Domain Name System Operations
Internet-Draft
Intended status: Informational
Expires: 3 December 2026

W. Hardaker
Google, Inc.
W. Kumari
Google

1 June 2026

DNS Root Server System Usage Considerations
draft-hardaker-dnsop-rss-usage-considerations-01

Abstract

This document explores various technologies developed to enhance the DNS, focusing specifically on interactions with the DNS Root Server System (RSS). It examines a number of the protocols and evaluates their impact on communication with the RSS.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-hardaker-dnsop-rss-usage-considerations/.

Discussion of this document takes place on the Domain Name System Operations Working Group mailing list (mailto:dnsop@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/dnsop/. Subscribe at https://www.ietf.org/mailman/listinfo/dnsop/.

Source for this draft and an issue tracker can be found at https://github.com/https://github.com/hardaker/draft-hardaker-dnsop-rss-usage-considerations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 December 2026.

Table of Contents

1. Introduction

This document explores various technologies developed to enhance the DNS, focusing specifically on interactions with the DNS Root Server System (RSS). It examines a number of the protocols and evaluates their impact on communication with the RSS.

While the necessity of a centralized source for a unique internet naming system is beyond the scope of this document, it is thoroughly addressed in [RFC2826].

The document begins by briefly describing and referencing various communication enhancements in Section 2. It then provides an analysis of how these enhancements impact communication with the RSS in Section 3.

1.1. Document Conventions

To evaluate the potential changes to RSS communication in Section 3, this document categorizes the solutions using the following keywords:

* *Minimal*: The technique addresses a problem with only a minimal amount of improvement.

* *Moderate*: The technique provides a moderate level of improvement in addressing a problem.

* *Significant*: The technique offers substantial improvement for communicating with the RSS, even though it does not entirely address the problem space.

* *Complete*: The technique fully enhances communication with the RSS and/or completely mitigates the defined concern.

2. Techniques Improving Communication with the RSS

This section outlines various techniques designed to improve communication with the DNS Root Server System (RSS), particularly in addressing security or efficiency concerns. These techniques are further analyzed in Section 3 to evaluate their effectiveness in mitigating each of the concerns.

2.1. QName Minimization

The original DNS protocol specifications [RFC1035] indicated that the entire query name being handled by a resolver should be sent to upstream authoritative servers; this behavior leaks all the labels in a query to all the authoritative servers used in the resolution process, even when the authoritative server doesn’t use all the labels when generating a response.

The "DNS Query Name Minimisation to Improve Privacy" [RFC9156] specification describes how recursive resolvers can minimize this privacy leakage by describing how the resolver "no longer always sends the full original QNAME and original QTYPE to the upstream name server."

2.2. Aggressive NSEC

The "Aggressive Use of DNSSEC-Validated Cache" [RFC8198] [RFC9077] specification describes how validating recursive resolvers can reduce the number of queries sent to authoritative servers by allowing "DNSSEC-validating resolvers to generate negative answers within a range and positive answers from wildcards."

(Ed note: The NSEC example used in this paragraph is an accurate example as of this writing, but may change over time.) Aggressive NSEC leverages NSEC records to prevent redundant queries for non-existent TLDs. Validating resolvers can use NSEC records to synthesize negative responses for non-existent TLDs based on previously received NSEC records. For example, a query for a non-existent TLD (e.g., ".example") will return an NSEC record cryptographically proving that the no names between ".events" and ".exchange" exist. Subsequent queries within the NSEC TTL for a non-existent TLD that falls between ".events" and ".exchange" (e.g., ".evil") can be answered immediately without sending a query to the RSS.

This technique is particularly effective in reducing queries to the RSS for non-existent TLDs, as once a single query between two valid TLDs has been sent, validating resolvers can make use of the returned NSEC records to prevent future queries between the two bounding TLDs from needing resolution. This improves both privacy (Section 3.1) and latency (Section 3.5) when communicating with the RSS, as fewer queries are sent and more responses can be generated immediately from the cache.

2.3. Encrypted and Authenticated DNS

There are a variety of protocols that enable encrypted DNS transactions both between stubs and recursive resolvers, and between recursive resolvers and authoritative servers. These include "DNS over Transport Layer Security" (TLS) [RFC7858] and "DNS over Datagram Transport Layer Security (DTLS)" [RFC8094] (along with supplemental information [RFC8310]) which collectively are referred to as "DNS over (D)TLS".

In addition, "DNS Queries over HTTPS (DoH)" [RFC8484], "DNS over Dedicated QUIC Connections" [RFC9250], and "Oblivious DNS over HTTPS" [RFC9230] enable DNS over encrypted HTTP transports.

The "Unilateral Opportunistic Deployment of Encrypted Recursive-to-Authoritative" DNS [RFC9539] specification defines how recursive resolvers can communicate with authoritative servers that support encrypted TLS sessions. However, the specification is currently published under an EXPERIMENTAL status.

In all cases for the purpose of this document, we define these these connections to be verified both in terms of authenticity of the server end point (although not necessarily of the data’s origin itself). As such this survey frequently rates authenticated and encrypted TLS solutions as strong candidates for solving communication problems, large scale deployments of resolver to authoritative DNS servers in particular lack a robust way for performing proper TLS server certificate authentication, which makes actual deployment a challenge.

2.4. Serve Stale

The "Serving Stale Data to Improve DNS Resiliency" [RFC8767] specification specifies how resolvers can continue to serve previously cached records even after their Time-To-Live (TTL) has expired. This approach enhances DNS resiliency by ensuring that responses remain available during periods when authoritative servers are unreachable, such as during network outages or server failures.

2.5. DNSSEC

DNSSEC [RFC9364] provides cryptographic assurance of the authenticity and integrity of DNS responses. Using digital signatures, DNSSEC ensures that data from the root and other signed zones cannot be maliciously modified without detection. This allows validating resolvers, and their clients, to verify the origin, authenticity, and correctness of DNS data.

2.6. LocalRoot

LocalRoot enables recursive resolvers to maintain and use a local copy of the root zone, eliminating the need to query the root servers directly. This concept has been documented for over a decade in [RFC7706], [RFC8806], and [LOCALROOT], and in academic research [NOROOTS], [ROOTPRIVACY]. It is implemented in [BINDMIRROR], [KNOTMODULE], [UNBOUNDAUTHZONE], [LOCALROOTISI].

The initial LocalRoot implementations relied on AXFR for transferring root zone data. More recent implementations instead support HTTP-based transfers, providing additional flexibility and scalability.

3. RSS Communication Improvement Effectiveness Comparison

This section evaluates the impact of the techniques described in Section 2 on recursive resolvers' communication with the Root Server System (RSS).

3.1. Privacy

Queries sent to the RSS include those within existing Top-Level Domains (TLDs) (e.g., ".com", ".org") and for queries under non-existent domains (e.g., "sensitive.internal", sensitive.con" (sic)).

When a resolver's cache lacks an answer for a domain within an associated TLD, the query is forwarded to the RSS. This exposes the query to the 12 Root Server Operators (RSOs) managing the 26 RSS identifiers (13 IPv4 and 13 IPv6) and the networks in between.

The privacy sensitivity of queries sent to the RSS can vary widely, ranging from unlikely sensitive (such as a query for just ".example" without any left-hand labels) to more critical queries that leak potentially personal or system-sensitive information that was not intended to leak beyond an internal network boundary (such as ".wpad" records). Names reaching the RSS can be single labels that reveal only the TLD's name (".com" or ".xxx"), which may or may not be sensitive in nature by themselves. Queries can also contain more labels that may leak more sensitive information ("private.sensitive.example").

Accidental leaks can stem from typos, web browser keyword searches, misconfigured software, or simply because it needed to be resolved, and no privacy-protecting techniques are deployed.

Note that beyond the analysis of a single record being observed, a larger or temporal analysis of all of a client's queries may reveal additional information and/or behavioral patterns ([ROOTPRIVACY]).

For example, the collection of unique ccTLDs observed during the course of a 24 hour period may reveal the political bias in a resolver’s clients.

To mitigate issues with potentially sensitive queries leaving a resolver, various techniques are available for use that include:

* *Aggressive NSEC: Significant*

  Because Aggressive NSEC greatly reduces the quantity of queries requiring NXDOMAIN responses, it can greatly enhance a resolver’s privacy by simply sending less queries.

  The rough worst-case scenario with a long lived cache is a transmission of 1 query per TLD in the root zone in the course of one TTL (2 days, or other implementation upper limit which can commonly be 1 day). Note that resolvers that prefer client NS records, which often have a lower TTL, may send data more frequently than what the root zone’s TTL specifies. Note that DNSSEC (or at least an understanding of the NSEC record) is required to implement Aggressive NSEC.

  Note that Aggressive NSEC does not prevent queries for existing TLDs from leaking.

* *QName Minimization: Significant*

  QName Minimisation greatly improves privacy in the case where any sensitive information exists in the labels before the TLD (e.g. sensitive.example). However, this cannot entirely minimize the leakage of TLD names themselves, which may or may not be sensitive in nature (".xxx" is used as a common example of a TLD that may be considered sensitive). Note that like the Aggressive NSEC technique above, the sent queries are typically cached for a period of time. Unlike NSEC Aggressive Caching though, DNSSEC is not required to implement QName Minimization.

* *Encrypted and authenticated DNS: Moderate*

  Encrypted and authenticated DNS protocols, such as DNS over (D)TLS, protect queries from intermediate observers by encrypting communication. However, as of this writing, only 2 of the 13 root server identifiers support encrypted DNS, limiting the effectiveness of this technique with respect to the RSS.

* *LocalRoot: Complete*

LocalRoot implementations maintain a local copy of the root zone, thereby completely eliminating the need to send queries to the RSS. This ensures complete privacy with respect the RSS, as no queries leave the resolver toward the RSS.

Furthermore, because the data is received and verified before being used, there are only two remaining sources of trust for the information used: IANA itself and the RZM which is responsible for creating the root zone, although even they have no visibility into how resolvers make use of the data.

3.2. Disconnected operations

At times, a region may become disconnected from the broader internet due to various causes, such as network outages, intentional disruptions, or natural disasters. In such scenarios, the Root Server System (RSS), as the pinnacle of the DNS hierarchy, becomes inaccessible to resolvers needing information about TLDs not in their cache. While a complete disconnection from the internet results in failures for all resolutions to external resources, local infrastructure may still be functioning and remain reachable (e.g., a ccTLD may be accessible even if the RSS is not).

Solutions available for resolvers to continue operating even when disconnected from the RSS include:

* *Serve Stale: Significant*

  Serve Stale allows resolvers to reuse cached data when authoritative servers are unreachable. This approach can significantly aid disconnected scenarios, provided the required records are already in the cache. However, it cannot assist when the necessary information has not been recently cached.

* *LocalRoot: Significant or Complete*

  LocalRoot implementations that populate their cache with root zone records offer significant protection, similar to Serve Stale. But cached records may still expire if not recently accessed.

  Implementations that ensure the root zone contents are always available (e.g., via RFC8806 parallel infrastructure, special cache retention flags, or when loaded from a persistently refreshed file) provide complete protection against RSS disconnection.

3.3. Record Protection

DNSSEC RRSIG records ensure the integrity and authenticity of DNS resource records (RRs). However, delegation-related records, such as NS records and associated glue records, are exceptions to this protection. These records, served by the parent zone, assist resolvers in reaching child zones but are not cryptographically signed.

Once the resolver verifies that the child zone is serving accurate information and the DS record validates the child's DNSKEY, the child's data becomes authoritative. If the parent-side delegation records are modified, resolvers may initially be directed to incorrect infrastructure. With DNSSEC validation enabled, this would result in a denial of service or, at worst, a temporary eavesdropping issue.

We analyze record protection by dividing it into in two parts: authoritative RR protection and non-authoritative delegation record protection (NS and glue).

3.3.1. Authoritative RR Protection

All root zone records, except NS and glue records, are protected by DNSSEC, ensuring tamper resistance. Solutions for safeguarding these records include:

* *DNSSEC: Significant*

  DNSSEC protects against record modification for records served from the RSS, assuming validation is performed using a root zone DNSSEC trust anchor and the chain of trust from it is followed all the way to the child zone. Note that not all TLDs in the root zone are protected, and thus this is considered Significant since most TLDs do offer DNSSEC support and most resolvers are child-centric.

  Note that DNSSEC, when combined with NSEC records, allows verification of negative answers received from the root. Thus, responses for non-existent records from the root are verifiable as authentic.

* *Encrypted and Authenticated DNS: Complete*

  If the resolver is able to connect to a root server instance that offers authenticated and encrypted DNS support, then any answers they receive over that protected path can be considered properly validated even without checking the corresponding DNSSEC records.

However, checking the DNSSEC records for validity themselves may still be recommended. Encrypted and authenticated DNS protection is considered Complete when the authentication of the TLS connection to the RSS can be properly verified.

* *LocalRoot: Complete*

  LocalRoot implementations download and verify the entire root zone using DNSSEC and ZONEMD records, ensuring all data is tamper-resistant. Proper DNSSEC validation of at least the ZONEMD record is required.

3.3.2. Non-Authoritative Data (Glue) Protection

Although DNSSEC protects many of the records within the root zone, the TLD's NS, A and AAAA records in the root zone are not signed. This lack of signing leaves these records vulnerable to attacks such as man-in-the-middle modifications or cache injection, especially with parent-centric validating resolvers.

These attacks could redirect traffic to non-responsive servers, causing denial-of-service issues.

Alternatively, the addresses can be modified to point to alternate addresses that do respond. While these responding addresses will be unable to alter DNSSEC signed records in the root zone, they can still act as eavesdroppers and modify any unsigned glue records being passed.

Note that NS and glue records from the root zone are typically cached for a lengthy period of time, which is a benefit for resolvers that receive the correct records but a detriment for those that receive modified records and have a parent-side preference.

Mitigation strategies include:

* *DNSSEC: None to Significant*

  DNSSEC prevents unauthorized modification of authoritative records in DNS zones, ensuring that unsigned data cannot be falsely inserted. However, as discussed above, it does not prevent NS and glue record modification. The protection offered by DNSSEC depends on whether the resolver uses DNSSEC to validate the child side's NS, A and AAAA records. Furthermore, the TLD must be signed for this protection to be effective.

* *Encrypted and authenticated DNS: Complete*

Encrypted and authenticated DNS provides secured communication with root server instances, assuming server endpoints are properly verified. Note, however, since NS glue records in the parent zone lack RRSIGs, proper validation the veracity of the data still requires consultation with the child zone for data authenticity verification.

* *LocalRoot: Complete*

  LocalRoot implementations download and verify the entire contents of the root zone, including NS and glue records, effectively eliminating this threat.

3.4. Bit Flipping

"Bit flipping" is the term used to describe accidental modifications to bits due to memory corruption in a device or during transmission. The net effect is that at least one bit may flip randomly from 0 to 1, or vice versa. Though rare, they have been measured in network traffic arriving at very popular servers of all types.

The root-servers.net zone is, unsurprisingly, a very popular domain: it bootstraps all Internet DNS resolutions. Researchers have shown that by registering alternate domain names with single or double bit flips in the root-servers.net domain name allows these alternate servers to receive requests that were intended to be sent to the real root-servers.net domain. These bit flips can cause problems similar to the above discussed glue record modifications (Section 3.3.2).

Note that in this section we only discuss bitflips that are received by or sent by the resolver. Bitflips that occur in packets leaving the resolver toward the client that submitted the original query are out of scope and not covered in this document as the resolver (and the RSS) has no control over them.

Solutions to detecting and rejecting bitflipped data include:

* *DNSSEC: Significant*

  Cryptographic techniques like DNSSEC properly identify and reject data with modifications of any kind, including bit flipping techniques. However, DNSSEC does not prevent NS and glue record modification since these records, as discussed above, are not protected by DNSSEC unless verified through to the client’s copy of the records.

  Research has shown that some validating resolvers fail to detect when some bit flipping situations have occurred, however.

* *Encrypted and authenticated DNS: Significant*

  Encrypted and authenticated DNS provides secured communication with root server instances that ensures no data has been modified in flight. However, any data containing bit-flips in the root server instance itself, in the resolver's memory, or in the outgoing data transmissions cannot be protected.

* *LocalRoot: Significant*

  LocalRoot implementations within resolvers download and verify the entire contents of the root zone using DNSSEC and ZONEMD, including associated glue records. This eliminates (or at least catches) all bit flips that occur on incoming data transfers for the root zone. However, it cannot protect against bit flips that occur in-memory or in outgoing responses, and is thus only Significant.

3.5. Latency

Even though almost all answers to user queries are served from the cache, some resolver operators have concerns about the latency of queries sent to the RSS. In addition, because negative answers are frequent and may be from end-user typos waiting for a response, latency to the RSS may at least matter a little. In fact, this is one motivation listed in [RFC8806] for implementing LocalRoot.

Techniques that support reducing latency to the root, often by having the answers already available, include:

* *Aggressive NSEC: Significant*

  With Aggressive NSEC deployed, queries containing right-most labels (TLD labels) that are not in the root may be answered immediately by generating the answer using an NSEC record in the (validating) resolver's cache. The result is similar to the privacy analysis showing that Aggressive NSEC provides significant latency reduction to the root zone.

* *LocalRoot: Complete*

  As above, a LocalRoot implementation already has all the records in the root zone and thus can answer immediately and without ever sending any queries to the RSS.

* *Serve Stale: N/A*

Note that though Serve Stale may have an answer in the cache that is usable, it does not help with latency since the answer should not be used until an attempt to query the RSS has already been made.

4. Summary

In summary, the following table summarizes the analysis in Section 3 given the DNS communication technologies in Section 2 and how they affect communication with the RSS.

	QName-Min	Aggr.-NSEC	Encr/ Auth	Serve-Stale	DNSSEC	LocalRoot
Privacy	Signif	Signif	Moderate			Complete
Disconnection				Signif		Complete*
Auth Prot			Complete		Signif	Complete
Non-auth Prot			Complete		Signif	Complete
Bit Flipping			Signif		Signif	Signif
Latency		Signif				Complete

Table 1

(*): as discussed above, this depends on the implementation with some implementations only being Significant while others are Complete.

5. Operational Considerations

TBD

6. Security Considerations

This document discusses a large number of security related cases in Section 3 and proposes mitigation strategies, their effectiveness, and associated trade-offs.

7. IANA Considerations

None.

8. Informative References

[BINDMIRROR]
"bind instructions for mirroring the root zone", n.d., <https://bind9.readthedocs.io/en/v9.18.41/reference.html>.

[DITL] "A Day In The Life of the Internet", n.d., <https://www.dns-oarc.net/oarc/data/ditl>.

[DNSMAGNITUDE]
"ICANN DNS Magnitude statistics page", n.d., <https://magnitude.research.icann.org/>.

[DNSMAGNITUDE2020]
"DNS Magnitude - A Popularity Figure for Domain Names, and its Application to L-root Traffic", n.d., <https://www.icann.org/en/system/files/files/dns-magnitude-05aug20-en.pdf>.

[KNOTMODULE]
"know module to support LocalRoot", n.d., <https://knot-resolver.readthedocs.io/en/latest/lib.html>.

[LOCALROOT]
"Populating resolvers with the root zone", n.d., <https://datatracker.ietf.org/doc/draft-wkumari-dnsop-localroot-bcp/>.

[LOCALROOTISI]
"The LocalRoot project to help operators use LocalRoot", n.d., <https://localroot.isi.edu/>.

[NOROOTS] "On Eliminating Root Nameservers from the DNS", n.d., <https://www.icir.org/mallman/pubs/All19b/All19b.pdf>.

[QUERYCOMPOSITION]
"Understanding DNS Query Composition at B-Root", n.d., <https://arxiv.org/pdf/2308.07966>.

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

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

[RFC7706] Kumari, W. and P. Hoffman, "Decreasing Access Time to Root Servers by Running One on Loopback", RFC 7706, DOI 10.17487/RFC7706, November 2015, <https://www.rfc-editor.org/rfc/rfc7706>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/rfc/rfc7858>.

[RFC8094] Reddy, T., Wing, D., and P. Patil, "DNS over Datagram Transport Layer Security (DTLS)", RFC 8094, DOI 10.17487/RFC8094, February 2017, <https://www.rfc-editor.org/rfc/rfc8094>.

[RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", RFC 8198, DOI 10.17487/RFC8198, July 2017, <https://www.rfc-editor.org/rfc/rfc8198>.

[RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", RFC 8310, DOI 10.17487/RFC8310, March 2018, <https://www.rfc-editor.org/rfc/rfc8310>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/rfc/rfc8484>.

[RFC8767] Lawrence, D., Kumari, W., and P. Sood, "Serving Stale Data to Improve DNS Resiliency", RFC 8767, DOI 10.17487/RFC8767, March 2020, <https://www.rfc-editor.org/rfc/rfc8767>.

[RFC8806] Kumari, W. and P. Hoffman, "Running a Root Server Local to a Resolver", RFC 8806, DOI 10.17487/RFC8806, June 2020, <https://www.rfc-editor.org/rfc/rfc8806>.

[RFC9077] van Dijk, P., "NSEC and NSEC3: TTLs and Aggressive Use", RFC 9077, DOI 10.17487/RFC9077, July 2021, <https://www.rfc-editor.org/rfc/rfc9077>.

[RFC9156] Bortzmeyer, S., Dolmans, R., and P. Hoffman, "DNS Query Name Minimisation to Improve Privacy", RFC 9156, DOI 10.17487/RFC9156, November 2021, <https://www.rfc-editor.org/rfc/rfc9156>.

[RFC9230] Kinnear, E., McManus, P., Pauly, T., Verma, T., and C.A. Wood, "Oblivious DNS over HTTPS", RFC 9230, DOI 10.17487/RFC9230, June 2022, <https://www.rfc-editor.org/rfc/rfc9230>.

[RFC9250] Huitema, C., Dickinson, S., and A. Mankin, "DNS over Dedicated QUIC Connections", RFC 9250, DOI 10.17487/RFC9250, May 2022, <https://www.rfc-editor.org/rfc/rfc9250>.

[RFC9364] Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <https://www.rfc-editor.org/rfc/rfc9364>.

[RFC9539] Gillmor, D. K., Ed., Salazar, J., Ed., and P. Hoffman, Ed., "Unilateral Opportunistic Deployment of Encrypted Recursive-to-Authoritative DNS", RFC 9539, DOI 10.17487/RFC9539, February 2024, <https://www.rfc-editor.org/rfc/rfc9539>.

[ROOTPRIVACY] "Analyzing and mitigating privacy with the DNS root service", n.d., <http://www.isi.edu/~hardaker/papers/2018-02-ndss-analyzing-root-privacy.pdf>.

[UNBOUNDAUTHZONE] "Unbound documentation for supporting LocalRoot", n.d., <https://unbound.docs.nlnetlabs.nl/en/latest/manpages/unbound.conf.html>.

Acknowledgments

Thank you to John Heidemann who provided valuable feedback.

Authors' Addresses

Wes Hardaker
Google, Inc.
Email: ietf@hardakers.net

Warren Kumari
Google
Email: warren@kumari.net