DNSIND Working Group
INTERNET-DRAFT
<draft-dunlap-dns-duxfr-00.txt>

K. Dunlap
Check Point Software
P. Vixie
ISC
September 1999

Dynamic Update Zone Transfer

Status of This Memo

This draft, file name draft-dunlap-dns-duxfr-00.txt, is intended to become a Proposed Standard RFC. Distribution of this document is unlimited. Comments should be sent to <namedroppers@internic.net> or to the authors.

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026. 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 not appropriate 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

To view the list Internet-Draft Shadow Directories, see
http://www.ietf.org/shadow.html.

Abstract

This document proposes an alternative extension to the DNS protocol for Incremental zone transfer (IXFR) [RFC1995]. This extension uses the mechanisms for adding and deleting Resource Records specified in [RFC2136] to transmit the changes between authoritative servers of a zone.

1 - Introduction

For rapid propagation of changes to a DNS database [STD13], it is necessary to reduce latency by actively notifying servers of the change. This is accomplished by the DNS NOTIFY Mechanism [RFC1996].

The simple method described for Incremental transfer (IXFR), in [RFC1995], does not adequately address the complexity of the problem.

Dynamic Update Zone Transfer (DUXFR), as proposed, is a mechanism to transmit the complexity of changes in the zone and still have the efficiency of IXFR means to propagate changed portions of a zone.

In this document, a slave name server which requests DUXFR is called a DUXFR client and a master or slave name server which responds to the request is called a DUXFR server.

2 - Brief Description of the Protocol

If a DUXFR client, which likely has an older version of a zone, thinks it needs a newer version of the zone (typically through SOA refresh timeout or the NOTIFY mechanism), it sends a DUXFR message containing the SOA serial number of its (presumably outdated) copy of the zone.

A DUXFR server should keep record of the newest version of the zone and the differences between that copy and several older versions. When a DUXFR request with an older version number is received, the DUXFR server needs to send only the differences required to make that version current. These differences are sent using the DNS UPDATE format packets for deletes and add specified in [RFC2136 2.5].

When a zone has been updated, it should be saved in stable storage before the new version is used to respond to DUXFR (or AXFR) queries. Otherwise, if the server crashes, data which is no longer available may have been distributed to slave servers, which can cause persistent database inconsistencies.

If a DUXFR query with the same or newer version number than that of the server is received, it is replied to with a single SOA record of the server's current version, just as in IXFR.

The Transport protocol for DUXFR queries is TCP/IP.

3 - Query Format

The DUXFR Query format is based on the standard DNS UPDATE Message Format. In the DNS Packet Header the Opcode is set to UPDATE and the Zone Type (ZTYPE) being set to AXFR. The Additional section containing the SOA record of the client's version of the zone.

4 - Response Format

The response packets to the DUXFR query are in the standard DNS UPDATE Message Format. The records in the Update Section are formatted using the four sets of semantics for adding and deleting Resource Records specified in the ``Update Section'' in [RFC2136 2.5]. The client will process these changes using the prerequisite for the transaction as the existence of the SOA serial number specified in the Additional section of the DUXFR query.

The response to a DUXFR query, when the server no longer has all the previous history from the version the client requests, will be a Response code (RCODE) of "Refused". It is recommended that the client retry with an AXFR query described in [RFC1034 4.3.5].

It is recommended that the Prerequisite sections of the DNS message be empty on transmission and ignored on reception. The Additional section may contain necessary data such as signatures as specified by other extensions to [RFC 2136].

5 - Version Overhead

A DUXFR server can not be required to hold all previous versions forever and may delete them anytime. In general, there is a trade-off between the size of storage space and the possibility of using DUXFR.

Information about older versions should be purged if the total length of a DUXFR response would be longer than that of an AXFR response. Given that the purpose of DUXFR is to reduce AXFR overhead, this strategy is quite reasonable. The strategy assures that the amount of storage required is at most twice that of the current zone information.

Information older than the SOA expire period may also be purged.

6 - IANA Considerations

No IANA services are required by this document.

7 - Security Considerations

DNS is related to several security problems, no attempt is made to fix them in this document.

The authors believe that this document does not introduce any additional security problems to the current DNS protocol.

8 - Examples

Given the following three generations of data with the current serial number of 3.

```
        Example.Com.           IN  SOA NS.Example.Com. admin.Example.Com.
(
                                       1 600 600 3600000 604800 )
                               IN  NS  NS.Example.Com.
        NS.Example.Com.        IN  A   192.168.1.5
        Vangogh.Example.Com.   IN  A   192.168.1.21
```

Vangogh.Example.Com. is removed and Monet.Example.Com. is added.

```
        Example.Com.         IN  SOA NS.Example.Com. admin.Example.Com. (
                                     2 600 600 3600000 604800 )
                             IN  NS  NS.Example.Com.
        NS.Example.Com.      IN  A   192.168.1.5
        Monet.Example.Com.   IN  A   192.168.6.27
                             IN  A   192.168.3.128
```

One of the IP address of Monet.Example.Com. is changed.

```
        Example.Com.         IN  SOA NS.Example.Com. admin.Example.Com. (
                                     3 600 600 3600000 604800 )
                             IN  NS  NS.Example.Com.
        NS.Example.Com.      IN  A   192.168.1.5
        Monet.Example.Com.   IN  A   192.168.6.42
                             IN  A   192.168.3.128
```

The following DUXFR query:

```
                +-------------------------------------------------+
        Header  | OPCODE=QUERY, QR=Request                        |
                +-------------------------------------------------+
        Zone    | QNAME=Example.Com., QCLASS=IN, QTYPE=AXFR       |
                +-------------------------------------------------+
        Prerequisite | <empty>                                    |
                +-------------------------------------------------+
        Update  | <empty>                                         |
                +-------------------------------------------------+
        Additional | Example.Com.         IN  SOA  serial=1       |
                +-------------------------------------------------+
```

The reply could be with the following DUXFR response with Update packets in the Answer Section:

```
                +-------------------------------------------------+
        Header  | OPCODE=QUERY, QR=Response                       |
                +-------------------------------------------------+
        Zone    | QNAME=Example.Com., QCLASS=IN, QTYPE=AXFR       |
                +-------------------------------------------------+
        Prerequisite | Example.Com.         IN  SOA  serial=1      |
                +-------------------------------------------------+
        Update  | Vangogh.Example.Com. 0 ANY A    192.168.1.21    |
                | Monet.Example.Com.     IN  A    192.168.6.42    |
                | Monet.Example.Com.     IN  A    192.168.3.128   |
                | Example.Com.         0 IN  SOA  serial=1        |
                | Example.Com.           IN  SOA  serial=2        |
                | Monet.Example.Com.   0 ANY A    192.168.6.42    |
                | Example.Com.         0 ANY SOA  serial=2        |
                | Example.Com.           IN  SOA  serial=3        |
                +-------------------------------------------------+
        Additional | <empty>                                      |
                +-------------------------------------------------+
```

or with the following Compressed DUXFR response with Update packets in the Answer Section:

```
                +-------------------------------------------------+
        Header  | OPCODE=QUERY, QR=Response                       |
                +-------------------------------------------------+
        Zone    | QNAME=Example.Com., QCLASS=IN, QTYPE=AXFR       |
                +-------------------------------------------------+
        Prerequisite | Example.Com.         IN  SOA  serial=1     |
                +-------------------------------------------------+
        Update  | Vangogh.Example.Com. 0 ANY A    192.168.1.21    |
                | Monet.Example.Com.     IN  A    192.168.6.42    |
                | Monet.Example.Com.     IN  A    192.168.3.128   |
                | Example.Com.         0 ANY SOA  serial=1        |
                | Example.Com.           IN  SOA  serial=3        |
                +-------------------------------------------------+
        Additional | <empty>                                      |
                +-------------------------------------------------+
```

References

[RFC1034]]
P. Mockapetris, ``Domain Names - Concepts and Facilities'' STD 13, RFC 1034, USC/Information Sciences Institute, November 1987.

[RFC1035]
P. Mockapetris, ``Domain Names - Implementation and Specification'' RFC 1035, USC/Information Sciences Institute, November 1987.

[RFC1996]
P. Vixie, ``A Mechanism for Prompt Notification of Zone Changes (DNS Notify)'' RFC 1996, August 1996

[RFC1995]
M. Ohta, ``Incremental Zone Transfer in DNS'' RFC 1995, August 1996.

[RFC2026]
S. Bradner, ``the Internet Standards Process -- Revision 3'' RFC 2026, Harvard University, October 1996.

[RFC2136]
P. Vixie, S. Thomson, Y. Rekhter and J. Bound, ``Dynamic Updates in the Domain Name System (DNS UPDATE)'' RFC 2136, April 1997

Author's Address

Kevin J. Dunlap
Check Point Software Technologies, Inc.
The Meta IP Group
119 South Main Street, Suite 200
Seattle, WA 98033
+1 206 674 3700
<kevind@MetaIP.CheckPoint.Com>

Paul Vixie
Internet Software Consortium
950 Charter Street
Redwood City, CA 94063
+1 650 779 7001
<vixie@isc.org>

Full Copyright Statement