely these IPs are not recursive name servers. Nearly 20% of the IPs made only request for delegated TLDs during collection; however, 90% of those IPs in general issued 10 or fewer queries – again most likely these IPs are not recursive name servers.

51.8% of all queries in the DITL 2014 data are for non-existent TLDs. These are subject to negative caching which often has lower limits (3 hours for BIND and 15 minutes for Windows). Unless root zone TTLs and SOA parameters change significantly, it would not affect the amount of “NXDOMAIN queries” received at the root.[5]

Study Results: Figure 4 below shows the cumulative distribution of the number of client IPs sending queries (red line) and providing measurements (blue line).

[5] Assuming these queries come from well-behaved clients in the first place.

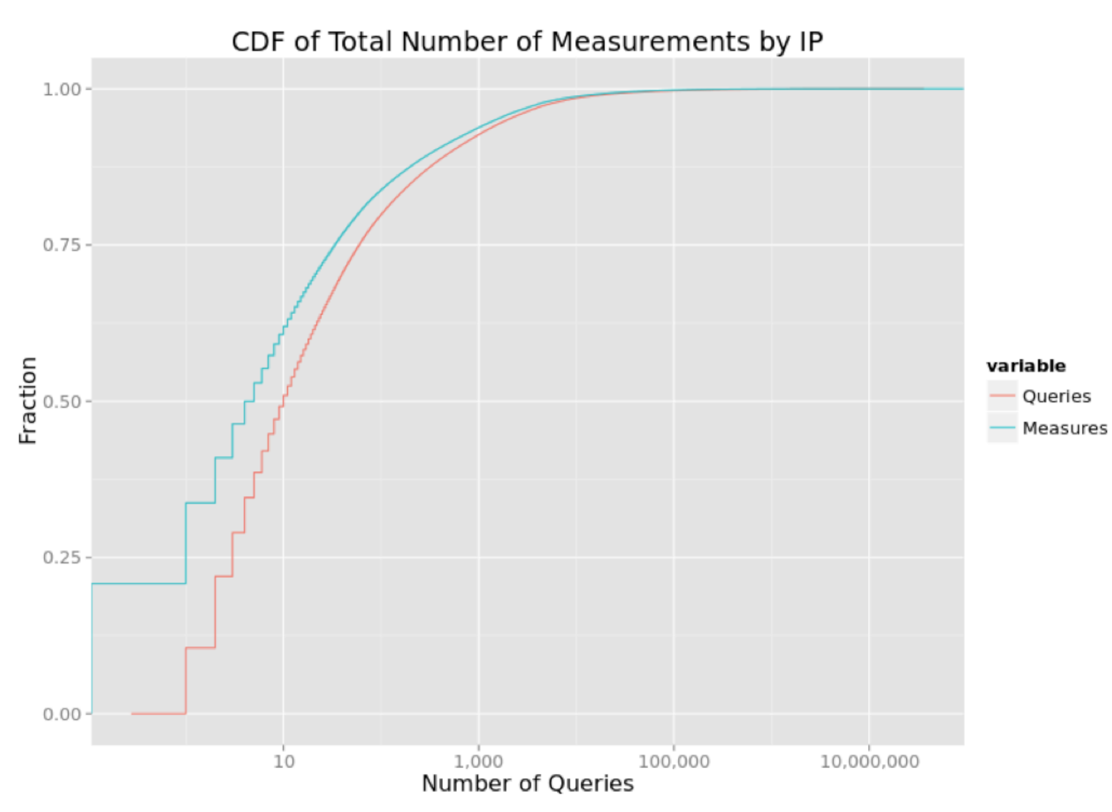

Figure 4: Cumulative Distribution of Total Number of Measurements by IP

The “CDF of Median Time Delta in Seconds Between Queries” graph below (Figure 5) shows the cumulative distribution of the per-client medians for a number of TLDs. In this plot, the busy clients are represented by data points on the left side of the plot, while low-volume clients are toward the right.

In a CDF plot such as this, a vertical feature represents a population of clients at that value. All of the TLDs have a large vertical component at the left side of the graph, between 0 and 3600 seconds. For the root-servers.net pseudo-TLD, we can see that about 85% of clients that sent queries for root-servers.net have a median time delta less than 1 hour. At the other extreme, for clients sending queries ending with .arpa, only about 30% of them have a median time delta less than one hour.

We also observe some vertical features at 43200 (12 hours) and 86400 (1 day). This tells us that some root clients send same-TLD queries at those intervals. It is easy to understand the change at 86400 since this is a common max-cache-ttl limit (see section 6.2) and a common value in TLD authoritative zones (see section 6.1). The change at 43200 is a little more mysterious. Since this value does not appear in any root zone TTLs or max-cache-ttl limits, we can perhaps conclude that it is a common value in TLDs or other deeper zones in the DNS.

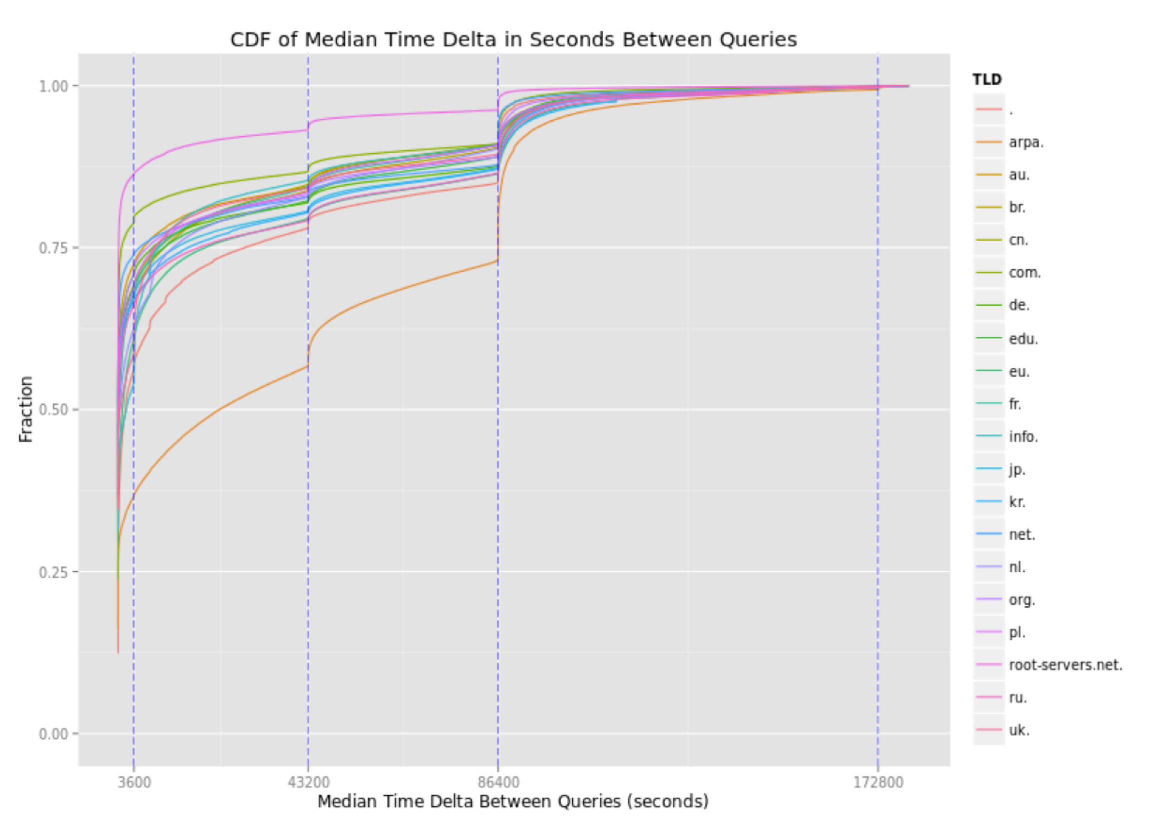

Figure 5: CDF of Median Time Delta in Seconds Between Queries

Table 7: Per-TLD percentiles and Median Time Deltas found in DITL 2014 data. This is same data as Figure 5, in tabular form.

	Median Time Deltas for these Percentiles		Percentiles for these Median Time Deltas			
TLD	50th %	95th%	1 hour	12 hours	1day	2 days
.	1694	87495	0.57	0.78	0.86	1.00
arpa.	23022	100438	0.37	0.57	0.74	0.99
au.	1	90807	0.72	0.84	0.91	1.00
br.	2	90025	0.69	0.84	0.91	1.00
cn.	13	89563	0.68	0.85	0.91	1.00
com.	16	86900	0.79	0.87	0.91	1.00
de.	18	90532	0.71	0.82	0.88	1.00
edu.	2	91772	0.69	0.82	0.89	1.00
eu.	994	94171	0.61	0.80	0.87	1.00
fr.	113	90195	0.68	0.84	0.91	1.00
info.	1822	89930	0.54	0.85	0.91	1.00
jp.	4	92523	0.68	0.81	0.88	1.00
kr.	0	94882	0.67	0.81	0.87	1.00

net.	2	86671	0.74	0.83	0.88	1.00
nl.	331	91733	0.63	0.83	0.90	1.00
org.	21	88402	0.71	0.83	0.89	1.00
pl.	4	89682	0.69	0.84	0.91	1.00
root-servers.net.	3	49483	0.86	0.93	0.96	1.00
ru.	60	93405	0.66	0.79	0.87	1.00
uk.	491	91224	0.59	0.84	0.90	1.00

Note that the “CDF of Median Time Delta” figure represents clients, rather than queries. Also note that it only includes queries for valid TLDs, while a significant percentage of queries (more than 50%) result in NXDOMAIN responses. To answer a question such as “what % of queries might be affected by a TTL change?” we need to weight the clients by the amount of queries that they send.

Figure 6 and Table 8 below shows how many queries come from clients with per-TLD median time between queries falling within certain ranges. For example, when we look at clients sending same-TLD queries to the roots every 0-60 seconds, we find those account for 5.8% of all queries. The largest category is the 18.1% of queries from clients sending same-TLD queries every 3600-10800 seconds, or 1-3 hours. The amount of queries in the 1 day and up category is essentially zero. This provides further evidence that root zone TTLs matter very little when it comes to actual caching behavior.

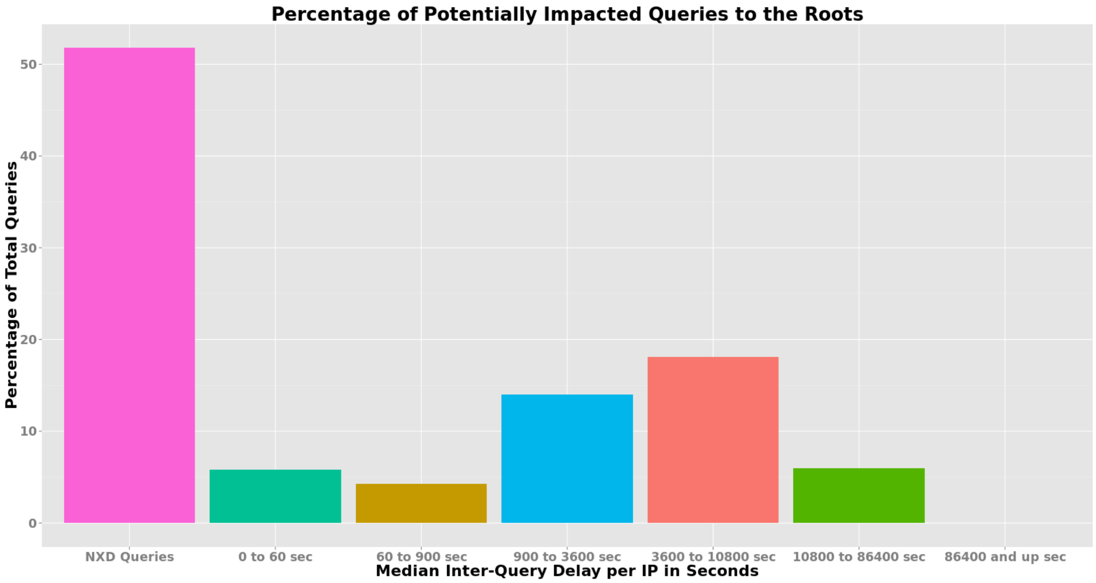

Figure 6: Percentage of Potentially Impacted Queries to the Roots

Table 8: Percentage of Potentially Impacted Queries to the Roots

Median Inter-Query Delay per IP	Percentage of Total Queries
NXD	51.8%

0 to 60 seconds	5.8%
60 to 9000 seconds	4.2%
900 to 3600 seconds	14.0%
3600 to 10800 seconds	18.1%
10800 to 86400 seconds	6.0%
86400 seconds or more	0.0001%

Based on this analysis, the study team finds:

- Query volume for the delegated TLDs follows a long tail distribution in which the top 20 TLDs account for 87.6% of the queries.
- CDF plots for the top 20 TLDs clearly show the vast majority of IPs disregard TTLs and issue queries for same-TLDs within short periods of time.
- Subpopulations of IPs appear to issue same-TLD queries for TLDs at specific intervals – 43,200 and 86,400 seconds (12 hours and 24 hours).
- 51.8% of all queries in the DITL 2014 data are for non-existent TLDs. These are subject to negative caching which often has lower limits (3 hours for BIND and 15 minutes for Windows). Unless root zone TTLs and SOA parameters change significantly, it would not affect the amount of "NXDOMAIN queries" received at the root.[6]
- 99.8% of queries for existent TLDs come from clients with per-TLD median time deltas between 0 and 86,400 seconds (24 hours), providing further evidence that root zone TTLs matter very little when it comes to actual caching behavior.

6.4. Interactions between SOA refresh and DNSSEC Signature Validity

The study team identified two potential problems that relate to the interaction between the SOA refresh value and the root zone's signature validity periods.

As mentioned earlier, the root zone SOA record has an Expire value of seven days. This means that, should a root server instance become "disconnected" and fail to receive updates from a master server, it would continue serving its stored copy of the root zone for seven days after the previous update. After that, it will return SERVFAIL in response to root zone queries.

While SERVFAIL responses are, in a sense, a bad thing, they are preferable to returning stale data, especially when it comes to DNSSEC. Upon receiving a SERVFAIL response,

[6] Assuming these queries come from well-behaved clients in the first place.

a recursive name server should retry its query at another authoritative server. A problem arises, however, when a root server returns stale data with expired signatures to a non-validating recursive name server that has validating clients. The non-validating recursive name server will cache the stale data based on the TTL. Its validating clients will continue receiving the stale data until the TTL expires.

The question we must answer is: are there any situations whereby a root name server would return data and signatures that will be cached beyond the signature validity end time?

6.4.1. Validity Period for ZSK Signatures

The ZSK is used to generate signatures for all root zone RRsets, except the DNSKEY RRset.The root zone signed and published at least twice per day (every 12 hours) and signatures from this key are given a validity period of 10 days.

Even though the signature validity period was recently increased to 10 days, a potential problem remains with root zone apex NS records served by a "disconnected" server and cached by a non-validator.

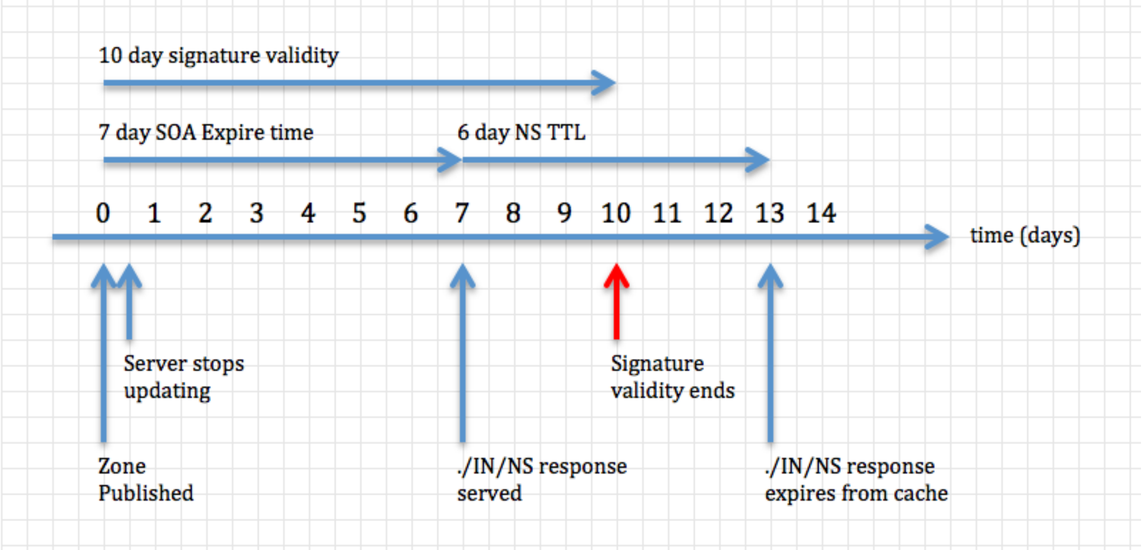

Figure 7: Timeline of a worst-case scenario demonstrating how a non-validating recursive name server might cache a signed NS RRset past its signature validity period.

As the above figure shows, a root name server which remains "disconnected" for more than 4 days will serve data that a non-validator might cache for a time exceeding the signature validity.

At day 0, the root server instance receives the most recent version of the zone.Shortly after, a problem occurs which prevents the instance from receiving zone updates, but allows the server to continue responding to queries.On or just before day 7, the server returns a ./IN/NS response to a non-validating client, which then caches it.On day 10, the end of the signature validity period is reached, yet the data remains cached because it was not validated.The cache may continue to return the stale response to its clients for another 3 days, up to day 13.

This problem could be alleviated by:

- Reducing the NS RRset TTL to a value less than or equal to 3 days, or
- Increasing the ZSK-generated signature validity period to a value greater than or equal to 13 days (only necessary for NS RRset signatures), or
- Reducing the SOA Expire time to a value less than or equal to 4 days

6.4.2. Validity Period for KSK Signatures

The root zone KSK signs only the DNSKEY RRset, which has a two day TTL.A potential problem, similar that for ZSK signatures, exists for the DNSKEY RRset.

Whereas signatures by the ZSK are generated each time the zone is published (i.e., at least twice per day), signatures from the KSK are generated well in advance, at the quarterly Root KSK Ceremonies.[7] At such a ceremony, the KSK signs the upcoming DNSKEY RRset nine times,[8] resulting in nine signatures, each covering a different (but overlapping) period of time. A new RRSIG record covering the DNSKEY RRset appears in the root zone every ten days.

As the below figure shows, a root name server which remains "disconnected" for more than 3 days will serve data that a non-validator might cache for a time exceeding the signature validity. In the worst-case scenario, end user clients may receive the bad data for up to 4 days.

At day 0, a new RRSIG covering the DNSKEY RRset is published and distributed to root name servers.Just before day 10, a root server instance experiences a problem which prevents the it from receiving zone updates, but allows the server to continue responding to queries. It will continue responding successfully to queries for the next 7 days.On day 15 the DNSKEY RRISG reaches the end of its validity period.The instance will continue sending this now-expired signature to clients for another 2 days.Just before day 17, a non-validating client receives the stale data and may cache it for another 2 days, up to day 19.

Note that this problem can only occur if the instance fails to update between days 5-10 after a new RRSIG is published.

[7] https://www.iana.org/dnssec/ceremonies

[8] https://www.verisigninc.com/assets/dps-zsk-operator-1527.pdf

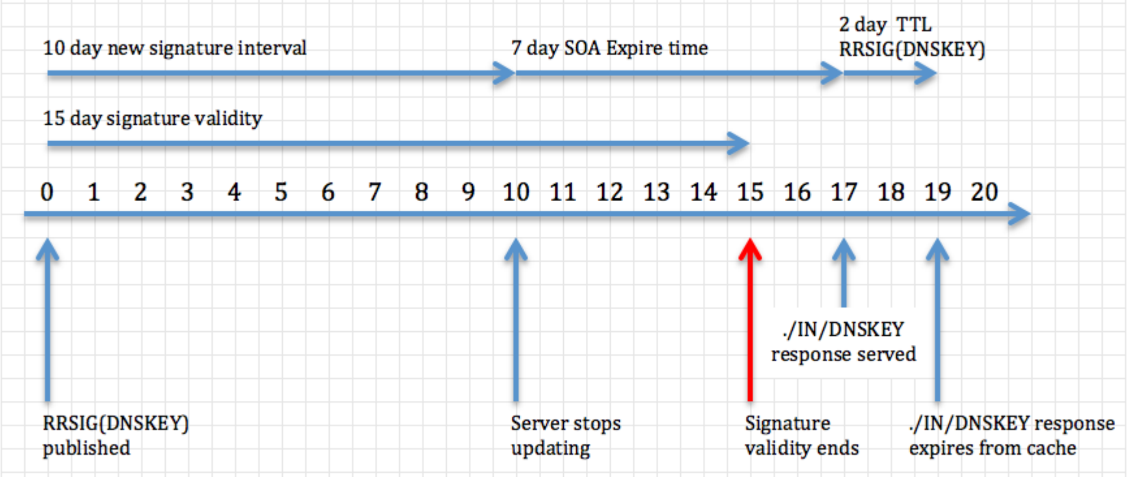

Figure 8: Timeline of a worst-case scenario demonstrating how a non-validating recursive name server might cache a signed DNSKEY RRset past its signature validity period.

This problem could be alleviated by:

- Increasing the KSK-generated signature validity period to a value greater than or equal to 19 days, or
- Reducing the SOA Expire time to a value less than or equal to 3 days, or
- Decreasing the interval for new KSK-generated signatures, and therefore the number of such signatures generated during a ceremony, to a value less than or equal to 6 days

6.4.3. Analysis of Proposed Remedies

In this section, the team conducts a preliminary analysis of the pros and cons of various remedies as proposed in Section 6.4.1 and 6.4.2.

Table 9: Pros and Cons of Proposed Remedies

	Proposed Remedies	Pros	Cons
Validity Period for ZSK Signature	Reducing the NS RRset TTL to<= 3 days	Unlikely to impact root server query rates.	First change to NS TTLs in 25 years.
	Increasing the ZSK-generated signature validity period to >=13 days (only necessary for NS RRset signatures),	Does not affect TTLs.	Longer validity perceived as weaker and enables longer replay attacks.

	Reducing the SOA Expire time to <= 4 days	Reducing SOA Expire also solves KSK validity problem.	Reduces amount of time for disconnected operation.
Validity Period for KSK Signature	Increasing the KSK-generated signature validity period >=19 days	Does not affect TTLs	Longer validity perceived as weaker and enables longer replay attacks.
	Reducing the SOA Expire time to <= 3 days	Reducing SOA Expire also solves KSK validity problem.	Reduces amount of time for disconnected operation.
	Decreasing the interval for new KSK-generated signatures, and therefore the number of such signatures generated during a ceremony, to a value less than or equal to 6 days	Does not affect TTLs	Slight increase in complexity to ZSK publication.

7. Findings

Finding 1: The root zone delegation TTLs are appropriate for today's environment.

In section 6.1 ("Survey of TTLs of TLDs") we find that most TLDs, and especially those with the most traffic, use authoritative NS TTLs matching those in the root zone. That is, 1 day and 2 days are the most common choices for NS TTLs in authoritative TLD zones.

Finding 2: Root zone TTLs values could be reduced to 1 day without any significant impact on the amount of traffic to root servers.

In section 6.2, we find that except for BIND, 1 day is a very common "max-cache-ttl" parameter.

In section 6.3, we find that 51.8% of all queries in the DITL 2014 data are for non-existent TLDs. These are subject to negative caching which often has lower limits (3 hours for BIND and 15 minutes for Windows). Of the remaining queries for existent TLDs, 99.8% of them come from clients with per-TLD median time between queries fall between 0 – 86,400 seconds (24 hours).

This likely means that root zone TTLs could be reduced to 1 day without any significant impact on traffic levels to root name servers.

Finding 3: Increasing root zone TTLs should only be done with careful consideration of DNSSEC-related implications.

While the same analysis leads us to conclude that any *increase* in root zone TTLs would not have a significant impact on root server traffic levels, we caution against increasing TTLs because of potential DNSSEC-related problems identified in section 6.4.

Finding 4: Root zone TTLs appear to not matter to most clients.

In section 6.3 ("DITL Analysis"), the results indicate that time intervals between queries under the same TLD are highly skewed toward small values.Most root server clients appear to send same-TLD queries at rates far higher than would be predicted by strict caching based on root zone TTLs.In other words, root zone TTLs appear not to matter to most clients. Of the top 20 TLDs, more than 50% of clients send same-TLD queries more than once per hour.

Finding 5: Few reasons exist today to consider changes to root zone TTLs.

Although the study finds that root zone TTLs could be reduced to 1 day without any significant impact on traffic levels to root name servers, and root zone TTLs also appear to be ignored by most root server clients, we find no compelling reasons to change the TTLs at this time. This is because:

- As a general principle of conservatism, there is little need to have root zone changes take effect more quickly than it is today. Although reducing root zone TTLs will have root zone changes take effect more quickly than it happens today, unlike many DNS zones, the root zone rarely has a need for the quick propagation of changes. Rather, changes to the root zone are made slowly, and deliberately. Delegations (TLDs) are added well in advance of queries from end users. Root name servers themselves are renumbered infrequently and with great care and planning.

- Although increasing root zone TTLs could lead data remaining in resolver caches longer and increasing the amount of time a recursive name server could function without communicating with a root name server, caution should be given against increasing it from what it is today because of potential DNSSEC-related problems identified in section 6.4.

- Where we do find legitimate, technical reasons to consider a TTL change, other solutions are also available.

Finding 6: Two potential problems related to the interaction between the SOA Expire value and the root zone's signature validity periods exist, and need to be addressed by the Internet Community.

In Section 6.4, the study team identifies two situations whereby a root name server would return data and signatures that will be cached beyond the signature validity end time. Specifically, in certain situations:

1. a root name server which remains "disconnected" for more than 4 days could serve data that a non-validator might cache it for a time exceeding the ZSK signature validity period.

2. a root name server which remains "disconnected" for more than 3 days could serve data that a non-validator might cache it for a time exceeding the KSK signature validity period. In the worst-case scenario, end user clients may receive the bad data for up to 4 days.

8. Recommendation

Based on the findings of this report, the RSSAC Caucus recommends that:

Recommendation 1: To address the DNSSEC problems identified in Section 6.4, the RSSAC should choose one or a combination of mitigation options (*e.g.* reducing the SOA Expire Time, reducing NS RRSet TTL, increase the KSK-generated Signature Validity Time, etc.), after carefully considering their operational implications.

In Section 6.4.3, the RSSAC caucus performed an initial analysis the tradeoffs of these options. Broader Community input maybe needed. These consultations could include, but not limited to, the DNS Operations Community, ICANN Security and Stability Advisory Committee, and the Internet Architecture Board.

One key question in this analysis is how long should a root name server continue serving data if it fails to update the zone.

Recommendation 2: No other changes to Root Zone TTLs should be made at this time.

9. Future Work

To better understand and/or corroborate the findings of this report, the study team proposes one future work item (outside RSSAC) for the DNS researchers to consider:

Trace-driven simulations of changing root zone TTLs

RSSAC Report on Root Zone TTLs

To better understand the traffic effects of changing root zone TTLs, the following trace driven simulations can be performed:

- collect inbound query data from a recursive server
- collect copies of a TLD zone
- replay queries through an empty cache towards a server serving the TLD zone in order to characterize the cache miss rate
- modify the delegation NS RRset TTLs in the TLD zone and repeat
- compare traffic impact of different TTLs in the TLD zone
- presumably sink traffic that would follow referrals from the "TLD server" somewhere local and benign in order to avoid flooding the internet

A prerequisite for any of the above is a clear description of the experiment to the recursive resolver operator and the TLD operator to make sure they approve of the experiment, and that there is no loss of private data, etc.

10. 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 Acknowledgments 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.

10.1. 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
Joe Abley
Jaap Akkerhuis
John Bond
Brian Dickson
Shumon Huque
Warren Kumari
Duane Wessels (work party leader)

Invited Expert
Matthew Thomas

ICANN Support staff
Kathy Schnitt
Steve Sheng (editor)
Barbara Roseman

10.2. 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.

10.3. Dissents

There were no dissents.

10.4. Withdrawals

There were no withdrawals.