Internet Engineering Task Force (IETF)
Internet-Draft
Intended status: Informational
Expires: May 3, 2018

L. Song, Ed.
D. Liu, Ed.
Beijing Internet Institute
P. Vixie, Ed.
TISF
Kato, Ed.
Keio University/WIDE Project
S. Kerr

October 30, 2017

Yeti DNS Testbed
draft-song-yeti-testbed-experience-05

Abstract

The Internet's Domain Name System (DNS) is built upon the foundation provided by the Root Server System -- that is, the critical infrastructure that serves the DNS root zone.

Yeti DNS is an experimental, non-production testbed that aims to provide an environment where technical and operational experiments can safely be performed without risk to production infrastructure. This testbed has been used by a broad community of participants to perform experiments that aim to inform operations and future development of the production DNS. Yeti DNS is an independently-coordinated project and is not affiliated with ICANN, IANA or any Root Server Operator.

The Yeti DNS testbed implementation includes various novel and experimental components including IPv6-only transport, independent, autonomous Zone Signing Key management, large cryptographic keys and a large number of component Yeti-Root Servers. These differences from the Root Server System have operational consequences such as large responses to priming queries and the coordination of a large pool of independent operators; by deploying such a system globally but outside the production DNS system, the Yeti DNS project provides an opportunity to gain insight into those consequences without threatening the stability of the DNS.

This document neither addresses the relevant policies under which the Root Server System is operated nor makes any proposal for changing any aspect of its implementation or operation. This document aims solely to document technical and operational findings following the deployment of a system which is similar but different from the Root Server System.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 3
2. Areas of Study 4
2.1. Implementation of a Root Server System-like Testbed . . . 5
2.2. Yeti-Root Zone Distribution 5
2.3. Yeti-Root Server Names and Addressing 5
2.4. IPv6-Only Yeti-Root Servers 5
2.5. DNSSEC in the Yeti-Root Zone 5
3. Yeti DNS Testbed Infrastructure 6
3.1. Root Zone Retrieval 7
3.2. Transformation of Root Zone to Yeti-Root Zone 8
3.2.1. ZSK and KSK Key Sets Shared Between DMs 8
3.2.2. Unique ZSK per DM; No Shared KSK 9
3.2.3. Preserving Root Zone NSEC Chain and ZSK RRSIGs . . . 11
3.3. Yeti-Root Zone Distribution 11

3.4. Synchronization of Service Metadata 11
3.5. Yeti-Root Server Naming Scheme 12
3.6. Yeti-Root Servers 13
3.7. Traffic Capture and Analysis 15
4. Operational Experience with Yeti DNS Testbed 15
4.1. Automated Hints File Maintenance 15
4.2. IPv6-only root operation 16
4.2.1. Impact of IPv6 fragmentation 16
4.2.2. How IPv6-only Root serve IPv4 users? 18
4.3. Experience on Multiple Signers 19
4.3.1. IXFR fallback to AXFR 19
4.3.2. Latency of Root Zone update 20
4.4. Root Label Compression in Knot 20
4.5. Increased ZSK Key Size 21
4.6. KSK Rollover 21
5. IANA Considerations 21
6. Acknowledgments 21
7. References 22
Appendix A. Yeti-Root Hints File 25
Appendix B. Controversy 26
Appendix C. About This Document 27
C.1. Venue 27
C.2. Revision History 28
C.2.1. draft-song-yeti-testbed-experience-00 through -03 . . 28
C.2.2. draft-song-yeti-testbed-experience-04 28
Authors' Addresses 28

1. Introduction

The Domain Name System (DNS), as originally specified in [RFC1034] and [RFC1035], has proved to be an enduring and important platform upon which almost every user of Internet services relies. Despite its longevity, extensions to the protocol, new implementations and refinements to DNS operations continue to emerge both inside and outside the IETF.

The Root Server System in particular has seen technical innovation and development in recent years, for example in the form of wide-scale anycast deployment, the mitigation of unwanted traffic on a global scale, the widespread deployment of Response Rate Limiting [RRL], the introduction of IPv6 transport, the deployment of DNSSEC, changes in DNSSEC key sizes and preparations to roll the root zone's trust anchor. Together, even the projects listed in this brief summary imply tremendous operational change, all the more impressive when considered the necessary caution when managing Internet critical infrastructure, and the context of the adjacent administrative changes involved in root zone management and the (relatively

speaking) massive increase in the the number of delegations in the root zone itself.

Aspects of the operational structure of the Root Server System have been described in such documents as [TNO2009], [ISC-TN-2003-1], [RSSAC001] and [RFC7720]. Such references, considered together, provide sufficient insight into the operations of the system as a whole that it is straightforward to imagine structural changes to the root server system's infrastructure, and to wonder what the operational implications of such changes might be.

The Yeti DNS Project was conceived in May 2015 to provide a captive, non-production testbed upon which the technical community could propose and run experiments designed to answer these kinds of questions. Coordination for the project was provided by TISF, the WIDE Project and the Beijing Internet Institute. Many volunteers collaborated to build a distributed testbed that at the time of writing includes 25 Yeti root servers with 16 operators and handles experimental traffic from individual volunteers, universities, DNS vendors and distributed measurement networks.

By design, the Yeti testbed system serves the root zone published by the IANA with only those structural modifications necessary to ensure that it is able to function usefully in the Yeti testbed system instead of the production Root Server system. In particular, no delegation for any top-level zone is changed, added or removed from the IANA-published root zone to construct the root zone served by the Yeti testbed system. In this document, for clarity, we refer to the zone derived from the IANA-published root zone as the Yeti-Root zone.

The Yeti DNS testbed serves a similar function to the Root Server System in the sense that they both serve similar zones (the Yeti-Root zone and the Root zone, respectively). However, the Yeti DNS testbed only serves clients that are explicitly configured to participate in the experiment, whereas the Root Server System serves the whole Internet. The known set of clients has allowed structural changes to be deployed in the Yeti DNS testbed whose impact on clients can be measured and analysed.

2. Areas of Study

Examples of topics that the Yeti DNS Testbed was built to address are included below, each illustrated with indicative questions.

2.1. Implementation of a Root Server System-like Testbed

- How can a captive testbed be constructed and deployed on the Internet, allowing useful public participation without any risk of disruption of the Root Server System?

- How can representative traffic be introduced into such a captive testbed such that insights into the impact of specific differences between the testbed and the Root Server System can be observed?

2.2. Yeti-Root Zone Distribution

- What are the scaling properties of Yeti-Root zone distribution as the number of Yeti-Root servers, Yeti-Root server instances or intermediate distribution points increase?

2.3. Yeti-Root Server Names and Addressing

- What naming schemes other than those closely analogous to the use of ROOT-SERVERS.NET in the production root zone are practical, and what are their respective advantages and disadvantages?

- What are the risks and benefits of signing the zone that contains the names of the Yeti-Root servers?

- What automatic mechanisms might be useful to improve the rate at which clients of Yeti-Root servers are able to react to a Yeti-Root server renumbering event?

2.4. IPv6-Only Yeti-Root Servers

- Are there negative operational effects in the use of IPv6-only Yeti-Root servers, compared to the use of servers that are dual-stack?

- What effect does the IPv6 fragmentation model have on the operation of Yeti-Root servers, compared with that of IPv4?

2.5. DNSSEC in the Yeti-Root Zone

- Is it practical to sign the Yeti-Root zone using multiple, independently-operated DNSSEC signers and multiple corresponding ZSKs?

- To what extent is [RFC5011] supported by resolvers?

- Does the KSK Rollover plan designed and in the process of being implemented by ICANN work as expected on the Yeti testbed?

- What is the operational impact of using much larger RSA key sizes in the ZSKs used in the Yeti-Root?

- What are the operational consequences of choosing DNSSEC algorithms other than RSA to sign the Yeti-Root zone?

3. Yeti DNS Testbed Infrastructure

The purpose of the testbed is to allow DNS queries from stub resolvers, mediated by recursive resolvers, to be delivered to Yeti-Root servers, and for corresponding responses generated on the Yeti-Root servers to be returned, as illustrated in Figure 1.

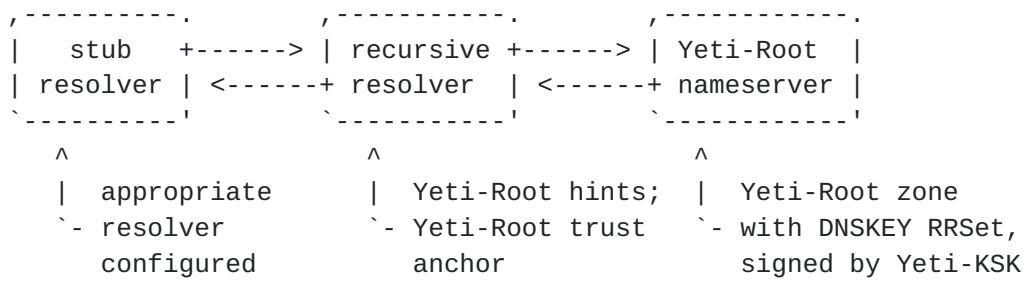

Figure 1: High-Level Testbed Components

To use the Yeti DNS testbed, a stub resolver must be explicitly configured to use recursive resolvers that have themselves been configured to use the Yeti-Root servers. On the resolvers, that configuration consists of a list of names and addresses for the Yeti-Root servers (often referred to as a "hints file") that replaces the normal Internet DNS hints. Resolvers also need to be configured with a DNSSEC trust anchor that corresponds to a KSK used in the Yeti DNS Project, in place of the normal trust anchor for the root zone.

The need for a Yeti-specific trust anchor in the resolver stems from the need to make minimal changes to the root zone, as retrieved from the IANA, to transform it into the Yeti-Root that can be used in the testbed. Those changes would be properly rejected by any validator using an accurate root zone trust anchor as bogus. Corresponding changes are required in the Yeti-Root hints file Appendix A.

The data flow from IANA to stub resolvers through the Yeti testbed is illustrated in Figure 2 and are described in more detail in the sections that follow.

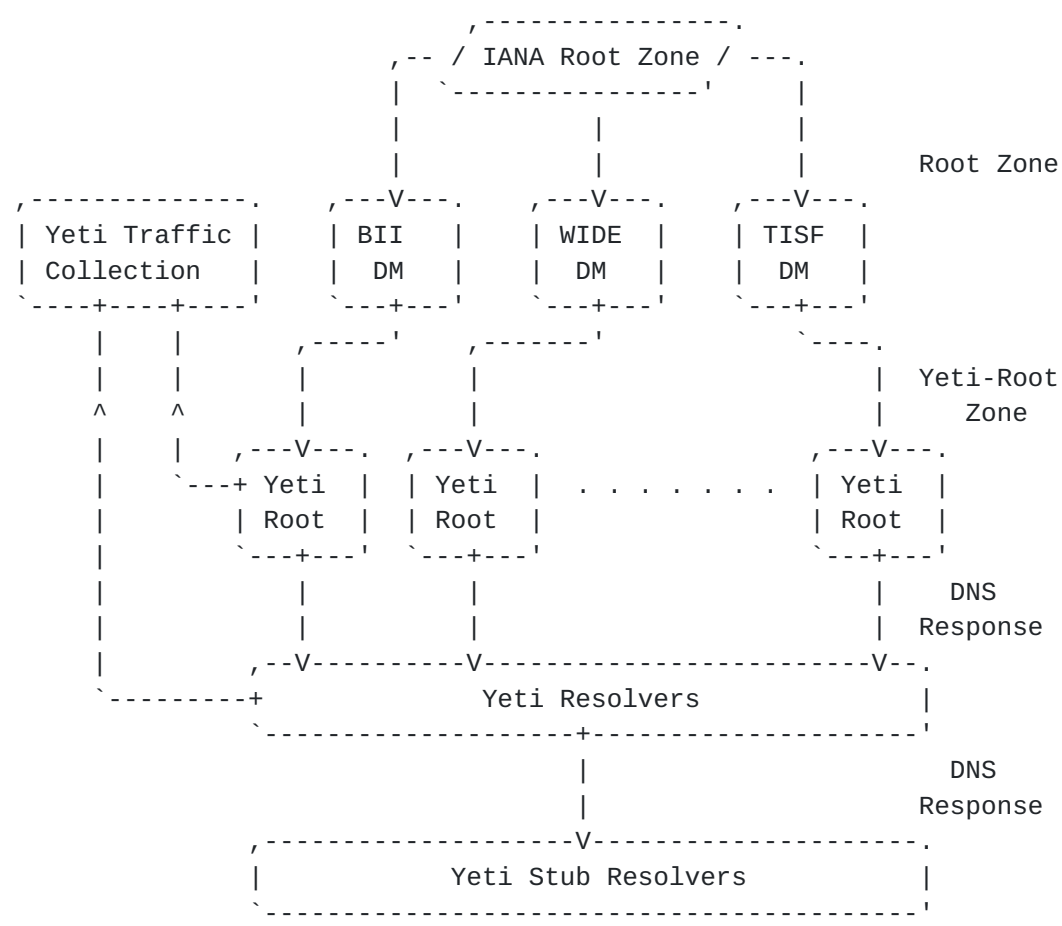

Figure 2: Testbed Data Flow

3.1. Root Zone Retrieval

Since Yeti DNS servers cannot receive DNS NOTIFY [RFC1996] messages from the Root Server System, a polling approach is used. Each Yeti Distribution Master (DM) requests the root zone SOA record from a nameserver that permits unauthenticated zone transfers of the root zone, and performs a zone transfer from that server if the retrieved value of SOA.SERIAL is greater than that of the last retrieved zone.

At the time of writing, unauthenticated zone transfers of the root zone are available directly from B-Root, C-Root, F-Root, G-Root and K-Root, and from L-Root via the two servers XFR.CJR.DNS.ICANN.ORG and XFR.LAX.DNS.ICANN.ORG, as well as via FTP from sites maintained by the Root Zone Maintainer and the IANA Functions Operator. The Yeti DNS Testbed retrieves the root zone from using zone transfers from F-Root. The schedule on which F-Root is polled by each Yeti DM is as follows:

DM Operator	Time
BII	UTC hour + 00 minutes
WIDE	UTC hour + 20 minutes
TISF	UTC hour + 40 minutes

The Yeti DNS testbed uses multiple DMs, each of which acts autonomously and equivalently to its siblings. Any single DM can act to distribute new revisions of the Yeti-Root zone, and is also responsible for signing the RRSets that are changed as part of the transformation of the Root Zone into the Yeti-Root zone described in Section 3.2. This shared control over the processing and distribution of the Yeti-Root zone approximates some of the ideas around shared zone control explored in [ITI2014].

3.2. Transformation of Root Zone to Yeti-Root Zone

Two distinct approaches have been deployed in the Yeti-DNS Testbed, separately, to transform the Root Zone into the Yeti-Root Zone. At a high level both approaches are equivalent in the sense that they replace a minimal set of information in the Root Zone with corresponding data corresponding to the Yeti DNS Testbed; the mechanisms by which the transforms are executed are different, however. Each is discussed in turn in Section 3.2.1 and Section 3.2.2, respectively.

A third approach has also been proposed, but not yet implemented. The motivations and changes implied by that approach are also described in Section 3.2.3.

3.2.1. ZSK and KSK Key Sets Shared Between DMs

The approach described here was the first to be implemented. It features entirely autonomous operation of each DM, but also requires secret key material (the private parts of all Yeti-Root KSK and ZSK key-pairs) to be distributed and maintained on each DM in a coordinated way.

The Root Zone is transformed as follows to produce the Yeti-Root Zone. This transformation is carried out autonomously on each Yeti DNS Project DM. Each DM carries an authentic copy of the current set of Yeti KSK and ZSK key pairs, synchronised between all DMs (see Section 3.4).

1. SOA.MNAME is set to www.yeti-dns.org.

2. SOA.RNAME is set to <dm-operator>.yeti-dns.org. where <dm-operator> is currently one of "wide", "bii" or "tisf".

3. All DNSKEY, RRSIG and NSEC records are removed.

4. The apex NS RRSet is removed, with the corresponding root server glue RRSets.

5. A Yeti DNSKEY RRSet is added to the apex, comprising the public parts of all Yeti KSK and ZSKs.

6. A Yeti NS RRSet is added to the apex that includes all Yeti-Root servers.

7. Glue records (AAAA, since Yeti-Root servers are v6-only) for all Yeti-Root servers are added.

8. The Yeti-Root Zone is signed: the NSEC chain is regenerated; the Yeti KSK is used to sign the DNSKEY RRSet, and the DM's local ZSK to generate every other RRSet.

Note that the SOA.SERIAL value published in the Yeti-Root Zone is identical to that found in the Root Zone.

3.2.2. Unique ZSK per DM; No Shared KSK

The approach described here was the second to be implemented. Each DM is provisioned with its own, dedicated ZSK key pairs that are not shared with other DMs. A Yeti-Root DNSKEY RRSet is constructed and signed upstream of all DMs as the union of the set of active KSKs and the set of active ZSKs for every individual DM. Each DM now only requires the secret part of its own dedicated ZSK key pairs to be available locally, and no other secret key material is shared. The high-level approach is illustrated in Figure 3.

Figure 3: Unique ZSK per DM

The process of retrieving the Root Zone from the Root Server System and replacing and signing the apex DNSKEY RRSet no longer takes place on the DMs, and instead takes place on a central Hidden Master. The production of signed DNSKEY RRSets is analogous to the use of Signed Key Responses (SKR) produced during ICANN KSK key ceremonies.

Each DM now retrieves source data (with pre-modified and Yeti-signed DNSKEY RRset, but otherwise unchanged) from the Yeti DNS Hidden Master instead of from the Root Server System.

Each DM carries out a similar transformation to that described in Section 3.2.1, except that DMs no longer need to modify or sign the DNSKEY RRSet.

The Yeti-Root Zone served by any particular Yeti-Root Server will include signatures generated using the ZSK from the DM that served the Yeti-Root Zone to that Yeti-Root Server. Signatures cached at resolvers might be retrieved from any Yeti-Root Server, and hence are expected to be a mixture of signatures generated by different ZSKs. Since all ZSKs can be trusted through the signature by the Yeti KSK over the DNSKEY RRSet, which includes all ZSKs, the mixture of signatures was predicted not to be a threat to reliable validation. Deployment and experimentation confirms this to be the case, even when individual ZSKs are rolled on different schedules.

A consequence of this approach is that the apex DNSKEY RRSet in the Yeti-Root zone is much larger than the corresponding DNSKEY RRSet in the Root Zone.

3.2.3. Preserving Root Zone NSEC Chain and ZSK RRSIGs

A change to the transformation described in Section 3.2.2 has been proposed that would preserve the NSEC chain from the Root Zone and all RRSIG RRs generated using the Root Zone's ZSKs. The DNSKEY RRSet would continue to be modified to replace the Root Zone KSKs, and the Yeti KSK would be used to generate replacement signatures over the apex DNSKEY and NS RRSets. Source data would continue to flow from the Root Server System through the Hidden Master to the set of DMs, but no DNSSEC operations would be required on the DMs and the source NSEC and most RRSIGs would remain intact.

This approach has been suggested in order to provide cryptographically-verifiable confidence that no owner name in the root zone had been changed in the process of producing the Yeti-Root zone from the Root Zone, addressing one of the concerns described in Appendix B in a way that can be verified automatically.

3.3. Yeti-Root Zone Distribution

Each Yeti DM is configured with a full list of Yeti-Root Server addresses to send NOTIFY messages to, and to form the basis for an address-based access-control list for zone transfers. Authentication by address could be replaced with more rigourous mechanisms (e.g. using Transaction Signatures (TSIG) [RFC2845]); this has not been done at the time of writing since the use of address-based controls avoid the need for the distribution of shared secrets amongst the Yeti-Root Server Operators.

Individual Yeti-Root Servers are configured with a full set of Yeti DM addresses to which SOA and AXFR requests may be sent in the conventional manner.

3.4. Synchronization of Service Metadata

Changes in the Yeti-DNS Testbed infrastructure such as the addition or removal of Yeti-Root servers, renumbering Yeti-Root Servers or DNSSEC key rollovers require coordinated changes to take place on all DMs. The Yeti-DNS Testbed is subject to more frequent changes than are observed in the Root Server System and includes substantially more Yeti-Root Servers than there are Root Servers, and hence a manual change process in the Yeti Testbed would be more likely to suffer from human error. An automated process was consequently implemented.

A repository of all service metadata involved in the operation of each DM was implemented as a separate git repository hosted at github.com, since this provided a simple, transparent and familiar

mechanism for participants to review. Requests to change the service metadata for a DM are submitted as pull requests from a fork of the corresponding repository; each DM operator reviews pull requests and merges them to indicate approval. Once merged, changes are pulled automatically to individual DMs and promoted to production.

3.5. Yeti-Root Server Naming Scheme

The current naming scheme for Root Servers was normalized to use single-character host names (A through M) under the domain ROOT-SERVERS.NET, as described in [RSSAC023]). The principal benefit of this naming scheme is that DNS label compression can be used to produce a priming response that will fit within 512 bytes, the maximum DNS message size using UDP transport without EDNS0 [RFC6891].

Yeti-Root Servers do not use this optimisation, but rather use free-form nameserver names chosen by their respective operators -- in other words, no attempt is made to minimise the size of the priming response through the use of label compression. This approach aims to challenge the need for a minimally-sized priming response in a modern DNS ecosystem where EDNS(0) is prevalent.

Priming responses from Yeti-Root Servers do not always include server addresses in the additional section, as is the case with priming responses from Root Servers. In particular, Yeti-Root Servers running BIND9 return an empty additional section, forcing resolvers to complete the priming process with a set of conventional recursive lookups in order to resolve addresses for each Yeti-Root server. Yeti-Root Servers running NSD appeared to return a fully-populated additional section.

Various approaches to normalise the composition of the priming response were considered, including:

- Require use of DNS implementations that exhibit the desired behaviour in the priming response (e.g. NSD) in favour of BIND9;

- Modification of BIND9 (and any other server with similar behaviour) for use by Yeti-Root Servers;

- Isolate the names of Yeti-Root Servers in one or more zones that could be slaved on each Yeti-Root Server, renaming servers as necessary, giving each a source of authoritative data with which the authority section of a priming response could be fully populated. This is the approach used in the Root Server System.

The potential mitigation of renaming all Yeti-Root Servers using a scheme that would allow their names to exist in the balliwick of the

root zone was not considered, since that approach implies the invention of new top-level labels not present in the Root Zone.

Given the relative infrequency of priming queries by individual resolvers and the additional complexity or other compromises implied by each of those mitigations, the decision was made to make no effort to ensure that the composition of priming responses was identical across servers. Even the empty additional sections generated by Yeti-Root Servers running BIND9 seem to be sufficient for all resolver software tested; resolvers simply perform a new recursive lookup for each authoritative server name they need to resolve.

3.6. Yeti-Root Servers

Various volunteers have donated authoritative servers to act as Yeti-Root servers. At the time of writing there are 25 Yeti-Root servers distributed globally, one of which is named using an IDNA2008 [RFC5890] label, shown in the following list in punycode.

Name	Operator	Location
bii.dns-lab.net	BII	CHINA
yet-ns.tsif.net	TSIF	USA
yeti-ns.wide.ad.jp	WIDE Project	Japan
yeti-ns.as59715.net	as59715	Italy
dahu1.yeti.eu.org	Dahu Group	France
ns-yeti.bondis.org	Bond Internet Systems	Spain
yeti-ns.ix.ru	Russia	MSK-IX
yeti.bofh.priv.at	CERT Austria	Austria
yeti.ipv6.ernet.in	ERNET India	India
yeti-dns01.dnsworkshop.org	dnsworkshop /informnis	Germany
yeti-ns.conit.co	CONIT S.A.S	Colombia
dahu2.yeti.eu.org	Dahu Group	France
yeti.aquaray.com	Aqua Ray SAS	France
yeti-ns.switch.ch	SWITCH	Switzerland
yeti-ns.lab.nic.cl	CHILE NIC	Chile
yeti-ns1.dns-lab.net	BII	China
yeti-ns2.dns-lab.net	BII	China
yeti-ns3.dns-lab.net	BII	China
ca...a23dc.yeti-dns.net	Yeti-ZA	South Africa
3f...374cd.yeti-dns.net	Yeti-AU	Australia
yeti1.ipv6.ernet.in	ERNET India	India
xn--r2bi1c.xn--h2bv6c0a.xn--h2brj9c	ERNET India	India
yeti-dns02.dnsworkshop.org	dnsworkshop /informnis	USA
yeti.mind-dns.nl	Monshouwer Internet Diensten	Netherlands
yeti-ns.datev.net	DATEV	Germany

The current list of Yeti-Root server is made available to a participating resolver first using a substitute hints file Appendix A and subsequently by the usual resolver priming process [I-D.ietf-dnsop-resolver-priming]. All Yeti-Root servers are IPv6-only, foreshadowing a future IPv6-only Internet, and hence the Yeti-Root hints file contains no IPv4 addresses and the Yeti-Root zone contains no IPv4 glue.

At the time of writing, all root servers within the Root Server System serve the ROOT-SERVERS.NET zone in addition to the root zone, and all but one also serve the ARPA zone. Yeti-Root servers serve the Yeti-Root zone only.

Significant software diversity exists across the set of Yeti-Root servers, as reported by their volunteer operators:

- Platform: 20 of 25 Yeti-Root servers are implemented on a VPS rather than bare metal.
- Operating System: 6 Yeti-Root servers run on on Linux (Ubuntu, Debian, CentOS, and ArchLinux); 5 run on FreeBSD and 1 on NetBSD.
- DNS software: 18 of 25 Yeti-Root servers use BIND9 (versions varying between 9.9.7 and 9.10.3); four use NSD (4.10 and 4.15); two use Knot (2.0.1 and 2.1.0) and one uses Bundy (1.2.0).

3.7. Traffic Capture and Analysis

Query and response traffic capture is available in the testbed in both Yeti resolvers and Yeti-Root servers in anticipation of experiments that require packet-level visibility into DNS traffic.

Traffic capture is performed on Yeti-Root servers using either dnscap <https://www.dns-oarc.net/tools/dnscap> or pcapdump (part of the pcaputils Debian package <https://packages.debian.org/sid/pcaputils>, with a patch to facilitate triggered file upload <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=545985>. PCAP-format files containing packet captures are uploaded using rsync to central storage.

4. Operational Experience with Yeti DNS Testbed

The following sections provide commentary on the operation and impact analyses of the Yeti-DNS Testbed described in Section 3. More detailed descriptions of observed phenomena are available in Yeti DNS mailing list archives and on the Yeti DNS blog.

4.1. Automated Hints File Maintenance

Renumbering events in the Root Server System are relatively rare. Although each such event is accompanied by the publication of an updated hints file in standard locations, the task of updating local copies of that file used by DNS resolvers is manual, and the process has an observably-long tail: for example, in 2015 J-Root was still receiving traffic at its old address some thirteen years after renumbering [Wessels2015].

The observed impact of these old, deployed hints file is minimal, likely due to the very low frequency of such renumbering events. Even the oldest of hints file would still contain some accurate root server addresses from which priming responses could be obtained.

By contrast, due to the experimental nature of the system and the fact that it is operated mainly by volunteers, Yeti-Root Servers are added, removed and renumbered with much greater frequency. A tool to facilitate automatic maintenance of hints files was therefore created, [hintUpdate].

The automated procedure followed by the hintUpdate tool is as follows.

1. Use the local resolver to obtain a response to the query ./IN/NS;

2. Use the local resolver to obtain a set of IPv4 and IPv6 addresses for each nameserver;

3. Validate all signatures obtained from the local resolvers, and confirm that all data is signed;

4. Compare the data obtained to that contained within the currently-active hints file; if there are differences, rotate the old one way and replace it with a new one.

This tool would not function unmodified when used in the Root Server System, since the names of individual Root Servers (e.g. A.ROOT-SERVERS.NET) are not signed. All Yeti-Root Server names are signed, however, and hence this tool functions as expected in that environment.

4.2. IPv6-only root operation

Yeti DNS testbed was designed to explore whether it can survive in pure IPv6 environment or not. So every root server required to run only with non-EUI64 IPv6 addressed. There are mainly two questions in designers' mind when constructing this testbed: 1) is there any gap between IPv6-only Root and IPv4 Root to provide full function of root server. 2) is it possible that IPv6-only root can serve the Internet, even part of which still only speak IPv4. There are some findings and impacts which IPv6-only property bring to Root system.

4.2.1. Impact of IPv6 fragmentation

In the Root Server System, structural changes with the potential to increase response sizes (and hence fragmentation, fallback to TCP transport or both) have been exercised with great care, since the impact on clients has been difficult to predict or measure. The Yeti DNS Testbed is experimental and has the luxury of a known client base, making it far easier to make such changes and measure their impact.

Many of the experimental design choices described in this document were expected to trigger larger responses. For example, the choice of naming scheme for Yeti-Root Servers described in Section 3.5 defeats label compression the priming response which introduce a large priming response (up to 1754 octets with 25 NS server and their glue) ; the Yeti-Root zone transformation approach described in Section 3.2.2 greatly enlarges the apex DNSKEY RRSet especially during the KSK rollover (up to 1975 octets with 3 ZSK and 2 KSK). An increased incidence of fragmentation was therefore expected.

The Yeti-DNS Testbed provides service on IPv6 only. IPv6 has a fragmentation model that is different from IPv4 -- in particular, fragmentation always takes place on a sending end-host, and not on an intermediate router.

Fragmentation may cause serious issues; if a single fragment is lost, it results in the loss of the entire datagram of which the fragment was a part, and in the DNS frequently triggers a timeout. It is known at this moment that only a limited number of security middle-box implementations support IPv6 fragments. Some public measurements and reports [I-D.taylor-v6ops-fragdrop] [RFC7872] shows that there is notable packets drop rate due to the mistreatment of middle-box on IPv6 fragment. One APNIC study [IPv6-frag-DNS] reported that 37% of endpoints using IPv6-capable DNS resolver cannot receive a fragmented IPv6 response over UDP.

To study the impact, RIPE Atlas probes are used to spot failures like timeout for DNSKEY queries via UDP. For each Yeti server, a Atlas measurement was setup asking for 100 IPv6-enabled probes from 5 regions, in each 2 hours sending DNS query for DNSKEY via UDP with DO bit set. An monitoring report during Yeti KSK rollover shows that statistically large packets will trigger higher failure rate (up to 7%) due to IPv6 fragmentation issues, which accordingly increase probability of retries and TCP fallback. Even within 1500 bytes, when response size reaches 1414 bytes, the failure rate reaches around 2%. Note that ICANN KSK rollover will produce packets exceeding 1414 Bytes.

Regarding the large DNS response via UDP, some existing root servers(A, B, G and J) truncating the response once the large IPv6 packet surpasses 1280 octets. In Yeti DNS Testbed, there are two proposals are discussed and implemented in Yeti experiments.One proposal is called DNS fragments [I-D.muks-dns-message-fragments] which is to fragment the large response in DNS level. Another proposal is called DNS ATR [I-D.song-atr-large-resp] which introduces an simple improvement on authoritative server by replying additional truncated response just after the normal large response.

The consequences of fragmentation were not limited to DNS using UDP transport. There are two cases reported where some Yeti root servers failed to transfer the Yeti-Root zone from a DM. When checking the DM log file, it is found that some root servers experienced " socket is not connected" errors when they pulled the zone file. Further experimentation revealed that combinations of NetBSD 6.1, NetBSD 7.0RC1, FreeBSD 10.0, Debian 3.2 and VMWare ESXI 5.5 resulted in a high TCP MSS value of 1440 octets being negotiated between client and server despite the presence of the IPV6_USE_MIN_MTU socket option, as described in [I-D.andrews-tcp-and-ipv6-use-minmtu]. The mismatch appears to cause outbound segments greater in size than 1280 octets to be dropped before sending.

One proposal to handle this issue is to change the Local TCP MSS to be 1220 (1280-ip6/tcp header)and advise it if IPV6_USE_MINMTU=1. Yeti root from WIDE and SWITCH set this during the test one year ago. Now at the time of writing, 11 out of 25 change the MSS setting in Yeti DNS Testbed.

4.2.2. How IPv6-only Root serve IPv4 users?

Although It is straightforward to setup the IPv6-only root, but it is unknown if it is practical for IPv6-only root to serve the production networks which are still largely speak only in IPv4. In Yeti DNS Testbed it is demonstrated that IPv6-only root can serve the Internet in a incremental approach, even for IPv4 network and users.

It is intuitive to propose to update the resolver to dual-stack and configured it with hint file including IPv6 glues. The dual-stack resolver connects IPv6 root with IPv4-only or dual-stack end users. However, when we approached some partners who agreed to try IPv6-only root in experimental network, they normally do not want to give up the IPv4 root for redundancy reason due to unstable IPv6 network performance. So it is adopted in campuses that one IPv4 resolver address (using current IPv4 addresses of A-M root) and one IPv6 resolver address (using Yeti root) are configured for their customer via DHCPv4 and DHCPv6 respectively. The end users can choose which DNS they use (normally IPv6 first or using Happy eyeballs). Ideally, the end users DNS traffic will largely be sent to the resolver consuming IPv6-only root when IPv6 is widely deployed.

For resolvers who resident in IPv4 only networks, they can forward the query to dual stack resolvers they have trust in. Or they can configure the resolver with a hint file containing a set of IPv4 addresses which are mapped to IPv6 addresses of root in a IPv4/IPv6 translation devices. The query will be routed to the translation devices and forward in IPv6 to IPv6-only root. It is designed and going to be implemented in CERNET2 using IVI [RFC6219] technology.

4.3. Experience on Multiple Signers

In Section 3 it is introduced how three Distributor Masters (DM) works and how they share the control over the Yeti root zone. This section will describe some findings and experiences on its operation.

4.3.1. IXFR fallback to AXFR

In DNS specifications authoritative name server uses full zone transfer (AXFR) [RFC5936], incremental Zone Transfer (IXFR)[RFC1995], and NOTIFY [RFC1996] to achieve coherency of the zone contents. IXFR is an optimization for large DNS zone transfer, which allows server only transfer the changed portion(s) to client. AXFR fallback usually happens at server side by simply returning IXFR client the entire new zone in condition that IXFR server cannot fulfill the given delta-update request.

One experiment in Yeti is designed to test multiple signers with Multiple ZSKs (MZSK). It is required that all public ZSKs used by DMs are included in the zone as a key set; and resolver can validate the message by picking one key from the key set. From DNSSEC point of view, it is technically workable. However, different signers do produce different RRSIG RR which introduces zone inconsistency from beginning in this case. In current setting of Yeti experiment, it is possible that one client does AXFR/IXFR from one server and later asks for IXFR from another server.

It is observed that when the IXFR client switched from one IXFR server to another, it received a IXFR response deleting RRSIG record that does not exist. One IXFR client running NSD 4.1.7 rejected IXFR response, made a log indicating a bad data and then asked for full zone transfer. Luckily, Yeti root zone is relatively small (691K), so the fallback to AXFR does not cause significant performance degeneration. But if operator does host big zone with MZSK model, it will cause problem based on current IXFR.

Another observation is that another IXFR client running Knot 2.1.0 in similar situation just accepts the IXFR response, ignores the differences and generates a merged zone with two RRSIG RRs. It not only produces larger response, but also causes DNSSEC failure when a new zone is generated given that old RRSIG is the signature of old zone RRs.

One possible solutions is asking for development of RRSIG-aware IXFR format in which the RRSIG is treated as a special and RRSIG RR should always be transfered in full (like it does in AXFR). Another solution is adopting the behavior of NSD 4.1.7 as a improvement for

IXFR protocol in which an IXFR client should fall back to AXFR automatically in the event of an IXFR incoherence error.

4.3.2. Latency of Root Zone update

Regarding the timing of Root Zone fetch and soa update, Each Yeti DM checks the root zone serial hourly (in 20 minutes interval) to see if the IANA root zone has changed . A new version of the Yeti root zone is generated if the IANA root zone has changed. In this model, root servers is expected pull the zone from one DM for each new update, because 20 min is expected to be enough for root zone publication. But it is not the true in Yeti testbed in a monitoring test.

It once was reported that one server running on Bundy 1.2.0 on FreeBSD 10.2-RELEASE had some bugs on SOA update with more than 10 hours delay. Besides that server, half of Yeti servers has more than 20 min delay, some even with 40 min delay. One possible reason may be that the server failed to pull the Zone on one DM due to network failure(for example IPv6 fragmentation issue introduce previously) and turn to another DM which introduces the delay. It is also observed that even in the same 20-minutes time frame, not all servers pull from a single DM. It is possible that some servers not use FCFS strategy to pull the zone after they receive the notify. They may pull the zone based on other metrics like the rtt , or manual preference.

4.4. Root Label Compression in Knot

[RFC1035] specifies that domain names can be compressed when encoded in DNS messages, being represented as one of

1. a sequence of labels ending in a zero octet;

2. a pointer; or

3. a sequence of labels ending with a pointer.

The purpose of this flexibility is to reduce the size of domain names encoded in DNS messages.

It was observed that Yeti-Root Servers running knot 2.0 would compress the zero-length label (the root domain, often represented as ".") using a pointer to an earlier example. Although legal, this encoding increases the encoded size of the root label from one octet to two; it was also found to break some client software, in particular the Go DNS library. Bug reports were filed against both knot and the Go DNS library, and both were resolved in subsequent releases.

4.5. Increased ZSK Key Size

The ZSK key size used in the Yeti-DNS Testbed was initially 1024 bits, consistent with the size of the ZSK used in the Root Zone at the time the Yeti DNS Project was started. It later became clear that the ZSK key size in the Root Zone was to be increased.

The ZSK key size in the Yeti-Root zone was subsequently increased in an attempt to identify any unexpected operational effects of doing so.

XXX Note to reviewers: observations following that change to be inserted here. XXX

The ZSK key size in the Root Zone was increased from 1024 bits to 2048 bits in October 2016. [Verisign2016].

4.6. KSK Rollover

The Root Zone KSK is expected to undergo a carefully-orchestrated rollover as described in [ICANN2016]. ICANN has commissioned various tests and has published an external test plan [ICANN2017].

The planned approach was also modelled in the Yeti-DNS Testbed.

XXX Note to reviewers: observations about the KSK rollover in the Yeti-Root zone to be inserted here. XXX

5. IANA Considerations

This document requests no action of the IANA.

6. Acknowledgments

The editors would like to acknowledge the contributions of the various and many subscribers to the Yeti DNS Project mailing lists, including the following people who were involved in the implementation and operation of the Yeti DNS testbed itself:

Tomohiro Ishihara, Antonio Prado, Stephane Bortzmeyer, Mickael Jouanne, Pierre Beyssac, Joao Damas, Pavel Khramtsov, Ma Yan, Otmar Lendl, Praveen Misra, Carsten Strotmann, Edwin Gomez, Remi Gacogne, Guillaume de Lafond, Yves Bovard, Hugo Salgado-Hernandez, Li Zhen, Daobiao Gong, Runxia Wan.

The editors also acknowledge the contributions of the Independent Submissions Editorial Board, and of the following reviewers whose opinions helped improve the clarity of this document:

Subramanian Moonesamy, Joe Abley.

7. References

[hintUpdate]
"Hintfile Auto Update", 2015,
<https://github.com/BII-Lab/Hintfile-Auto-Update>.

[I-D.andrews-tcp-and-ipv6-use-minmtu]
Andrews, M., "TCP Fails To Respect IPV6_USE_MIN_MTU",
draft-andrews-tcp-and-ipv6-use-minmtu-04 (work in
progress), October 2015.

[I-D.ietf-dnsop-resolver-priming]
Koch, P., Larson, M., and P. Hoffman, "Initializing a DNS
Resolver with Priming Queries", draft-ietf-dnsop-resolver-
priming-07 (work in progress), March 2016.

[I-D.muks-dns-message-fragments]
Sivaraman, M., Kerr, S., and D. Song, "DNS message
fragments", draft-muks-dns-message-fragments-00 (work in
progress), July 2015.

[I-D.song-atr-large-resp]
Song, L., "ATR: Additional Truncated Response for Large
DNS Response", draft-song-atr-large-resp-00 (work in
progress), September 2017.

[I-D.taylor-v6ops-fragdrop]
Jaeggli, J., Colitti, L., Kumari, W., Vyncke, E., Kaeo,
M., and T. Taylor, "Why Operators Filter Fragments and
What It Implies", draft-taylor-v6ops-fragdrop-02 (work in
progress), December 2013.

[ICANN2016]
"Root Zone KSK Rollover Plan", 2016,
<https://www.iana.org/reports/2016/
root-ksk-rollover-design-20160307.pdf>.

[ICANN2017]
"2017 KSK Rollover External Test Plan", July 2016,
<https://www.icann.org/en/system/files/files/
ksk-rollover-external-test-plan-22jul16-en.pdf>.

[IPv6-frag-DNS]
"Dealing with IPv6 fragmentation in the DNS", August 2017,
<https://blog.apnic.net/2017/08/22/
dealing-ipv6-fragmentation-dns>.

[ISC-TN-2003-1]
Abley, J., "Hierarchical Anycast for Global Service Distribution", March 2003, <http://ftp.isc.org/isc/pubs/tn/isc-tn-2003-1.txt>.

[ITI2014] "Identifier Technology Innovation Report", May 2014, <https://www.icann.org/en/system/files/files/iti-report-15may14-en.pdf>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, DOI 10.17487/RFC1995, August 1996, <https://www.rfc-editor.org/info/rfc1995>.

[RFC1996] Vixie, P., "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", RFC 1996, DOI 10.17487/RFC1996, August 1996, <https://www.rfc-editor.org/info/rfc1996>.

[RFC2826] Internet Architecture Board, "IAB Technical Comment on the Unique DNS Root", RFC 2826, DOI 10.17487/RFC2826, May 2000, <https://www.rfc-editor.org/info/rfc2826>.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <https://www.rfc-editor.org/info/rfc2845>.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, DOI 10.17487/RFC5011, September 2007, <https://www.rfc-editor.org/info/rfc5011>.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <https://www.rfc-editor.org/info/rfc5890>.

[RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010, <https://www.rfc-editor.org/info/rfc5936>.

[RFC6219] Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, "The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition", RFC 6219, DOI 10.17487/RFC6219, May 2011, <https://www.rfc-editor.org/info/rfc6219>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

[RFC7720] Blanchet, M. and L-J. Liman, "DNS Root Name Service Protocol and Deployment Requirements", BCP 40, RFC 7720, DOI 10.17487/RFC7720, December 2015, <https://www.rfc-editor.org/info/rfc7720>.

[RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <https://www.rfc-editor.org/info/rfc7872>.

[RRL] Vixie, P. and V. Schryver, "Response Rate Limiting (RRL)", June 2012, <http://www.redbarn.org/dns/ratelimits>.

[RSSAC001] "Service Expectations of Root Servers", December 2015, <https://www.icann.org/en/system/files/files/rssac-001-root-service-expectations-04dec15-en.pdf>.

[RSSAC023] "History of the Root Server System", November 2016, <https://www.icann.org/en/system/files/files/rssac-023-04nov16-en.pdf>.

[TNO2009] Gijsen, B., Jamakovic, A., and F. Roijers, "Root Scaling Study: Description of the DNS Root Scaling Model", September 2009, <https://www.icann.org/en/system/files/files/root-scaling-model-description-29sep09-en.pdf>.

[Verisign2016] Wessels, D., "Increasing the Strength of the Zone Signing Key for the Root Zone", May 2016, <https://blog.verisign.com/security/increasing-the-strength-of-the-zone-signing-key-for-the-root-zone/>.

[Wessels2015]
Wessels, D., "Thirteen Years of "Old J-Root"", 2015, <https://indico.dns-oarc.net/event/24/session/10/contribution/10/material/slides/0.pdf>.

Appendix A. Yeti-Root Hints File

The following hints file (complete and accurate at the time of writing) causes a DNS resolver to use the Yeti DNS testbed in place of the production Root Server System and hence participate in experiments running on the testbed.

Note that some lines have been wrapped in the text that follows in order to fit within the production constraints of this document. Wrapped lines are indicated with a blackslash character ("\"), following common convention.

```
.                       3600000  IN   NS     bii.dns-lab.net
bii.dns-lab.net         3600000  IN   AAAA   240c:f:1:22::6
.                       3600000  IN   NS     yeti-ns.tisf.net
yeti-ns.tisf.net        3600000  IN   AAAA   2001:559:8000::6
.                       3600000  IN   NS     yeti-ns.wide.ad.jp
yeti-ns.wide.ad.jp      3600000  IN   AAAA   2001:200:1d9::35
.                       3600000  IN   NS     yeti-ns.as59715.net
yeti-ns.as59715.net     3600000  IN   AAAA   \
                            2a02:cdc5:9715:0:185:5:203:53
.                       3600000  IN   NS     dahu1.yeti.eu.org
dahu1.yeti.eu.org       3600000  IN   AAAA   \
                            2001:4b98:dc2:45:216:3eff:fe4b:8c5b
.                       3600000  IN   NS     ns-yeti.bondis.org
ns-yeti.bondis.org      3600000  IN   AAAA   2a02:2810:0:405::250
.                       3600000  IN   NS     yeti-ns.ix.ru
yeti-ns.ix.ru           3600000  IN   AAAA   2001:6d0:6d06::53
.                       3600000  IN   NS     yeti.bofh.priv.at
yeti.bofh.priv.at       3600000  IN   AAAA   2a01:4f8:161:6106:1::10
.                       3600000  IN   NS     yeti.ipv6.ernet.in
yeti.ipv6.ernet.in      3600000  IN   AAAA   2001:e30:1c1e:1::333
.                       3600000  IN   NS     yeti-dns01.dnsworkshop.org
yeti-dns01.dnsworkshop.org \
                        3600000  IN   AAAA   2001:1608:10:167:32e::53
.                       3600000  IN   NS     yeti-ns.conit.co
yeti-ns.conit.co        3600000  IN   AAAA   \
                            2604:6600:2000:11::4854:a010
.                       3600000  IN   NS     dahu2.yeti.eu.org
dahu2.yeti.eu.org       3600000  IN   AAAA   2001:67c:217c:6::2
.                       3600000  IN   NS     yeti.aquaray.com
```

```
yeti.aquaray.com      3600000  IN   AAAA   2a02:ec0:200::1
.                     3600000  IN   NS     yeti-ns.switch.ch
yeti-ns.switch.ch     3600000  IN   AAAA   2001:620:0:ff::29
.                     3600000  IN   NS     yeti-ns.lab.nic.cl
yeti-ns.lab.nic.cl    3600000  IN   AAAA   2001:1398:1:21::8001
.                     3600000  IN   NS     yeti-ns1.dns-lab.net
yeti-ns1.dns-lab.net  3600000  IN   AAAA   2001:da8:a3:a027::6
.                     3600000  IN   NS     yeti-ns2.dns-lab.net
yeti-ns2.dns-lab.net  3600000  IN   AAAA   2001:da8:268:4200::6
.                     3600000  IN   NS     yeti-ns3.dns-lab.net
yeti-ns3.dns-lab.net  3600000  IN   AAAA   2400:a980:30ff::6
.                     3600000  IN   NS     \
                        ca978112ca1bbdcafac231b39a23dc.yeti-dns.net
ca978112ca1bbdcafac231b39a23dc.yeti-dns.net \
                      3600000  IN   AAAA   2c0f:f530::6
.                     3600000  IN   NS     \
                        3e23e8160039594a33894f6564e1b1.yeti-dns.net
3e23e8160039594a33894f6564e1b1.yeti-dns.net \
                      3600000  IN   AAAA   2803:80:1004:63::1
.                     3600000  IN   NS     \
                        3f79bb7b435b05321651daefd374cd.yeti-dns.net
3f79bb7b435b05321651daefd374cd.yeti-dns.net \
                      3600000  IN   AAAA   2401:c900:1401:3b:c::6
.                     3600000  IN   NS     \
                        xn--r2bi1c.xn--h2bv6c0a.xn--h2brj9c
xn--r2bi1c.xn--h2bv6c0a.xn--h2brj9c \
                      3600000  IN   AAAA   2001:e30:1c1e:10::333
.                     3600000  IN   NS     yeti1.ipv6.ernet.in
yeti1.ipv6.ernet.in   3600000  IN   AAAA   2001:e30:187d::333
.                     3600000  IN   NS     yeti-dns02.dnsworkshop.org
yeti-dns02.dnsworkshop.org \
                      3600000  IN   AAAA   2001:19f0:0:1133::53
.                     3600000  IN   NS     yeti.mind-dns.nl
yeti.mind-dns.nl      3600000  IN   AAAA   2a02:990:100:b01::53:0
```

Appendix B. Controversy

The Yeti DNS Project, its infrastructure and the various experiments that have been carried out using that infrastructure, have been described by people involved in the project in many public meetings at technical venues since its inception. The mailing lists using which the operation of the infrastructure has been coordinated are open to join, and their archives are public. The project as a whole has been the subject of robust public discussion.

Some commentators have expressed concern that the Yeti DNS Project is, in effect, operating an "alternate root," challenging the IAB's

comments published in [RFC2826]. Other such alternate roots are considered to have caused end-user confusion and instability in the namespace of the DNS by the introduction of new top-level labels or the different use of top-level labels present in the Root Server System. The coordinators of the Yeti DNS Project do not consider the Yeti DNS Project to be an alternate root in this sense, since by design the namespace enabled by the Yeti-Root Zone is identical to that of the Root Zone.

Some commentators have expressed concern that the Yeti DNS Project seeks to influence or subvert administrative policy relating to the Root Server System, in particular in the use of DNSSEC trust anchors not published by the IANA and the use of Yeti-Root Servers in regions where governments or other organisations have expressed interest in operating a Root Server. The coordinators of the Yeti-Root project observe that their mandate is entirely technical and has no ambition to influence policy directly; they do hope, however, that technical findings from the Yeti DNS Project might act as a useful resource for the wider technical community.

Finally, some concern has been expressed about the possible applications of the Yeti DNS Project to the governments of countries where access to the Internet is subject to substantial centralised control, in contrast to most other jurisdictions where such controls are either lighter or not present. The coordinators of the Yeti DNS Project have taken care to steer all discussions and related decisions about the technical work of the project to public venues in the interests of full transparency, and encourage anybody concerned about the decision-making process to participate in those venues and review their archives directly.

Appendix C. About This Document

This section (and sub-sections) has been included as an aid to reviewers of this document, and should be removed prior to publication.

C.1. Venue

The authors propose that this document proceeed as an Independent Submission, since it documents work that, although relevant to the IETF, has been carried out externally to any IETF working group. However, a suitable venue for discussion of this document is the dnsop working group.

Information about the Yeti DNS project and discussion relating to particular experiments described in this document can be found at <https://yeti-dns.org/>.

This document is maintained in GitHub at <https://github.com/BII-Lab/yeti-testbed-experience>.

C.2. Revision History

C.2.1. draft-song-yeti-testbed-experience-00 through -03

Change history is available in the public GitHub repository where this document is maintained: <https://github.com/BII-Lab/yeti-testbed-experience>.

C.2.2. draft-song-yeti-testbed-experience-04

Substantial editorial review and rearrangement of text by Joe Abley at request of BII.

Added what is intended to be a balanced assessment of the controversy that has arisen around the Yeti DNS Project, at the request of the Independent Submissions Editorial Board.

Changed the focus of the document from the description of individual experiments on a Root-like testbed to the construction and motivations of the testbed itself, since that better describes the output of the Yeti DNS Project to date. In the considered opinion of this reviewer, the novel approaches taken in the construction of the testbed infrastructure and the technical challenges met in doing so are useful to record, and the RFC series is a reasonable place to record operational experiences related to core Internet infrastructure.

Note that due to draft cut-off deadlines some of the technical details described in this revision of the document may not exactly match operational reality; however, this revision provides an indicative level of detail, focus and flow which it is hoped will be helpful to reviewers.

Authors' Addresses

Linjian Song (editor)
Beijing Internet Institute
2508 Room, 25th Floor, Tower A, Time Fortune
Beijing 100028
P. R. China

Email: songlinjian@gmail.com
URI: http://www.biigroup.com/

Dong Liu (editor)
Beijing Internet Institute
2508 Room, 25th Floor, Tower A, Time Fortune
Beijing 100028
P. R. China

Email: dliu@biigroup.com
URI: http://www.biigroup.com/

Paul Vixie (editor)
TISF
11400 La Honda Road
Woodside, California 94062
US

Email: vixie@tisf.net
URI: http://www.redbarn.org/

Akira Kato (editor)
Keio University/WIDE Project
Graduate School of Media Design, 4-1-1 Hiyoshi, Kohoku
Yokohama 223-8526
JAPAN

Email: kato@wide.ad.jp
URI: http://www.kmd.keio.ac.jp/

Shane Kerr
Antoon Coolenlaan 41
Uithoorn 1422 GN
NL

Email: shane@time-travellers.org