Network Working Group
Internet Draft
Expiration Date: July 1997

Robert Elz
University of Melbourne

Randy Bush
RGnet, Inc.

January 1997

Clarifications to the DNS Specification

draft-ietf-dnsind-clarify-03.txt

Status of this Memo

This document is an Internet-Draft. 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."

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

1. Abstract

Please do not bother with this draft, it was intended to be posted before the San Jose IETF, but missed the deadline by minutes. It was (other than this paragraph) later posted to the namedroppers mailing list. A new version, which responds to comments received from that mailing list posting will be posted to the I-D directories within days, please wait for that one, and don't waste time on this. This version is appearing solely to keep the I-D numbering sequence sane, the mailing list version was called -03, so we must have a -03 so the next real version can be -04 ...

This draft considers some areas that have been identified as problems with the specification of the Domain Name System, and proposes remedies for the defects identified. Five separate issues are considered:

+ IP packet header address usage from multi-homed servers,
+ TTLs in sets of records with the same name, class, and type,
+ correct handling of zone cuts,
+ the issue of what is an authoritative, or canonical, name,
+ and the issue of what makes a valid DNS label.

The first three of these are areas where the correct behaviour has been somewhat unclear, we seek to rectify that. The other two are already adequately specified, however the specifications seem to be sometimes ignored. We seek to reinforce the existing specification.

Contents

2. Introduction

Several problem areas in the Domain Name System specification [RFC1034, RFC1035] have been noted through the years [RFC1123]. This draft addresses several additional problem areas. The issues here are independent. Those issues are the question of which source address a multi-homed DNS server should use when replying to a query, the issue of differing TTLs for DNS records with the same label, class and type, and the issue of canonical names, what they are, how CNAME records relate, what names are legal in what parts of the DNS, and what is the valid syntax of a DNS name.

Suggestions for clarifications to the DNS specification to avoid these problems are made in this memo. The solutions proposed herein

are intended to stimulate discussion. It is possible that the sense of either may be reversed before the next iteration of this draft, but less likely now than it was before the previous version.

3. Server Reply Source Address Selection

Most, if not all, DNS clients, whether servers acting as clients for the purposes of recursive query resolution, or resolvers, expect the address from which a reply is received to be the same address as that to which the query eliciting the reply was sent. This, along with the identifier (ID) in the reply is used for disambiguating replies, and filtering spurious responses. This may, or may not, have been intended when the DNS was designed, but is now a fact of life.

Some multi-homed hosts running DNS servers fail to anticipate this usage, and consequently send replies from the "wrong" source address, causing the reply to be discarded by the client.

3.1. UDP Source Address Selection

To avoid these problems, servers when responding to queries using UDP must cause the reply to be sent with the source address field in the IP header set to the address that was in the destination address field of the IP header of the packet containing the query causing the response. If this would cause the response to be sent from an IP address which is not permitted for this purpose, then the response may be sent from any legal IP address allocated to the server. That address should be chosen to maximise the possibility that the client will be able to use it for further queries. Servers configured in such a way that not all their addresses are equally reachable from all potential clients need take particular care when responding to queries sent to anycast, multicast, or similar, addresses.

3.2. Port Number Selection

Replies to all queries must be directed to the port from which they were sent. With queries received via TCP this is an inherent part of the transport protocol, for queries received by UDP the server must take note of the source port and use that as the destination port in the response. Replies should always be sent from the port to which

they were directed. Except in extraordinary circumstances, this will be the well known port assigned for DNS queries [RFC1700].

4. Resource Record Sets

Each DNS Resource Record (RR) has a label, class, type, and data. While it is meaningless for two records to ever have label, class, type and data all equal (servers should suppress such duplicates if encountered), it is possible for many record types to exist with the same label class and type, but with different data. Such a group of records is hereby defined to be a Resource Record Set (RRSet).

4.1. Sending RRs from an RRSet

A query for a specific (or non-specific) label, class, and type, will always return all records in the associated RRSet - whether that be one or more RRs, or the response shall be marked as "truncated" if the entire RRSet will not fit in the response.

4.2. TTLs of RRs in an RRSet

Resource Records also have a time to live (TTL). It is possible for the RRs in an RRSet to have different TTLs, however no uses for this have been found which cannot be better accomplished in other ways. This can, however, cause partial replies (not marked "truncated") from a caching server, where the TTLs for some but not all of the RRs in the RRSet have expired.

Consequently the use of differing TTLs in an RRSet is hereby deprecated, the TTLs of all RRs in an RRSet must be the same.

Should a client receive a response containing RRs from an RRSet with differing TTLs, it should treat the RRs for all purposes as if all TTLs in the RRSet had been set to the value of the lowest TTL in the RRSet.

4.3. Receiving RRSets

Servers must never merge RRs from a response with RRs in their cache to form an RRSet. If a response contains data which would form an RRSet with data in a server's cache the server must either ignore the

RRs in the response, or use those to replace the existing RRSet in the cache, as appropriate. Consequently the issue of TTLs varying between the cache and a response does not cause concern, one will be ignored. That is, one of the data sets is always incorrect if the data from an answer differs from the data in the cache. The challenge for the server is to determine which of the data sets is correct, assuming that one is, and retain that, while ignoring the other. Note that if a server receives an answer containing an RRSet that is identical to that in its cache, with the possible exception of the TTL value, it may update the TTL in its cache with the TTL of the received answer. It should do this if the received answer would be considered more authoritative (as discussed in the next section) than the previously cached answer.

4.3.1. Ranking data

When considering whether to accept an RRSet in a reply, or retain an RRSet already in its cache instead, a server should consider the relative likely trustworthiness of the various data. That is, an authoritative answer from a reply should replace cached data that had been obtained from additional information in an earlier reply, but additional information from a reply will be ignored if the cache contains data from an authoritative answer or a zone file.

The accuracy of data available is assumed from its source. Trustworthiness shall be, in order from most to least:

+ Data from a primary zone file, other than glue data,
+ Data from a zone transfer, other than glue,
+ That from the answer section of an authoritative reply,
+ Glue from a primary zone, or glue from a zone transfer,
+ Data from the authority section of an authoritative answer,
+ Data from the answer section of a non-authoritative answer,
+ Additional information from an authoritative answer,
+ Data from the authority section of a non-authoritative answer,
+ Additional information from non-authoritative answers.

When DNS security [DNSSEC] is in use, and authenticated data has been received and verified, it shall be considered more trustworthy than unauthenticated data of the same type. Note that throughout this document, "authoritative" is used to mean a reply with the AA bit set. DNSSEC uses trusted chains of SIG and KEY records to determine what data is authenticated, the AA bit is almost irrelevant. However DNSSEC aware servers must still correctly set the AA bit in responses to enable correct operation with servers that are not security aware (almost all currently).

Note that, glue excluded, it is impossible for data from two primary zone files, two secondary zones (data from zone transfers) or data from primary and secondary zones to ever conflict. Where glue for the same name exists in multiple zones, and differs in value, the nameserver should select data from a primary zone file in preference to secondary, but otherwise may choose any single set of such data. Choosing that which appears to come from a source nearer the authoritative data source may make sense where that can be determined. Choosing primary data over secondary allows the source of incorrect glue data to be discovered more readily, when such data does exist.

"Glue" above includes any record in a zone file that is not properly part of that zone, including nameserver records of delegated sub-zones (NS records), address records that accompany those NS records

(A, AAAA, etc), and any other stray data that might appear.

4.4. Sending RRSets (reprise)

A Resource Record Set should only be included once in any DNS reply. It may occur in any of the Answer, Authority, or Additional Information sections, as required, however should not be repeated in the same, or any other, section, except where explicitly required by a specification. For example, an AXFR response requires the SOA record (always an RRSet containing a single RR) be both the first and last record of the reply. Where duplicates are required this way, the TTL transmitted in each case must be the same.

5. Zone Cuts

A "Zone" is a set of one, or usually, more, domains collected and treated as a unit. A "Zone Cut" is the division between one zone and another. A zone comprises some subset of the DNS tree, rooted at a domain known as the "origin" of the zone. The origin domain itself, and some, or all, of its sub-domains, form the zone. The existence of a zone cut is indicated by the presence, in the zone, of a NameServer (NS) record for any domain other than the origin of the zone.

5.1. Zone authority

The authoritative servers for a zone are listed in the NS records for the origin of the zone, which, along with a Start of Authority (SOA) record are the mandatory records in every zone. Such a server is authoritative for all resource records in a zone which are not in another zone. The NS records that indicate a zone cut are the property of the child zone created, as are any other records for the origin of that child zone, or any sub-domains of it. A server for the parent zone should not return authoritative answers for queries related to names in a child zone, which includes the NS records at the zone cut, unless it also happens to be a server for the child zone of course.

Other than the DNSSEC cases mentioned immediately below, servers should ignore data other than NS records, and necessary A records to locate the servers listed in the NS records, that may happen to be configured in a zone at a zone cut.

5.2. DNSSEC issues

The DNS security mechanisms [DNSSEC] complicate this somewhat, as some of the new resource record types added are very unusual when compared with other DNS RRs. In particular the NXT ("next") RR type contains information about which names exist in a zone, and hence which do not, and thus must necessarily relate to the zone in which it exists. In fact, the same domain name may have different NXT

records in the parent zone and the child zone, and both are valid, and are not an RRSet.

Since NXT records are intended to be automatically generated, rather than configured by DNS operators, servers may, but are not required to, retain all differing NXT records they receive regardless of the rules in section 4.3.

To indicate that a subzone is insecure, securely, that is, from a secure parent zone, DNSSEC requires that a KEY RR indicating that the subzone is insecure, and the parent zone's authenticating SIG RR(s) be present in the parent zone, as they by definition cannot be in the subzone. Where a subzone is secure, the KEY and SIG can be duplicated in both zone files, but should always be present in the subzone.

Note that in none of these cases should a server for the parent zone, not also being a server for the subzone, set the AA bit in any response for a label at a zone cut.

6. Naming issues

It has sometimes been inferred from some sections of the DNS specification [RFC1034, RFC1035] that a host, or perhaps an interface of a host, is permitted exactly one authoritative, or official, name, called the canonical name. There is no such requirement in the DNS.

6.1. CNAME records

The DNS CNAME ("canonical name") record exists to provide the canonical name associated with an alias name. There may be only one such canonical name for any one alias. That name should generally be a name that exists elsewhere in the DNS, though some applications for aliases with no accompanying canonical name exist. An alias name (label of a CNAME record) may, if DNSSEC is in use, have SIG, NXT, and KEY RRs, but may have no other data. That is, for any label in the DNS (any domain name) exactly one of the following is true:

+ one CNAME record exists, optionally accompanied by SIG, NXT, and

KEY RRs,
+ other records exist, possibly many records, none of them being CNAME records
+ the name does not exist at all.

If the canonical name associated with an alias does not exist, a lookup of the alias seeking anything but one of the CNAME, SIG, NXT, or KEY RR (or the pseudo-type ANY) should indicate that the name does not exist, just as if the alias itself did not exist. A CNAME (or ANY) type lookup should return the CNAME RR itself. Lookups for SIG, NXT or KEY records should return any such associated RR's that the alias may own (as would an ANY lookup).

6.1.1. CNAME terminology

It has been traditional to refer to the label of a CNAME record as "a CNAME". This is unfortunate, as "CNAME" is an abbreviation of "canonical name", and the label of a CNAME record is most certainly not a canonical name. It is, however, an entrenched usage, care must therefore be taken to be very clear whether the label, or the value (the canonical name) of a CNAME resource record is intended. In this document, the label of a CNAME resource record will always be referred to as an alias.

6.2. PTR records

Confusion about canonical names has lead to a belief that a PTR record should have exactly one RR in its RRSet. This is incorrect, the relevant section of RFC1034 (section 3.6.2) indicates that the value of a PTR record should be a canonical name. That is, it should not be an alias. There is no implication in that section that only one PTR record is permitted for a name, and no such restriction should be inferred.

6.3. MX and NS records

The domain name used as the value of a NS resource record, or part of the value of a MX resource record should not be an alias. Not only is the specification quite clear on this point, but using an alias in either of these positions neither works as well as might be hoped, nor well fulfills the ambition that may have led to this approach.

Searching for either NS or MX records causes "additional section processing" in which address records associated with the value of the record sought are appended to the answer. This helps avoid needless extra queries which are easily anticipated when the first was made.

Additional section processing does not include CNAME records, let alone the address records that may be associated with the canonical name derived from the alias. Thus, if an alias is used as the value of an NS or MX record, no address will be returned together with the NS or MX value. This can cause extra queries, and extra network burden, on every query, that could have been trivially avoided by resolving the alias and placing the canonical name directly in the affected record just once when it was updated or installed. In some particular hard cases the lack of the additional section address records in the results of a NS lookup can actually cause the request to fail.

7. Name syntax

Occasionally it is assumed that the Domain Name System serves only the purpose of mapping Internet host names to data, and mapping Internet addresses to host names. This is not correct, the DNS is a general (if somewhat limited) hierarchical database, and can store almost any kind of data, for almost any purpose.

The DNS itself places only one restriction upon the particular labels that can be used to identify resource records. That one restriction relates to the length of the label and the full name. Any one label is limited to 63 octets, and a full name is limited to 255 octets (including the separators). That restriction aside, any binary string whatever can be used as the label of any resource record, and as the value of one of the records that includes a domain name as some or all of its value (SOA, NS, MX, PTR, CNAME, SRV, and any others that may be added). Implementations of the DNS protocols must not place any restrictions on the labels that can be used.

Note however, that the various applications that make use of DNS data can have restrictions imposed upon what particular data is acceptable in their environment. For example, that any binary label can have an MX record does not imply that any binary name can be used as the host part of an e-mail address. Clients of the DNS can impose whatever restrictions are appropriate to their circumstances to the values they use as keys for DNS lookup requests, and to the values returned by the DNS.

See also [RFC1123] section 6.1.3.5.

8. Security Considerations

This document does not consider security.

In particular, nothing in section 3 is any way related to, or useful for, any security related purposes.

Section 4.3.1 is also not related to security. Security of DNS data will be obtained by the Secure DNS [DNSSEC], which is orthogonal to this memo.

It is not believed that anything in this document adds to any security issues that may exist with the DNS, nor does it do anything to lessen them.

9. References

[RFC1034] Domain Names - Concepts and Facilities, (STD 13)
P. Mockapetris, ISI, November 1987.

[RFC1035] Domain Names - Implementation and Specification (STD 13)
P. Mockapetris, ISI, November 1987

[RFC1123] Requirements for Internet hosts - application and support, (STD 3) R. Braden, January 1989

[RFC1700] Assigned Numbers (STD 2)
J. Reynolds, J. Postel, October 1994.

[DNSSEC] Domain Name System Security Extensions,
D. E. Eastlake, 3rd, C. W. Kaufman,
Work in Progress (soon to be an RFC), August 1996.

10. Acknowledgements

This memo arose from discussions in the DNSIND working group of the IETF in 1995 and 1996, the members of that working group are largely responsible for the ideas captured herein. Particular thanks to Donald E. Eastlake, 3rd, for assistance with the DNSSEC issues in

this document.

11. Authors' addresses

Robert Elz
Computer Science
University of Melbourne
Parkville, Victoria, 3052
Australia.

EMail: kre@munnari.OZ.AU

Randy Bush
RGnet, Inc.
10361 NE Sasquatch Lane
Bainbridge Island, Washington, 98110
United States.

EMail: randy@psg.com