

Network Working Group  
Internet-Draft  
Expires: January 15, 2005

R. Austein  
ISC  
July 17, 2004

**EDNS NSID Extension**  
**draft-austein-dnsext-nsid-01**

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on January 15, 2005.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

With the increased use of DNS anycast, load balancing, and other mechanisms allowing more than one DNS name server to share a single IP address, it is sometimes difficult to tell which of a pool of name servers has answered a particular query. While existing ad-hoc mechanism allow an operator to send follow-up queries when it is necessary to debug such a configuration, the only completely reliable way to obtain the identity of the name server which actually responded is to have the name server include this information in the response itself. This note proposes a protocol enhancement to support this functionality.



## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Proposed Mechanism . . . . .	<a href="#">4</a>
<a href="#">2.1</a>	The SI Flag . . . . .	<a href="#">4</a>
<a href="#">2.2</a>	The NSID Option . . . . .	<a href="#">4</a>
<a href="#">3.</a>	Open Issues . . . . .	<a href="#">5</a>
<a href="#">3.1</a>	What Should the NSID Payload Be? . . . . .	<a href="#">5</a>
<a href="#">3.2</a>	Should Recursive Name Servers Respond to SI? . . . . .	<a href="#">6</a>
<a href="#">4.</a>	Acknowledgements . . . . .	<a href="#">8</a>
<a href="#">5.</a>	References . . . . .	<a href="#">9</a>
<a href="#">5.1</a>	Normative References . . . . .	<a href="#">9</a>
<a href="#">5.2</a>	Informative References . . . . .	<a href="#">9</a>
	Author's Address . . . . .	<a href="#">9</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">10</a>



## **1. Introduction**

With the increased use of DNS anycast, load balancing, and other mechanisms allowing more than one DNS name server to share a single IP address, it is sometimes difficult to tell which of a pool of name servers has answered a particular query.

Existing ad-hoc mechanisms such as those described in [\[I-D.ietf-dnsop-serverid\]](#) allow an operator to send follow-up queries when it is necessary to debug such a configuration, but there are situations in which this is not a totally satisfactory solution, since anycast routing may have changed, or the server pool in question may be behind some kind of extremely dynamic load balancing hardware. Thus, while these ad-hoc mechanisms are certainly better than nothing (and have the advantage of already being deployed), a better solution seems desirable.

Given that a DNS query is an idempotent operation with no retained state, it would appear that the only completely reliable way to obtain the identity of the name server which actually responded to a particular query is to have that name server include identifying information in the response itself. This note proposes a protocol enhancement to achieve this.



## **2. Proposed Mechanism**

This note proposes using an EDNS [[RFC2671](#)] flag bit to signal the resolver's desire for information identifying the name server, and an EDNS option to hold the name server's response (should it chose to honor the resolver's request).

### **2.1 The SI Flag**

A resolver signals its desire for information identifying the server by setting the SI (Send Identification) flag in the extended flags field of the OPT pseudo-RR.

The value of the SI flag is [TBD].

The semantics of the SI flag are not transitive. That is: the SI flag is a request that the name server which receives the query identify itself; in a so-called forwarding setup, the first hop name server is the one that should identify itself. If the resolver side of a forwarding name server wishes to receive identifying information, it is free to set the SI flag in its own queries, but that is a separate matter.

A name server which understands the SI flag should echo its value back in the response message, regardless of whether the name server chose to honor the request.

### **2.2 The NSID Option**

A name server which understands the SI flag and chooses to honor it responds by including identifying information in a NSID option in an EDNS OPT pseudo-RR in the response message.

The OPTION-CODE for the NSID option is [TBD].

The precise format of the identifying information is still an open issue at this point, and is discussed further in [Section 3.1](#).





### **3. Open Issues**

There are a couple of open issues in this proposal which would need to be settled before it could be used. The author has opinions on both of these and has stated those opinions below, but would appreciate feedback from the community.

#### **3.1 What Should the NSID Payload Be?**

There are several options for the payload of the NSID option.

- o It could be the "real" name of the specific name server within the name server pool.
- o It could be the "real" IP address (IPv4 or IPv6) of the name server within the name server pool.
- o It could be some sort of hash of the DNS name or IP address, perhaps including some kind of nonce.
- o It could be some sort of probabilistically unique identifier initially derived from some sort of random number generator then preserved across reboots of the name server.
- o It could be an arbitrary string of octets chosen at the discretion of the name server operator.

Each of these options has advantages and disadvantages.

- o Using the "real" name is simple, but assumes that the name server has a "real" name, which it may not.
- o Using the "real" address is also simple, and the name server almost certainly does have at least one non-anycast IP address for maintenance operations, but assumes that the operator of the name server is willing to divulge its non-anycast address, which might not be the case.
- o Using a hash (with or without a nonce) provides a fixed length value that the resolver can use to tell two name servers apart without necessarily being able to tell where either one of them "really" is, but makes debugging more difficult if one happens to be in a friendly open environment. Furthermore, a nonce may not add much value, since a hash based on an IPv4 address still only involves a 32-bit search space, and DNS names used for servers that operators might have to debug at 4am tend not to be very random at all.
- o Probabilistically unique identifiers have similar properties to hashed identifiers, but (given a sufficiently good random number generator) are immune to the search space issues. However, the strength of this approach is also its weakness: there is no algorithmic transformation by which even the server operator can associate name server instances with identifiers while debugging, which might be annoying. This approach also requires the name server instance to preserve the probabilistically unique identifier across reboots, but this does not appear to be a serious

Austein

Expires January 15, 2005

[Page 5]

restriction, since authoritative nameservers almost always have nonvolatile storage (such as a disk drive) in any case, and in rare cases where an authoritative name server does not have any way to store such an identifier, nothing terrible will happen if the name server just generates a new identifier every time it reboots.

- o Using an arbitrary octet string means that at least half of the name servers that support this option will probably end up identifying themselves as "My Name Server", which is not particularly useful.

Given that one common reason for using anycast DNS techniques is an attempt to harden a critical name server against denial of service attacks, the author believes that at least some name server operators are likely to want an identifier other than the "real" name or "real" address of the name server instance. Given all of the issues listed above, the best approach might be a combination of several of the above approaches. Tentative proposal:

- o Define the NSID payload to be an opaque byte string.
- o Operators for whom divulging the unicast address is an issue could use the raw binary representation of a probabilistically unique random number. This should probably be the default implementation behavior.
- o Operators for whom divulging the unicast address is not an issue could just use the raw binary representation of a unicast address for simplicity. This would only be done via an explicit configuration choice by the operator.
- o Operators who really need or want the ability to set the NSID payload to an arbitrary value could do so, but this would only be done via an explicit configuration choice by the operator.

This approach appears to provide enough information for useful debugging without unintentionally leaking the maintenance addresses of anycast name servers to nogoodniks, while also allowing name server operators who do not find such leakage threatening to provide more information at their own discretion.

This proposal begs the question of whether the NSID payload would also need to include a type octet indicating which of these three options the name server operator had chosen: the author suspects that such an octet would not be necessary, but this is another subject on which the author would welcome feedback.

### **3.2 Should Recursive Name Servers Respond to SI?**

Most of the discussion of name server identification to date has focused on identifying authoritative name servers, since the best known cases of anycast name servers are a subset of the name servers

Austein

Expires January 15, 2005

[Page 6]

for the root zone. However, given that anycast DNS techniques are equally applicable to recursive name servers as well as authoritative name servers, it may be useful for the name server side of a recursive name server to support this mechanism as well. The semantics proposed for the SI bit in [Section 2.1](#) are intended to support this model.

#### **4. Acknowledgements**

Steve Bellovin, Randy Bush, David Conrad, Mike Patton, Paul Vixie, Suzanne Woolf, and the law firm of Dewey, Chetham, and Howe.

## **5. References**

### **5.1 Normative References**

[RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", [RFC 2671](#), August 1999.

### **5.2 Informative References**

[I-D.ietf-dnsop-serverid]  
Conrad, D., "Identifying an Authoritative Name Server",  
[draft-ietf-dnsop-serverid-01](#) (work in progress), November  
2002.

#### Author's Address

Rob Austein  
ISC  
950 Charter Street  
Redwood City, CA 94063  
USA

EMail: [sra@isc.org](mailto:sra@isc.org)





## Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the IETF's procedures with respect to rights in IETF Documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

## Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

