Internet Engineering Task Force (IETF)
Request for Comments: 5936
Updates: 1034, 1035
Category: Standards Track
ISSN: 2070-1721
June 2010

E. Lewis
NeuStar, Inc.
A. Hoenes, Ed.
TR-Sys

DNS Zone Transfer Protocol (AXFR)

Abstract

The standard means within the Domain Name System protocol for maintaining coherence among a zone's authoritative name servers consists of three mechanisms. Authoritative Transfer (AXFR) is one of the mechanisms and is defined in RFC 1034 and RFC 1035.

The definition of AXFR has proven insufficient in detail, thereby forcing implementations intended to be compliant to make assumptions, impeding interoperability. Yet today we have a satisfactory set of implementations that do interoperate. This document is a new definition of AXFR -- new in the sense that it records an accurate definition of an interoperable AXFR mechanism.

Status of This Memo

Table of Contents

1. Introduction

The Domain Name System standard facilities for maintaining coherent servers for a zone consist of three elements. Authoritative Transfer (AXFR) is defined in "Domain Names - Concepts and Facilities" [RFC1034] (referred to in this document as RFC 1034) and "Domain Names - Implementation and Specification" [RFC1035] (henceforth RFC 1035). Incremental Zone Transfer (IXFR) is defined in "Incremental Zone Transfer in DNS" [RFC1995]. A mechanism for prompt notification of zone changes (NOTIFY) is defined in "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)" [RFC1996]. The goal of these mechanisms is to enable a set of DNS name servers to remain coherently authoritative for a given zone.

This document re-specifies the AXFR mechanism as it is deployed in the Internet at large, hopefully with the precision expected from modern Internet Standards, and thereby updates RFC 1034 and RFC 1035.

1.1. Definition of Terms

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [BCP14].

Use of "newer"/"new" and "older"/"old" DNS refers to implementations written after and prior to the publication of this document.

"General-purpose DNS implementation" refers to DNS software developed for widespread use. This includes resolvers and servers freely accessible as libraries and standalone processes. This also includes proprietary implementations used only in support of DNS service offerings.

"Turnkey DNS implementation" refers to custom-made, single-use implementations of DNS. Such implementations consist of software that employs the DNS protocol message format yet does not conform to the entire range of DNS functionality.

The terms "AXFR session", "AXFR server", and "AXFR client" will be introduced in the first paragraph of Section 2, after some more context has been established.

1.2. Scope

In general terms, authoritative name servers for a given zone can use various means to achieve coherency of the zone contents they serve. For example, there are DNS implementations that assemble answers from data stored in relational databases (as opposed to master files), relying on the database's non-DNS means to synchronize the database instances. Some of these non-DNS solutions interoperate in some fashion. However, AXFR, IXFR, and NOTIFY are the only protocol-defined in-band mechanisms to provide coherence of a set of name servers, and they are the only mechanisms specified by the IETF.

This document does not cover incoherent DNS situations. There are applications of the DNS in which servers for a zone are designed to be incoherent. For these configurations, a coherency mechanism as described here would be unsuitable.

A DNS implementation is not required to support AXFR, IXFR, and NOTIFY, but it should have some means for maintaining name server coherency. A general-purpose DNS implementation will likely support AXFR (and in the same vein IXFR and NOTIFY), but turnkey DNS implementations may exist without AXFR.

1.3. Context

Besides describing the mechanisms themselves, there is the context in which they operate to consider. In the initial specifications of AXFR (and IXFR and NOTIFY), little consideration was given to security and privacy issues. Since the original definition of AXFR, new opinions have appeared on the access to an entire zone's contents. In this document, the basic mechanisms will be discussed separately from the permission to use these mechanisms.

1.4. Coverage and Relationship to Original AXFR Specification

This document concentrates on just the definition of AXFR. Any effort to update the specification of the IXFR or NOTIFY mechanisms is left to different documents.

The original "specification" of the AXFR sub-protocol is scattered through RFC 1034 and RFC 1035. Section 2.2 of RFC 1035 (on page 5) depicts the scenario for which AXFR has been designed. Section 4.3.5 of RFC 1034 describes the zone synchronization strategies in general and rules for the invocation of a full zone transfer via AXFR; the fifth paragraph of that section contains a very short sketch of the AXFR protocol; Section 5.5 of RFC 2181 has corrected a significant flaw in that specification. Section 3.2.3 of RFC 1035 has assigned the code point for the AXFR QTYPE (see Section 2.1.2 below for more

details). Section 4.2 of RFC 1035 discusses how the DNS uses the transport layer and briefly explains why UDP transport is deemed inappropriate for AXFR; the last paragraph of Section 4.2.2 gives details regarding TCP connection management for AXFR. Finally, the second paragraph of Section 6.3 in RFC 1035 mandates server behavior when zone data changes occur during an ongoing zone transfer using AXFR.

This document will update the specification of AXFR. To this end, it fully specifies the record formats and processing rules for AXFR, largely expanding on paragraph 5 of Section 4.3.5 of RFC 1034, and it details the transport considerations for AXFR, thus amending Section 4.2.2 of RFC 1035. Furthermore, it discusses backward-compatibility issues and provides policy/management considerations, as well as specific security considerations for AXFR. The goal of this document is to define AXFR as it is understood by the DNS community to exist today.

2. AXFR Messages

An AXFR session consists of an AXFR query message and the sequence of AXFR response messages returned for it. In this document, the AXFR client is the sender of the AXFR query, and the AXFR server is the responder. (Use of terms such as master, slave, primary, and secondary are not important for defining AXFR.) The use of the word "session" without qualification refers to an AXFR session.

An important aspect to keep in mind is that the definition of AXFR is restricted to TCP [RFC0793] (see Section 4 for details). The design of the AXFR process has certain inherent features that are not easily ported to UDP [RFC0768].

The basic format of an AXFR message is the DNS message as defined in Section 4 ("MESSAGES") of RFC 1035 [RFC1035], updated by the following documents.

o The "Basic" DNS specification:

  - "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)" [RFC1996]

  - "Dynamic Updates in the Domain Name System (DNS UPDATE)" [RFC2136]

  - "Clarifications to the DNS Specification" [RFC2181]

- "Extension Mechanisms for DNS (EDNS0)" [RFC2671]

- "Secret Key Transaction Authentication for DNS (TSIG)" [RFC2845]

- "Secret Key Establishment for DNS (TKEY RR)" [RFC2930]

- "Obsoleting IQUERY" [RFC3425]

- "Handling of Unknown DNS Resource Record (RR) Types" [RFC3597]

- "HMAC SHA (Hashed Message Authentication Code, Secure Hash Algorithm) TSIG Algorithm Identifiers" [RFC4635]

- "Domain Name System (DNS) IANA Considerations" [RFC5395]

o Further additions related to the DNS Security Extensions (DNSSEC), defined in these base documents:

- "DNS Security Introduction and Requirements" [RFC4033]

- "Resource Records for the DNS Security Extensions" [RFC4034]

- "Protocol Modifications for the DNS Security Extensions" [RFC4035]

- "Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)" [RFC4509]

- "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence" [RFC5155]

- "Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC" [RFC5702]

- "Clarifications and Implementation Notes for DNSSECbis" [DNSSEC-U]

These documents contain information about the syntax and semantics of DNS messages. They do not interfere with AXFR but are also helpful in understanding what will be carried via AXFR.

For convenience, the synopsis of the DNS message header from [RFC5395] (and the IANA registry for DNS Parameters [DNSVALS]) is reproduced here informally:

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                      ID                       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|QR|   OpCode  |AA|TC|RD|RA| Z|AD|CD|   RCODE   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                QDCOUNT/ZOCOUNT                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                ANCOUNT/PRCOUNT                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                NSCOUNT/UPCOUNT                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    ARCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

This document makes use of the field names as they appear in this diagram. The names of sections in the body of DNS messages are capitalized in this document for clarity, e.g., "Additional section".

The DNS message size limit from [RFC1035] for DNS over UDP (and its extension via the EDNS0 mechanism specified in [RFC2671]) is not relevant for AXFR, as explained in Section 4. The upper limit on the permissible size of a DNS message over TCP is only restricted by the TCP framing defined in Section 4.2.2 of RFC 1035, which specifies a two-octet message length field, understood to be unsigned, and thus causing a limit of 65535 octets. This limit is not changed by EDNS0.

Note that the TC (truncation) bit is never set by an AXFR server nor considered/read by an AXFR client.

2.1. AXFR Query

An AXFR query is sent by a client whenever there is a reason to ask. This might be because of scheduled or triggered zone maintenance activities (see Section 4.3.5 of RFC 1034 and DNS NOTIFY [RFC1996], respectively) or as a result of a command line request, say for debugging.

2.1.1. Header Values

These are the DNS message header values for an AXFR query.

ID	Selected by client; see Note a)
QR	MUST be 0 (Query)
OPCODE	MUST be 0 (Standard Query)

```
      Flags:
         AA       "n/a" -- see Note b)
         TC       "n/a" -- see Note b)
         RD       "n/a" -- see Note b)
         RA       "n/a" -- see Note b)
         Z        "mbz" -- see Note c)
         AD       "n/a" -- see Note b)
         CD       "n/a" -- see Note b)

      RCODE       MUST be 0 (No error)

      QDCOUNT     Number of entries in Question section;   MUST be 1

      ANCOUNT     Number of entries in Answer section;     MUST be 0

      NSCOUNT     Number of entries in Authority section;  MUST be 0

      ARCOUNT     Number of entries in Additional section -- see Note
d)
```

Notes:

a) Set to any value that the client is not already using with the same server. There is no specific means for selecting the value in this field. (Recall that AXFR is done only via TCP connections -- see Section 4, "Transport".)

A server MUST reply using messages that use the same message ID to allow a client to have multiple queries outstanding concurrently over the same TCP connection -- see Note a) in Section 2.2.1 for more details.

b) "n/a" -- The value in this field has no meaning in the context of AXFR query messages. For the client, it is RECOMMENDED that the value be zero. The server MUST ignore this value.

c) "mbz" -- The client MUST set this bit to 0; the server MUST ignore it.

d) The client MUST set this field to the number of resource records it places into the Additional section. In the absence of explicit specification of new RRs to be carried in the Additional section of AXFR queries, the value MAY be 0, 1, or 2. See Section 2.1.5, "Additional Section", for details on the currently applicable RRs.

2.1.2. Question Section

The Question section of the AXFR query MUST conform to Section 4.1.2 of RFC 1035, and contain a single resource record with the following values:

QNAME the name of the zone requested

QTYPE AXFR (= 252), the pseudo-RR type for zone transfer [DNSVALS]

QCLASS the class of the zone requested [DNSVALS]

2.1.3. Answer Section

The Answer section MUST be empty.

2.1.4. Authority Section

The Authority section MUST be empty.

2.1.5. Additional Section

Currently, two kinds of resource records are defined that can appear in the Additional section of AXFR queries and responses: EDNS and DNS transaction security. Future specifications defining RRs that can be carried in the Additional section of normal DNS transactions need to explicitly describe their use with AXFR, should that be desired.

The client MAY include one OPT resource record [RFC2671]. If the server does not support EDNS0, the client MUST send this section without an OPT resource record if there is a retry. However, the protocol does not define an explicit indication that the server does not support EDNS0; that needs to be inferred by the client. Often, the server will return a FormErr(1) that might be related to the OPT resource record. Note that, at the time of this writing, only the EXTENDED-RCODE field of the OPT RR is meaningful in the context of AXFR; future specifications of EDNS flags and/or EDNS options must describe their usage in the context of AXFR, if applicable.

The client MAY include one transaction integrity and authentication resource record, currently a choice of TSIG [RFC2845] or SIG(0) [RFC2931]. If the server has indicated that it does not recognize the resource record, and that the error is indeed caused by the resource record, the client probably should not try again. Removing the security data in the face of an obstacle ought to only be done with full awareness of the implication of doing so.

In general, if an AXFR client is aware that an AXFR server does not support a particular mechanism, the client SHOULD NOT attempt to engage the server using the mechanism (or engage the server at all). A client could become aware of a server's abilities via a configuration setting or via some other (as yet) undefined means.

The range of permissible resource records that MAY appear in the Additional section might change over time. If either a change to an existing resource record (like the OPT RR for EDNS) is made or a new Additional section record is created, the new definitions ought to include a discussion on the applicability and impact upon AXFR. Future resource records residing in the Additional section might have an effect that is orthogonal to AXFR, and so can ride through the session as opaque data. In this case, a "wise" implementation ought to be able to pass these records through without disruption.

2.2. AXFR Response

The AXFR response will consist of one or more messages. The special case of a server closing the TCP connection without sending an AXFR response is covered in Section 2.3.

An AXFR response that is transferring the zone's contents will consist of a series (which could be a series of length 1) of DNS messages. In such a series, the first message MUST begin with the SOA resource record of the zone, and the last message MUST conclude with the same SOA resource record. Intermediate messages MUST NOT contain the SOA resource record. The AXFR server MUST copy the Question section from the corresponding AXFR query message into the first response message's Question section. For subsequent messages, it MAY do the same or leave the Question section empty.

The AXFR protocol treats the zone contents as an unordered collection (or to use the mathematical term, a "set") of RRs. Except for the requirement that the transfer must begin and end with the SOA RR, there is no requirement to send the RRs in any particular order or grouped into response messages in any particular way. Although servers typically do attempt to send related RRs (such as the RRs forming an RRset, and the RRsets of a name) as a contiguous group or, when message space allows, in the same response message, they are not required to do so, and clients MUST accept any ordering and grouping of the non-SOA RRs. Each RR SHOULD be transmitted only once, and AXFR clients MUST ignore any duplicate RRs received.

Each AXFR response message SHOULD contain a sufficient number of RRs to reasonably amortize the per-message overhead, up to the largest number that will fit within a DNS message (taking the required content of the other sections into account, as described below).

Some old AXFR clients expect each response message to contain only a single RR. To interoperate with such clients, the server MAY restrict response messages to a single RR. As there is no standard way to automatically detect such clients, this typically requires manual configuration at the server.

To indicate an error in an AXFR response, the AXFR server sends a single DNS message when the error condition is detected, with the response code set to the appropriate value for the condition encountered. Such a message terminates the AXFR session; it MUST contain a copy of the Question section from the AXFR query in its Question section, but the inclusion of the terminating SOA resource record is not necessary.

An AXFR server may send a number of AXFR response messages free of an error condition before it sends the message indicating an error.

2.2.1. Header Values

These are the DNS message header values for AXFR responses.

```
   ID          MUST be copied from request -- see Note a)

   QR          MUST be 1 (Response)

   OPCODE      MUST be 0 (Standard Query)

   Flags:
      AA       normally 1 -- see Note b)
      TC       MUST be 0 (Not truncated)
      RD       RECOMMENDED: copy request's value; MAY be set to 0
      RA       SHOULD be 0 -- see Note c)
      Z        "mbz" -- see Note d)
      AD       "mbz" -- see Note d)
      CD       "mbz" -- see Note d)

   RCODE       See Note e)

   QDCOUNT     MUST be 1 in the first message;
               MUST be 0 or 1 in all following messages;
               MUST be 1 if RCODE indicates an error

   ANCOUNT     See Note f)

   NSCOUNT     MUST be 0

   ARCOUNT     See Note g)
```

Notes:

a) Because some old implementations behave differently than is now desired, the requirement on this field is stated in detail. New DNS servers MUST set this field to the value of the AXFR query ID in each AXFR response message for the session. AXFR clients MUST be able to manage sessions resulting from the issuance of multiple outstanding queries, whether AXFR queries or other DNS queries. A client SHOULD discard responses that do not correspond (via the message ID) to any outstanding queries.

Unless the client is sure that the server will consistently set the ID field to the query's ID, the client is NOT RECOMMENDED to issue any other queries until the end of the zone transfer. A client MAY become aware of a server's abilities via a configuration setting.

b) If the RCODE is 0 (no error), then the AA bit MUST be 1. For any other value of RCODE, the AA bit MUST be set according to the rules for that error code. If in doubt, it is RECOMMENDED that it be set to 1. It is RECOMMENDED that the value be ignored by the AXFR client.

c) It is RECOMMENDED that the server set the value to 0; the client MUST ignore this value.

The server MAY set this value according to the local policy regarding recursive service, but doing so might confuse the interpretation of the response, as AXFR cannot be retrieved recursively. A client MAY note the server's policy regarding recursive service from this value, but SHOULD NOT conclude that the AXFR response was obtained recursively, even if the RD bit was 1 in the query.

d) "mbz" -- The server MUST set this bit to 0; the client MUST ignore it.

e) In the absence of an error, the server MUST set the value of this field to NoError(0). If a server is not authoritative for the queried zone, the server SHOULD set the value to NotAuth(9). (Reminder: Consult the appropriate IANA registry [DNSVALS].) If a client receives any other value in response, it MUST act according to the error. For example, a malformed AXFR query or the presence of an OPT resource record sent to an old server will result in a

FormErr(1) value. This value is not set as part of the AXFR-specific response processing. The same is true for other values indicating an error.

f) The count of answer records MUST equal the number of resource records in the AXFR Answer section. When a server is aware that a client will only accept response messages with a single resource record, then the value MUST be 1. A server MAY be made aware of a client's limitations via configuration data.

g) The server MUST set this field to the number of resource records it places into the Additional section. In the absence of explicit specification of new RRs to be carried in the Additional section of AXFR response messages, the value MAY be 0, 1, or 2. See Section 2.1.5 above for details on the currently applicable RRs and Section 2.2.5 for additional considerations specific to AXFR servers.

2.2.2. Question Section

In the first response message, this section MUST be copied from the query. In subsequent messages, this section MAY be copied from the query, or it MAY be empty. However, in an error response message (see Section 2.2), this section MUST be copied as well. The content of this section MAY be used to determine the context of the message, that is, the name of the zone being transferred.

2.2.3. Answer Section

The Answer section MUST be populated with the zone contents. See Section 3 below on encoding zone contents.

2.2.4. Authority Section

The Authority section MUST be empty.

2.2.5. Additional Section

The contents of this section MUST follow the guidelines for the OPT, TSIG, and SIG(0) RRs, or whatever other future record is possible here. The contents of Section 2.1.5 apply analogously as well.

The following considerations specifically apply to AXFR responses:

If the client has supplied an EDNS OPT RR in the AXFR query and if the server supports EDNS as well, it SHOULD include one OPT RR in the first response message and MAY do so in subsequent response messages (see Section 2.2); the specifications of EDNS options to be carried in the OPT RR may impose stronger requirements.

If the client has supplied a transaction security resource record (currently a choice of TSIG and SIG(0)) and the server supports the method chosen by the client, it MUST place the corresponding resource record into the AXFR response message(s), according to the rules specified for that method.

2.3. TCP Connection Aborts

If an AXFR client sends a query on a TCP connection and the connection is closed at any point, the AXFR client MUST consider the AXFR session terminated. The message ID MAY be used again on a new connection, even if the question and AXFR server are the same.

Facing a dropped connection, a client SHOULD try to make some determination as to whether the connection closure was the result of network activity or due to a decision by the AXFR server. This determination is not an exact science. It is up to the AXFR client to react, but the implemented reaction SHOULD NOT be either an endless cycle of retries or an increasing (in frequency) retry rate.

An AXFR server implementer should take into consideration the dilemma described above when a connection is closed with an outstanding query in the pipeline. For this reason, a server ought to reserve this course of action for situations in which it believes beyond a doubt that the AXFR client is attempting abusive behavior.

3. Zone Contents

The objective of the AXFR session is to request and transfer the contents of a zone, in order to permit the AXFR client to faithfully reconstruct the zone as it exists at the primary server for the given zone serial number. The word "exists" here designates the externally visible behavior, i.e., the zone content that is being served (handed out to clients) -- not its persistent representation in a zone file or database used by the server -- and that for consistency should be served subsequently by the AXFR client in an identical manner.

Over time the definition of a zone has evolved from denoting a static set of records to also cover a dynamically updated set of records, and then a potentially continually regenerated set of records (e.g., RRs synthesized "on the fly" from rule sets or database lookup results in other forms than RR format) as well.

3.1. Records to Include

In the Answer section of AXFR response messages, the resource records within a zone for the given serial number MUST appear. The definition of what belongs in a zone is described in RFC 1034, Section 4.2, "How the database is divided into zones" (in particular Section 4.2.1, "Technical considerations"), and it has been clarified in Section 6 of RFC 2181.

Zones for which it is impractical to list the entire zone for a serial number are not suitable for AXFR retrieval. A typical (but not limiting) description of such a zone is a zone consisting of responses generated via other database lookups and/or computed based upon ever-changing data.

3.2. Delegation Records

In Section 4.2.1 of RFC 1034, this text appears (keep in mind that the "should" in the quotation predates [BCP14], cf. Section 1.1):

> The RRs that describe cuts ... should be exactly the same as the corresponding RRs in the top node of the subzone.

There has been some controversy over this statement and the impact on which NS resource records are included in a zone transfer.

The phrase "that describe cuts" is a reference to the NS set and applicable glue records. It does not mean that the cut point and apex resource records are identical. For example, the SOA resource record is only found at the apex. The discussion here is restricted to just the NS resource record set and glue, as these "describe cuts".

DNSSEC resource records have special specifications regarding their occurrence at a zone cut and the apex of a zone. This was first described in Sections 5.3 ff. and 6.2 of RFC 2181 (for the initial specification of DNSSEC), which parts of RFC 2181 now in fact are historical. The current DNSSEC core document set (see second bullet in Section 2 above) gives the full details for DNSSEC(bis) resource record placement, and Section 3.1.5 of RFC 4035 normatively specifies their treatment during AXFR; the alternate NSEC3 resource record defined later in RFC 5155 behaves identically to the NSEC RR, for the purpose of AXFR.

Informally:

- o The DS RRSet only occurs at the parental side of a zone cut and is authoritative data in the parent zone, not the secure child zone.

- o The DNSKEY RRSet only occurs at the apex of a signed zone and is part of the authoritative data of the zone it serves.

o Independent RRSIG RRSets occur at the signed parent side of a zone cut and at the apex of a signed zone; they are authoritative data in the respective zone; simple queries for RRSIG resource records may return both RRSets at once if the same server is authoritative for the parent zone and the child zone (Section 3.1.5 of RFC 4035 describes how to distinguish these RRs); this seeming ambiguity does not occur for AXFR, since each such RRSIG RRset belongs to a single zone.

o Different NSEC [RFC4034] (or NSEC3 [RFC5155]) resource records equally may occur at the parental side of a zone cut and at the apex of a zone; each such resource record belongs to exactly one of these zones and is to be included in the AXFR of that zone.

One issue is that in operations there are times when the NS resource records for a zone might be different at a cut point in the parent and at the apex of a zone. Sometimes this is the result of an error, and sometimes it is part of an ongoing change in name servers. The DNS protocol is robust enough to overcome inconsistencies up to (but not including) there being no parent-indicated NS resource record referencing a server that is able to serve the child zone. This robustness is one quality that has fueled the success of the DNS. Still, the inconsistency is an error state, and steps need to be taken to make it apparent (if it is unplanned).

Another issue is that the AXFR server could be authoritative for a different set of zones than the AXFR client. It is possible that the AXFR server be authoritative for both halves of an inconsistent cut point and that the AXFR client is authoritative for just the parent side of the cut point.

When facing a situation in which a cut point's NS resource records do not match the authoritative set, the question arises whether an AXFR server responds with the NS resource record set that is in the zone being transferred or the one that is at the authoritative location.

The AXFR response MUST contain the cut point NS resource record set registered with the zone whether it agrees with the authoritative set or not. "Registered with" can be widely interpreted to include data residing in the zone file of the zone for the particular serial number (in zone file environments) or as any data configured to be in the zone (database), statically or dynamically.

The reasons for this requirement are:

1) The AXFR server might not be able to determine that there is an inconsistency given local data; hence, requiring consistency would mean a lot more needed work and even network retrieval of data. An authoritative server ought not be required to perform any queries.

2) By transferring the inconsistent NS resource records from a server that is authoritative for both the cut point and the apex to a client that is not authoritative for both, the error is exposed. For example, an authorized administrator can manually request the AXFR and inspect the results to see the inconsistent records. (A server authoritative for both halves would otherwise always answer from the more authoritative set, concealing the error.)

3) The inconsistent NS resource record set might indicate a problem in a registration database.

4) This requirement is necessary to ensure that retrieving a given (zone, serial) pair by AXFR yields the exact same set of resource records, no matter which of the zone's authoritative servers is chosen as the source of the transfer.

If an AXFR server were allowed to respond with the authoritative NS RRset of a child zone instead of a parent-side NS RRset in the zone being transferred, the set of records returned could vary depending on whether or not the server happened to be authoritative for the child zone as well.

The property that a given (zone, serial) pair corresponds to a single, well-defined set of records is necessary for the correct operation of incremental transfer protocols such as IXFR [RFC1995]. For example, a client may retrieve a zone by AXFR from one server, and then apply an incremental change obtained by IXFR from a different server. If the two servers have different ideas of the zone contents, the client can end up attempting to incrementally add records that already exist or to delete records that do not exist.

3.3. Glue Records

As quoted in the previous section, Section 4.2.1 of RFC 1034 provides guidance and rationale for the inclusion of glue records as part of an AXFR response. And, as also argued in the previous section of this document, even when there is an inconsistency between the address in a glue record and the authoritative copy of the name server's address, the glue resource record that is registered as part of the zone for that serial number is to be included.

This applies to glue records for any address family [IANA-AF].

The AXFR response MUST contain the appropriate glue records as registered with the zone. The interpretation of "registered with" in the previous section applies here. Inconsistent glue records are an operational matter.

3.4. Name Compression

Compression of names in DNS messages is described in RFC 1035, Section 4.1.4, "Message compression". The issue highlighted here relates to a comment made in RFC 1034, Section 3.1, "Name space specifications and terminology", which says:

> When you receive a domain name or label, you should preserve its case.

("Should" in the quote predates [BCP14].)

Since the primary objective of AXFR is to enable the client to serve the same zone content as the server, unlike such normal DNS responses that are expected to preserve the case in the query, the actual zone transfer needs to retain the case of the labels in the zone content. Hence, name compression in an AXFR message SHOULD be performed in a case-preserving manner, unlike how it is done for "normal" DNS responses. That is, although when comparing a domain name for matching, "a" equals "A", when comparing for the purposes of message compression for AXFR, "a" is not equal to "A". Note that this is not the usual definition of name comparison in the DNS protocol and represents a new understanding of the requirement on AXFR servers.

Rules governing name compression of RDATA in an AXFR message MUST abide by the specification in "Handling of Unknown DNS Resource Record (RR) Types" [RFC3597], specifically, Section 4 on "Domain Name Compression".

3.5. Occluded Names

Dynamic Update [RFC2136] operations, and in particular their interaction with DNAME [RFC2672], can have a side effect of occluding names in a zone. The addition of a delegation point via dynamic update will render all subordinate domain names to be in a limbo, still part of the zone but not available to the lookup process. The addition of a DNAME resource record has the same impact. The subordinate names are said to be "occluded".

Occluded names MUST be included in AXFR responses. An AXFR client MUST be able to identify and handle occluded names. The rationale for this action is based on a speedy recovery if the dynamic update operation was in error and is to be undone.

4. Transport

AXFR sessions are currently restricted to TCP by Section 4.3.5 of RFC 1034, which states:

> Because accuracy is essential, TCP or some other reliable protocol must be used for AXFR requests.

The restriction to TCP is also mentioned in Section 6.1.3.2 of "Requirements for Internet Hosts - Application and Support" [RFC1123].

The most common scenario is for an AXFR client to open a TCP connection to the AXFR server, send an AXFR query, receive the AXFR response, and then close the connection. But variations of that most simple scenario are legitimate and likely: in particular, sending a query for the zone's SOA resource record first over the same TCP connection, and reusing an existing TCP connection for other queries.

Therefore, the assumption that a TCP connection is dedicated to a single AXFR session is incorrect. This wrong assumption has led to implementation choices that prevent either multiple concurrent zone transfers or the use of an open connection for other queries.

Since the early days of the DNS, operators who have sets of name servers that are authoritative for a common set of zones have found it desirable to be able to have multiple concurrent zone transfers in progress; this way, a name server does not have to wait for one zone transfer to complete before the next can begin. RFC 1035 did not exclude this possibility, but legacy implementations failed to support this functionality efficiently, over a single TCP connection. The remaining presence of such legacy implementations makes it necessary that new general-purpose client implementations still provide options for graceful fallback to the old behavior in their support of concurrent DNS transactions and AXFR sessions on a single TCP connection.

4.1. TCP

In the original definition, there arguably is an implicit assumption (probably unintentional) that a TCP connection is used for one and only one AXFR session. This is evidenced in the lack of an explicit requirement to copy the Question section and/or the message ID into responses, no explicit ordering information within the AXFR response messages, and the lack of an explicit notice indicating that a zone transfer continues in the next message.

The guidance given below is intended to enable better performance of the AXFR exchange as well as provide guidelines on interactions with older software. Better performance includes being able to multiplex DNS message exchanges including zone transfer sessions. Guidelines for interacting with older software are generally applicable to new AXFR clients. In the reverse situation -- older AXFR client and newer AXFR server -- the server ought to operate within the specification for an older server.

4.1.1. AXFR Client TCP

An AXFR client MAY request a connection to an AXFR server for any reason. An AXFR client SHOULD close the connection when there is no apparent need to use the connection for some time period. The AXFR server ought not have to maintain idle connections; the burden of connection closure ought to be on the client. "Apparent need" for the connection is a judgment for the AXFR client and the DNS client. If the connection is used for multiple sessions, or if it is known that sessions will be coming, or if there is other query/response traffic anticipated or currently on the open connection, then there is "apparent need".

An AXFR client can cancel the delivery of a zone only by closing the connection. However, this action will also cancel all other outstanding activity using the connection. There is no other mechanism by which an AXFR response can be cancelled.

When a TCP connection is closed remotely (relative to the client), whether by the AXFR server or due to a network event, the AXFR client MUST cancel all outstanding sessions and non-AXFR transactions. Recovery from this situation is not straightforward. If the disruption was a spurious event, attempting to restart the connection would be proper. If the disruption was caused by a failure that proved to be persistent, the AXFR client would be wise not to spend too many resources trying to rebuild the connection. Finally, if the connection was dropped because of a policy at the AXFR server (as can be the case with older AXFR servers), the AXFR client would be wise not to retry the connection. Unfortunately, knowing which of the three cases above (momentary disruption, failure, policy) applies is not possible with certainty, and can only be assessed by heuristics. This exemplifies the general complications for clients in connection-oriented protocols not receiving meaningful error responses.

An AXFR client MAY use an already opened TCP connection to start an AXFR session. Using an existing open connection is RECOMMENDED over opening a new connection. (Non-AXFR session traffic can also use an open connection.) If in doing so the AXFR client realizes that the responses cannot be properly differentiated (lack of matching query IDs, for example) or the connection is terminated for a remote reason, then the AXFR client SHOULD NOT attempt to reuse an open connection with the specific AXFR server until the AXFR server is updated (which is, of course, not an event captured in the DNS protocol).

4.1.2. AXFR Server TCP

An AXFR server MUST be able to handle multiple AXFR sessions on a single TCP connection, as well as to handle other query/response transactions over it.

If a TCP connection is closed remotely, the AXFR server MUST cancel all AXFR sessions in place. No retry activity is necessary; that is initiated by the AXFR client.

Local policy MAY dictate that a TCP connection is to be closed. Such an action SHOULD be in reaction to limits such as those placed on the number of outstanding open connections. Closing a connection in response to a suspected security event SHOULD be done only in extreme cases, when the server is certain the action is warranted. An isolated request for a zone not on the AXFR server SHOULD receive a response with the appropriate response code and not see the connection broken.

4.2. UDP

With the addition of EDNS0 and applications that require many small zones, such as in web hosting and some ENUM scenarios, AXFR sessions on UDP would now seem desirable. However, there are still some aspects of AXFR sessions that are not easily translated to UDP.

Therefore, this document does not update RFC 1035 in this respect: AXFR sessions over UDP transport are not defined.

5. Authorization

A zone administrator has the option to restrict AXFR access to a zone. This was not envisioned in the original design of the DNS but has emerged as a requirement as the DNS has evolved. Restrictions on AXFR could be for various reasons including a desire (or in some instances, having a legal requirement) to keep the bulk version of the zone concealed or to prevent the servers from handling the load

incurred in serving AXFR. It has been argued that these reasons are questionable, but this document, driven by the desire to leverage the interoperable practice that has evolved since RFC 1035, acknowledges the factual requirement to provide mechanisms to restrict AXFR.

A DNS implementation SHOULD provide means to restrict AXFR sessions to specific clients.

An implementation SHOULD allow access to be granted to Internet Protocol addresses and ranges, regardless of whether a source address could be spoofed. Combining this with techniques such as Virtual Private Networks (VPNs) [RFC2764] or Virtual LANs has proven to be effective.

A general-purpose implementation is RECOMMENDED to implement access controls based upon "Secret Key Transaction Authentication for DNS (TSIG)" [RFC2845] and/or "DNS Request and Transaction Signatures (SIG(0)s)" [RFC2931].

A general-purpose implementation SHOULD allow access to be open to all AXFR requests. That is, an operator ought to be able to allow any AXFR query to be granted.

A general-purpose implementation SHOULD NOT have a default policy for AXFR requests to be "open to all". For example, a default could be to restrict transfers to addresses selected by the DNS administrator(s) for zones on the server.

6. Zone Integrity

An AXFR client MUST ensure that only a successfully transferred copy of the zone data can be used to serve this zone. Previous description and implementation practice has introduced a two-stage model of the whole zone synchronization procedure: Upon a trigger event (e.g., when polling of a SOA resource record detects a change in the SOA serial number, or when a DNS NOTIFY request [RFC1996] is received), the AXFR session is initiated, whereby the zone data are saved in a zone file or database (this latter step is necessary anyway to ensure proper restart of the server); upon successful completion of the AXFR operation and some sanity checks, this data set is "loaded" and made available for serving the zone in an atomic operation, and flagged "valid" for use during the next restart of the DNS server; if any error is detected, this data set MUST be deleted, and the AXFR client MUST continue to serve the previous version of the zone, if it did before. The externally visible behavior of an AXFR client implementation MUST be equivalent to that of this two-stage model.

If an AXFR client rejects data obtained in an AXFR session, it SHOULD remember the serial number and MAY attempt to retrieve the same zone version again. The reason the same retrieval could make sense is that the reason for the rejection could be rooted in an implementation detail of one AXFR server used for the zone and not present in another AXFR server used for the zone.

Ensuring that an AXFR client does not accept a forged copy of a zone is important to the security of a zone. If a zone operator has the opportunity, protection can be afforded via dedicated links, physical or virtual via a VPN among the authoritative servers. But there are instances in which zone operators have no choice but to run AXFR sessions over the global public Internet.

Besides best attempts at securing TCP connections, DNS implementations SHOULD provide means to make use of "Secret Key Transaction Authentication for DNS (TSIG)" [RFC2845] and/or "DNS Request and Transaction Signatures (SIG(0)s)" [RFC2931] to allow AXFR clients to verify the contents. These techniques MAY also be used for authorization.

7. Backwards Compatibility

Describing backwards compatibility is difficult because of the lack of specifics in the original definition. In this section, some hints at building in backwards compatibility are given, mostly repeated from the relevant earlier sections.

Backwards compatibility is not necessary, but the greater the extent of an implementation's compatibility, the greater its interoperability. For turnkey implementations, this is not usually a concern. For general-purpose implementations, this takes on varying levels of importance, depending on the implementer's desire to maintain interoperability.

It is unfortunate that a need to fall back to older behavior cannot be discovered, and thus has to be noted in a configuration file. An implementation SHOULD, in its documentation, encourage operators to periodically review AXFR clients and servers it has made notes about repeatedly, as old software gets updated from time to time.

7.1. Server

An AXFR server has the luxury of being able to react to an AXFR client's abilities, with the exception of knowing whether the client can accept multiple resource records per AXFR response message. The knowledge that a client is so restricted cannot be discovered; hence, it has to be set by configuration.

An implementation of an AXFR server MAY permit configuring, on a per AXFR client basis, the necessity to revert to a single resource record per message; in that case, the default SHOULD be to use multiple records per message.

7.2. Client

An AXFR client has the opportunity to try other features (i.e., those not defined by this document) when querying an AXFR server.

Attempting to issue multiple DNS queries over a TCP transport for an AXFR session SHOULD be aborted if it interrupts the original request, and SHOULD take into consideration whether the AXFR server intends to close the connection immediately upon completion of the original (connection-causing) zone transfer.

8. Security Considerations

This document is a clarification of a mechanism outlined in RFCs 1034 and 1035 and as such does not add any new security considerations. RFC 3833 [RFC3833] is devoted entirely to security considerations for the DNS; its Section 4.3 delineates zone transfer security aspects from the security threats addressed by DNSSEC.

Concerns regarding authorization, traffic flooding, and message integrity are mentioned in "Authorization" (Section 5), "TCP" (Section 4.1), and "Zone Integrity" (Section 6).

9. IANA Considerations

IANA has added a reference to this RFC in the AXFR (252) row of the "Resource Record (RR) TYPEs" subregistry of the "Domain Name System (DNS) Parameters" registry.

10. Internationalization Considerations

The AXFR protocol is transparent to the parts of DNS zone content that can possibly be subject to Internationalization considerations. It is assumed that for DNS labels and domain names, the issue has been solved via "Internationalizing Domain Names in Applications (IDNA)" [RFC3490] or its successor(s).

11. Acknowledgments

Earlier draft versions of this document have been edited by Andreas Gustafsson. In his latest draft version, this acknowledgment appeared:

Many people have contributed input and commentary to earlier versions of this document, including but not limited to Bob Halley, Dan Bernstein, Eric A. Hall, Josh Littlefield, Kevin Darcy, Robert Elz, Levon Esibov, Mark Andrews, Michael Patton, Peter Koch, Sam Trenholme, and Brian Wellington.

Comments on later draft versions have come from these individuals: Mark Andrews, Paul Vixie, Wouter Wijngaards, Iain Calder, Tony Finch, Ian Jackson, Andreas Gustafsson, Brian Wellington, Niall O'Reilly, Bill Manning, and other participants of the DNSEXT working group. Significant comments from the IETF at large have been received from Subramanian Moonesamy, Chris Lonvick, and Vijay K. Gurbani.

Edward Lewis served as a patiently listening sole document editor for two years.

12. References

All "RFC" references below -- like all RFCs -- and information about the RFC series can be obtained from the RFC Editor web site at http://www.rfc-editor.org.

12.1. Normative References

[BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1123] Braden, R., Ed., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

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

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

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

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.

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

[RFC2672] Crawford, M., "Non-Terminal DNS Name Redirection", RFC 2672, August 1999.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[RFC2930] Eastlake 3rd, D., "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, September 2000.

[RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.

[RFC3425] Lawrence, D., "Obsoleting IQUERY", RFC 3425, November 2002.

[RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, September 2003.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC4509] Hardaker, W., "Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)", RFC 4509, May 2006.

[RFC4635] Eastlake 3rd, D., "HMAC SHA (Hashed Message Authentication Code, Secure Hash Algorithm) TSIG Algorithm Identifiers", RFC 4635, August 2006.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, March 2008.

[RFC5395] Eastlake 3rd, D., "Domain Name System (DNS) IANA Considerations", BCP 42, RFC 5395, November 2008.

[RFC5702] Jansen, J., "Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC", RFC 5702, October 2009.

12.2. Informative References

[DNSVALS] IANA Registry "Domain Name System (DNS) Parameters", http://www.iana.org/.

[IANA-AF] IANA Registry "Address Family Numbers", http://www.iana.org/.

[RFC2764] Gleeson, B., Lin, A., Heinanen, J., Armitage, G., and A. Malis, "A Framework for IP Based Virtual Private Networks", RFC 2764, February 2000.

[RFC3490] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.

[RFC3833] Atkins, D. and R. Austein, "Threat Analysis of the Domain Name System (DNS)", RFC 3833, August 2004.

[DNSSEC-U] Weiler, S. and D. Blacka, "Clarifications and Implementation Notes for DNSSECbis", Work in Progress, March 2010.

Authors' Addresses

Edward Lewis
46000 Center Oak Plaza
Sterling, VA 20166
US

EMail: ed.lewis@neustar.biz

Alfred Hoenes, Editor
TR-Sys
Gerlinger Str. 12
Ditzingen D-71254
Germany

EMail: ah@TR-Sys.de