IPng Working Group
Internet Draft

Matt Crawford
Fermilab
Christian Huitema
Susan Thomson
Telcordia
May 17, 2000

DNS Extensions to Support IPv6 Address Aggregation and Renumbering
<draft-ietf-ipngwg-dns-lookups-08.txt>

Status of this Memo

1. Abstract

This document defines changes to the Domain Name System to support renumberable and aggregatable IPv6 addressing. The changes include a new resource record type to store an IPv6 address in a manner which expedites network renumbering and updated definitions of existing query types that return Internet addresses as part of additional section processing.

For lookups keyed on IPv6 addresses (often called reverse lookups), this document defines a new zone structure which allows a zone to be used without modification for parallel copies of an address space (as for a multihomed provider or site) and across network renumbering events.

2. Introduction

Maintenance of address information in the DNS is one of several obstacles which have prevented site and provider renumbering from being feasible in IP version 4. Arguments about the importance of network renumbering for the preservation of a stable routing system and for other purposes may be read in documents cited here as [RENUM]. To support the storage of IPv6 addresses without impeding renumbering we define the following extensions.

- A new resource record type, "A6", is defined to map a domain name to an IPv6 address, with a provision for indirection for leading "prefix" bits.

- Existing queries that perform additional section processing to locate IPv4 addresses are redefined to do that processing for both IPv4 and IPv6 addresses.

- A new domain, IP6.ARPA, is defined to support lookups based on IPv6 address.

- A new prefix-delegation method is defined, relying on new DNS features [BITLBL, DNAME].

The changes are designed to be compatible with existing application programming interfaces. The existing support for IPv4 addresses is retained. Transition issues related to the coexistence of both IPv4 and IPv6 addresses in DNS are discussed in [TRANS].

This memo proposes a replacement for the specification in RFC 1886 and a departure from current implementation practices. The changes are designed to facilitate network renumbering and multihoming. Domains employing the A6 record for IPv6 addresses can insert automatically-generated AAAA records in zone files to ease transition. It is expected that after a reasonable period, RFC 1886 will become Historic.

The next three major sections of this document are an overview of the facilities defined or employed by this specification, the specification itself, and examples of use.

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 [KWORD]. The key word "SUGGESTED" signifies a strength between MAY and SHOULD: it is believed that compliance with the suggestion has tangible benefits in most instances.

3. Overview

This section provides an overview of the DNS facilities for storage of IPv6 addresses and for lookups based on IPv6 address, including those defined here and elsewhere.

3.1. Name-to-Address Lookup

IPv6 addresses are stored in one or more A6 resource records. A single A6 record may include a complete IPv6 address, or a contiguous portion of an address and information leading to one or more prefixes. Prefix information comprises a prefix length and a DNS name which is in turn the owner of one or more A6 records defining the prefix or prefixes which are needed to form one or more complete IPv6 addresses. When the prefix length is zero, no DNS name is present and all the leading bits of the address are significant. There may be multiples levels of indirection and the existence of multiple A6 records at any level multiplies the number of IPv6 addresses which are formed.

An application looking up an IPv6 address will generally cause the DNS resolver to access several A6 records, and multiple IPv6 addresses may be returned even if the queried name was the owner of only one A6 record. The authenticity of the returned address(es) cannot be directly verified by DNS Security [DNSSEC]. The A6 records which contributed to the address(es) may of course be verified if signed.

Implementers are reminded of the necessity to limit the amount of work a resolver will perform in response to a client request. This principle MUST be extended to also limit the generation of DNS requests in response to one name-to-address (or address-to-name) lookup request.

3.2. Underlying Mechanisms for Reverse Lookups

This section describes the new DNS features which this document exploits. This section is an overview, not a specification of those features. The reader is directed to the referenced documents for more details on each.

3.2.1. Delegation on Arbitrary Boundaries

This new scheme for reverse lookups relies on a new type of DNS label called the "bit-string label" [BITLBL]. This label compactly

represents an arbitrary string of bits which is treated as a hierarchical sequence of one-bit domain labels. Resource records can thereby be stored at arbitrary bit-boundaries.

Examples in section 6 will employ the following textual representation for bit-string labels, which is a subset of the syntax defined in [BITLBL]. A base indicator "x" for hexadecimal and a sequence of hexadecimal digits is enclosed between "\[" and "]". The bits denoted by the digits represent a sequence of one-bit domain labels ordered from most to least significant. (This is the opposite of the order they would appear if listed one bit at a time, but it appears to be a convenient notation.) The digit string may be followed by a slash ("/") and a decimal count. If omitted, the implicit count is equal to four times the number of hexadecimal digits.

Consecutive bit-string labels are equivalent (up to the limit imposed by the size of the bit count field) to a single bit-string label containing all the bits of the consecutive labels in the proper order. As an example, either of the following domain names could be used in a QCLASS=IN, QTYPE=PTR query to find the name of the node with IPv6 address 3ffe:7c0:40:9:a00:20ff:fe81:2b32.

\[x3FFE07C0004000090A0020FFFE812B32/128].IP6.ARPA.

\[x0A0020FFFE812B32/64].\[x0009/16].\[x3FFE07C00040/48].IP6.ARPA.

3.2.2. Reusable Zones

DNS address space delegation is implemented not by zone cuts and NS records, but by a new analogue to the CNAME record, called the DNAME resource record [DNAME]. The DNAME record provides alternate naming to an entire subtree of the domain name space, rather than to a single node. It causes some suffix of a queried name to be substituted with a name from the DNAME record's RDATA.

For example, a resolver or server providing recursion, while looking up a QNAME a.b.c.d.e.f may encounter a DNAME record

d.e.f. DNAME w.xy.

which will cause it to look for a.b.c.w.xy.

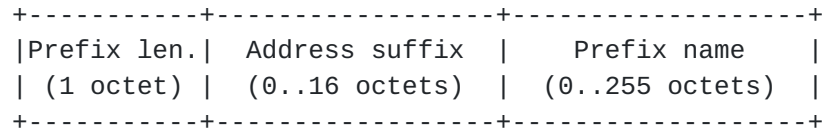
4. Specifications

4.1. The A6 Record Type

The A6 record type is specific to the IN (Internet) class and has type number 38 (decimal).

4.1.1. Format

The RDATA portion of the A6 record contains two or three fields.



- o A prefix length, encoded as an eight-bit unsigned integer with value between 0 and 128 inclusive.

- o An IPv6 address suffix, encoded in network order (high-order octet first). There MUST be exactly enough octets in this field to contain a number of bits equal to 128 minus prefix length, with 0 to 7 leading pad bits to make this field an integral number of octets. Pad bits, if present, MUST be set to zero when loading a zone file and ignored (other than for SIG [DNSSEC] verification) on reception.

- o The name of the prefix, encoded as a domain name. By the rules of [DNSIS], this name MUST NOT be compressed.

The domain name component SHALL NOT be present if the prefix length is zero. The address suffix component SHALL NOT be present if the prefix length is 128.

It is SUGGESTED that an A6 record intended for use as a prefix for other A6 records have all the insignificant trailing bits in its address suffix field set to zero.

4.1.2. Processing

A query with QTYPE=A6 causes type A6 and type NS additional section processing for the prefix names, if any, in the RDATA field of the A6 records in the answer section. This processing SHOULD be recursively applied to the prefix names of A6 records included as

additional data. When space in the reply packet is a limit, inclusion of additional A6 records takes priority over NS records.

It is an error for a A6 record with prefix length L1 > 0 to refer to a domain name which owns a A6 record with a prefix length L2 > L1. If such a situation is encountered by a resolver, the A6 record with the offending (larger) prefix length MUST be ignored. Robustness precludes signaling an error if addresses can still be formed from valid A6 records, but it is SUGGESTED that zone maintainers from time to time check all the A6 records their zones reference.

4.1.3. Textual Representation

The textual representation of the RDATA portion of the A6 resource record in a zone file comprises two or three fields separated by whitespace.

- A prefix length, represented as a decimal number between 0 and 128 inclusive,
- the textual representation of an IPv6 address as defined in [AARCH] (although some leading and/or trailing bits may not be significant),
- a domain name, if the prefix length is not zero.

The domain name MUST be absent if the prefix length is zero. The IPv6 address MAY be be absent if the prefix length is 128. A number of leading address bits equal to the prefix length SHOULD be zero, either implicitly (through the :: notation) or explicitly, as specified in section 4.1.1.

4.1.4. Name Resolution Procedure

To obtain the IPv6 address or addresses which belong to a given name, a DNS client MUST obtain one or more complete chains of A6 records, each chain beginning with a record owned by the given name and including a record owned by the prefix name in that record, and so on recursively, ending with an A6 record with a prefix length of zero. One IPv6 address is formed from one such chain by taking the value of each bit position from the earliest A6 record in the chain which validly covers that position, as indicated by the prefix length. The set of all IPv6 addresses for the given name comprises the addresses formed from all complete chains of A6 records beginning at that name, discarding records which have invalid prefix lengths as defined in section 4.1.2.

If some A6 queries fail and others succeed, a client might obtain a non-empty but incomplete set of IPv6 addresses for a host. In many situations this may be acceptable. The completeness of a set of A6 records may always be determined by inspection.

4.2. Zone Structure for Reverse Lookups

Very little of the new scheme's data actually appears under IP6.ARPA; only the first level of delegation needs to be under that domain. More levels of delegation could be placed under IP6.ARPA if some top-level delegations were done via NS records instead of DNAME records, but this would incur some cost in renumbering ease at the level of TLAs [AGGR]. Therefore, it is declared here that all address space delegations SHOULD be done by the DNAME mechanism rather than NS.

In addition, since uniformity in deployment will simplify maintenance of address delegations, it is SUGGESTED that address and prefix information be stored immediately below a DNS label "IP6". Stated another way, conformance with this suggestion would mean that "IP6" is the first label in the RDATA field of DNAME records which support IPv6 reverse lookups.

When any "reserved" or "must be zero" bits are adjacent to a delegation boundary, the higher-level entity MUST retain those bits in its own control and delegate only the bits over which the lower-level entity has authority.

To find the name of a node given its IPv6 address, a DNS client MUST perform a query with QCLASS=IN, QTYPE=PTR on the name formed from the 128 bit address as one or more bit-string labels [BITLBL], followed by the two standard labels "IP6.ARPA". If recursive service was not obtained from a server and the desired PTR record was not returned, the resolver MUST handle returned DNAME records as specified in [DNAME], and NS records as specified in [DNSCF], and iterate.

5. Modifications to Existing Query Types

All existing query types that perform type A additional section processing, i.e. the name server (NS), mail exchange (MX), and mailbox (MB) query types, and the experimental AFS data base (AFSDB) and route through (RT) types, must be redefined to perform type A, A6 and AAAA additional section processing, with type A having the highest priority for inclusion and type AAAA the lowest. This redefinition means that a name server may add any relevant IPv4 and

IPv6 address information available locally to the additional section of a response when processing any one of the above queries. The recursive inclusion of A6 records referenced by A6 records already included in the additional section is OPTIONAL.

6. Usage Illustrations

This section provides examples of use of the mechanisms defined in the previous section. All addresses and domains mentioned here are intended to be fictitious and for illustrative purposes only. Example delegations will be on 4-bit boundaries solely for readability; this specification is indifferent to bit alignment.

Use of the IPv6 aggregatable address format [AGGR] is assumed in the examples.

6.1. A6 Record Chains

Let's take the example of a site X that is multi-homed to two "intermediate" providers A and B. The provider A is itself multi-homed to two "transit" providers, C and D. The provider B gets its transit service from a single provider, E. For simplicity suppose that C, D and E all belong to the same top-level aggregate (TLA) with identifier (including format prefix) '2345', and the TLA authority at ALPHA-TLA.ORG assigns to C, D and E respectively the next level aggregate (NLA) prefixes 2345:00C0::/28, 2345:00D0::/28 and 2345:000E::/32.

C assigns the NLA prefix 2345:00C1:CA00::/40 to A, D assigns the prefix 2345:00D2:DA00::/40 to A and E assigns 2345:000E:EB00::/40 to B.

A assigns to X the subscriber identification '11' and B assigns the subscriber identification '22'. As a result, the site X inherits three address prefixes:

- 2345:00C1:CA11::/48 from A, for routes through C.
- 2345:00D2:DA11::/48 from A, for routes through D.
- 2345:000E:EB22::/48 from B, for routes through E.

Let us suppose that N is a node in the site X, that it is assigned to subnet number 1 in this site, and that it uses the interface identifier '1234:5678:9ABC:DEF0'. In our configuration, this node will have three addresses:

- 2345:00C1:CA11:0001:1234:5678:9ABC:DEF0
- 2345:00D2:DA11:0001:1234:5678:9ABC:DEF0
- 2345:000E:EB22:0001:1234:5678:9ABC:DEF0

6.1.1. Authoritative Data

We will assume that the site X is represented in the DNS by the domain name X.EXAMPLE, while A, B, C, D and E are represented by A.NET, B.NET, C.NET, D.NET and E.NET. In each of these domains, we assume a subdomain "IP6" that will hold the corresponding prefixes. The node N is identified by the domain name N.X.EXAMPLE. The following records would then appear in X's DNS.

```
$ORIGIN X.EXAMPLE.
N            A6 64 ::1234:5678:9ABC:DEF0 SUBNET-1.IP6
SUBNET-1.IP6 A6 48 0:0:0:1::  IP6
IP6          A6 48 0::0       SUBSCRIBER-X.IP6.A.NET.
IP6          A6 48 0::0       SUBSCRIBER-X.IP6.B.NET.
```

And elsewhere there would appear

```
SUBSCRIBER-X.IP6.A.NET. A6 40 0:0:0011:: A.NET.IP6.C.NET.
SUBSCRIBER-X.IP6.A.NET. A6 40 0:0:0011:: A.NET.IP6.D.NET.

SUBSCRIBER-X.IP6.B.NET. A6 40 0:0:0022:: B-NET.IP6.E.NET.

A.NET.IP6.C.NET. A6 28 0:0001:CA00:: C.NET.ALPHA-TLA.ORG.

A.NET.IP6.D.NET. A6 28 0:0002:DA00:: D.NET.ALPHA-TLA.ORG.

B-NET.IP6.E.NET. A6 32 0:0:EB00::    E.NET.ALPHA-TLA.ORG.

C.NET.ALPHA-TLA.ORG. A6 0 2345:00C0::
D.NET.ALPHA-TLA.ORG. A6 0 2345:00D0::
E.NET.ALPHA-TLA.ORG. A6 0 2345:000E::
```

6.1.2. Glue

When, as is common, some or all DNS servers for X.EXAMPLE are within the X.EXAMPLE zone itself, the top-level zone EXAMPLE must carry enough "glue" information to enable DNS clients to reach those nameservers. This is true in IPv6 just as in IPv4. However, the A6 record affords the DNS administrator some choices. The glue could be any of

- a minimal set of A6 records duplicated from the X.EXAMPLE zone,
- a (possibly smaller) set of records which collapse the structure of that minimal set,
- or a set of A6 records with prefix length zero, giving the entire global addresses of the servers.

The trade-off is ease of maintenance against robustness. The best and worst of both may be had together by implementing either the first or second option together with the third. To illustrate the glue options, suppose that X.EXAMPLE is served by two nameservers NS1.X.EXAMPLE and NS2.X.EXAMPLE, having interface identifiers ::1:11:111:1111 and ::2:22:222:2222 on subnets 1 and 2 respectively. Then the top-level zone EXAMPLE would include one (or more) of the following sets of A6 records as glue.

```
$ORIGIN EXAMPLE.            ; first option
X               NS NS1.X
                NS NS2.X
NS1.X           A6 64 ::1:11:111:1111 SUBNET-1.IP6.X
NS2.X           A6 64 ::2:22:222:2222 SUBNET-2.IP6.X
SUBNET-1.IP6.X  A6 48 0:0:0:1::       IP6.X
SUBNET-2.IP6.X  A6 48 0:0:0:2::       IP6.X
IP6.X           A6 48 0::0            SUBSCRIBER-X.IP6.A.NET.
IP6.X           A6 48 0::0            SUBSCRIBER-X.IP6.B.NET.


$ORIGIN EXAMPLE.            ; second option
X               NS NS1.X
                NS NS2.X
NS1.X           A6 48 ::1:1:11:111:1111 SUBSCRIBER-X.IP6.A.NET.
                A6 48 ::1:1:11:111:1111 SUBSCRIBER-X.IP6.B.NET.
NS2.X           A6 48 ::2:2:22:222:2222 SUBSCRIBER-X.IP6.A.NET.
                A6 48 ::2:2:22:222:2222 SUBSCRIBER-X.IP6.B.NET.


$ORIGIN EXAMPLE.            ; third option
X               NS NS1.X
                NS NS2.X
NS1.X           A6 0  2345:00C1:CA11:1:1:11:111:1111
                A6 0  2345:00D2:DA11:1:1:11:111:1111
                A6 0  2345:000E:EB22:1:1:11:111:1111
NS2.X           A6 0  2345:00C1:CA11:2:2:22:222:2222
                A6 0  2345:00D2:DA11:2:2:22:222:2222
                A6 0  2345:000E:EB22:2:2:22:222:2222
```

The first and second glue options are robust against renumbering of

X.EXAMPLE's prefixes by providers A.NET and B.NET, but will fail if those providers' own DNS is unreachable. The glue records of the third option are robust against DNS failures elsewhere than the zones EXAMPLE and X.EXAMPLE themselves, but must be updated when X's address space is renumbered.

If the EXAMPLE zone includes redundant glue, for instance the union of the A6 records of the first and third options, then under normal circumstances duplicate IPv6 addresses will be derived by DNS clients. But if provider DNS fails, addresses will still be obtained from the zero-prefix-length records, while if the EXAMPLE zone lags behind a renumbering of X.EXAMPLE, half of the addresses obtained by DNS clients will still be up-to-date.

The zero-prefix-length glue records can of course be automatically generated and/or checked in practice.

6.1.3. Variations

Several more-or-less arbitrary assumptions are reflected in the above structure. All of the following choices could have been made differently, according to someone's notion of convenience or an agreement between two parties.

First, that site X has chosen to put subnet information in a separate A6 record rather than incorporate it into each node's A6 records.

Second, that site X is referred to as "SUBSCRIBER-X" by both of its providers A and B.

Third, that site X chose to indirect its provider information through A6 records at IP6.X.EXAMPLE containing no significant bits. An alternative would have been to replicate each subnet record for each provider.

Fourth, B and E used a slightly different prefix naming convention between themselves than did A, C and D. Each hierarchical pair of network entities must arrange this naming between themselves.

Fifth, that the upward prefix referral chain topped out at ALPHA-TLA.ORG. There could have been another level which assigned the TLA values and holds A6 records containing those bits.

Finally, the above structure reflects an assumption that address fields assigned by a given entity are recorded only in A6 records

held by that entity. Those bits could be entered into A6 records in the lower-level entity's zone instead, thus:

```
IP6.X.EXAMPLE. A6 40 0:0:11::   IP6.A.NET.
IP6.X.EXAMPLE. A6 40 0:0:22::   IP6.B.NET.

IP6.A.NET.     A6 28 0:1:CA00:: IP6.C.NET.
and so on.
```

Or the higher-level entities could hold both sorts of A6 records (with different DNS owner names) and allow the lower-level entities to choose either mode of A6 chaining. But the general principle of avoiding data duplication suggests that the proper place to store assigned values is with the entity that assigned them.

It is possible, but not necessarily recommended, for a zone maintainer to forego the renumbering support afforded by the chaining of A6 records and to record entire IPv6 addresses within one zone file.

6.2. Reverse Mapping Zones

Supposing that address space assignments in the TLAs with Format Prefix (001) binary and IDs 0345, 0678 and 09AB were maintained in zones called ALPHA-TLA.ORG, BRAVO-TLA.ORG and CHARLIE-TLA.XY, then the IP6.ARPA zone would include

```
$ORIGIN IP6.ARPA.
\[x234500/24]   DNAME   IP6.ALPHA-TLA.ORG.
\[x267800/24]   DNAME   IP6.BRAVO-TLA.ORG.
\[x29AB00/24]   DNAME   IP6.CHARLIE-TLA.XY.
```

Eight trailing zero bits have been included in each TLA ID to reflect the eight reserved bits in the current aggregatable global unicast addresses format [AGGR].

6.2.1. The TLA level

ALPHA-TLA's assignments to network providers C, D and E are reflected in the reverse data as follows.

```
\[xC/4].IP6.ALPHA-TLA.ORG.   DNAME  IP6.C.NET.
\[xD/4].IP6.ALPHA-TLA.ORG.   DNAME  IP6.D.NET.
\[x0E/8].IP6.ALPHA-TLA.ORG.  DNAME  IP6.E.NET.
```

6.2.2. The ISP level

The providers A through E carry the following delegation information in their zone files.

```
\[x1CA/12].IP6.C.NET.  DNAME  IP6.A.NET.
\[x2DA/12].IP6.D.NET.  DNAME  IP6.A.NET.
\[xEB/8].IP6.E.NET.    DNAME  IP6.B.NET.
\[x11/8].IP6.A.NET.    DNAME  IP6.X.EXAMPLE.
\[x22/8].IP6.B.NET.    DNAME  IP6.X.EXAMPLE.
```

Note that some domain names appear in the RDATA of more than one DNAME record. In those cases, one zone is being used to map multiple prefixes.

6.2.3. The Site Level

Consider the customer X.EXAMPLE using IP6.X.EXAMPLE for address-to-name translations. This domain is now referenced by two different DNAME records held by two different providers.

```
$ORIGIN IP6.X.EXAMPLE.
\[x0001/16]                    DNAME   SUBNET-1
\[x123456789ABCDEF0].SUBNET-1  PTR     N.X.EXAMPLE.
and so on.
```

SUBNET-1 need not have been named in a DNAME record; the subnet bits could have been joined with the interface identifier. But if subnets are treated alike in both the A6 records and in the reverse zone, it will always be possible to keep the forward and reverse definition data for each prefix in one zone.

6.3. Lookups

A DNS resolver looking for a hostname for the address 2345:00C1:CA11:0001:1234:5678:9ABC:DEF0 would acquire certain of the DNAME records shown above and would form new queries. Assuming that it began the process knowing servers for IP6.ARPA, but that no server it consulted provided recursion and none had other useful additional information cached, the sequence of queried names and responses would be (all with QCLASS=IN, QTYPE=PTR):

To a server for IP6.ARPA:
QNAME=\[x234500C1CA110001123456789ABCDEF0/128].IP6.ARPA.

Answer:
\[x234500/24].IP6.ARPA. DNAME IP6.ALPHA-TLA.ORG.

To a server for IP6.ALPHA-TLA.ORG:
QNAME=\[xC1CA110001123456789ABCDEF0/104].IP6.ALPHA-TLA.ORG.

Answer:
\[xC/4].IP6.ALPHA-TLA.ORG. DNAME IP6.C.NET.

To a server for IP6.C.NET.:
QNAME=\[x1CA110001123456789ABCDEF0/100].IP6.C.NET.

Answer:
\[x1CA/12].IP6.C.NET. DNAME IP6.A.NET.

To a server for IP6.A.NET.:
QNAME=\[x110001123456789ABCDEF0/88].IP6.A.NET.

Answer:
\[x11/8].IP6.A.NET. DNAME IP6.X.EXAMPLE.

To a server for IP6.X.EXAMPLE.:
QNAME=\[x0001123456789ABCDEF0/80].IP6.X.EXAMPLE.

Answer:
\[x0001/16].IP6.X.EXAMPLE. DNAME SUBNET-1.IP6.X.EXAMPLE.
\[x123456789ABCDEF0/64].SUBNET-1.X.EXAMPLE. PTR N.X.EXAMPLE.

All the DNAME (and NS) records acquired along the way can be cached to expedite resolution of addresses topologically near to this address. And if another global address of N.X.EXAMPLE were resolved within the TTL of the final PTR record, that record would not have to be fetched again.

6.4. Operational Note

In the illustrations in section 6.1, hierarchically adjacent entities, such as a network provider and a customer, must agree on a DNS name which will own the definition of the delegated prefix(es). One simple convention would be to use a bit-string label representing exactly the bits which are assigned to the lower-level entity by the higher. For example, "SUBSCRIBER-X" could be replaced by "\[x11/8]". This would place the A6 record(s) defining the

delegated prefix at exactly the same point in the DNS tree as the DNAME record associated with that delegation. The cost of this simplification is that the lower-level zone must update its upward-pointing A6 records when it is renumbered. This cost may be found quite acceptable in practice.

7. Transition from RFC 1886 and Deployment Notes

When prefixes have been "delegated upward" with A6 records, the number of DNS resource records required to establish a single IPv6 address increases by some non-trivial factor. Those records will typically, but not necessarily, come from different DNS zones (which can independently suffer failures for all the usual reasons). When obtaining multiple IPv6 addresses together, this increase in RR count will be proportionally less -- and the total size of a DNS reply might even decrease -- if the addresses are topologically clustered. But the records could still easily exceed the space available in a UDP response which returns a large RRset [DNSCLAR] to an MX, NS, or SRV query, for example. The possibilities for overall degradation of performance and reliability of DNS lookups are numerous, and increase with the number of prefix delegations involved, especially when those delegations point to records in other zones.

DNS Security [DNSSEC] addresses the trustworthiness of cached data, which is a problem intrinsic to DNS, but the cost of applying this to an IPv6 address is multiplied by a factor which may be greater than the number of prefix delegations involved if different signature chains must be verified for different A6 records. If a trusted centralized caching server (as in [TSIG], for example) is used, this cost might be amortized to acceptable levels. One new phenomenon is the possibility that IPv6 addresses may be formed from a A6 records from a combination of secure and unsecured zones.

Until more deployment experience is gained with the A6 record, it is recommended that prefix delegations be limited to one or two levels. A reasonable phasing-in mechanism would be to start with no prefix delegations (all A6 records having prefix length 0) and then to move to the use of a single level of delegation within a single zone. (If the TTL of the "prefix" A6 records is kept to an appropriate duration the capability for rapid renumbering is not lost.) More aggressively flexible delegation could be introduced for a subset of hosts for experimentation.

7.1. Transition from AAAA and Coexistence with A Records

Administrators of zones which contain A6 records can easily accommodate deployed resolvers which understand AAAA records but not A6 records. Such administrators can do automatic generation of AAAA records for all of a zone's names which own A6 records by a process which mimics the resolution of a hostname to an IPv6 address (see section 4.1.4). Attention must be paid to the TTL assigned to a generated AAAA record, which MUST be no more than the minimum of the TTLs of the A6 records that were used to form the IPv6 address in that record. For full robustness, those A6 records which were in different zones should be monitored for changes (in TTL or RDATA) even when there are no changes to zone for which AAAA records are being generated. If the zone is secure [DNSSEC], the generated AAAA records MUST be signed along with the rest of the zone data.

A zone-specific heuristic MAY be used to avoid generation of AAAA records for A6 records which record prefixes, although such superfluous records would be relatively few in number and harmless. Examples of such heuristics include omitting A6 records with a prefix length less than the largest value found in the zone file, or records with an address suffix field with a certain number of trailing zero bits.

On the client side, when looking up and IPv6 address, the order of A6 and AAAA queries MAY be configurable to be one of: A6, then AAAA; AAAA, then A6; A6 only; or both in parallel. The default order (or only order, if not configurable) MUST be to try A6 first, then AAAA. If and when the AAAA becomes deprecated a new document will change the default.

The guidelines and options for precedence between IPv4 and IPv6 addresses are specified in [TRANS]. All mentions of AAAA records in that document are henceforth to be interpreted as meaning A6 and/or AAAA records in the order specified in the previous paragraph.

7.2. Transition from Nibble Labels to Binary Labels

Implementations conforming to RFC 1886 perform reverse lookups as follows:

> An IPv6 address is represented as a name in the IP6.INT domain by a sequence of nibbles separated by dots with the suffix ".IP6.INT". The sequence of nibbles is encoded in reverse order, i.e. the low-order nibble is encoded first, followed by the next low-order nibble and so on. Each nibble is represented by a hexadecimal digit. For example, a name for the address

2345:00C1:CA11:0001:1234:5678:9ABC:DEF0 of the example in section 6.3 would be sought at the DNS name "0.f.e.d.c.b.a.9.-8.7.6.5.4.3.2.1.1.0.0.0.1.1.a.c.1.c.0.0.5.4.3.2.ip6.int."

Implementations conforming to this specification will perform a lookup of a binary label in IP6.ARPA as specified in Section 3.2. It is RECOMMENDED that for a transition period implementations first lookup the binary label in IP6.ARPA and if this fails try to lookup the 'nibble' label in IP6.INT.

8. Security Considerations

The signing authority [DNSSEC] for the A6 records which determine an IPv6 address is distributed among several entities, reflecting the delegation path of the address space which that address occupies. DNS Security is fully applicable to bit-string labels and DNAME records. And just as in IPv4, verification of name-to-address mappings is logically independent of verification of address-to-name mappings.

With or without DNSSEC, the incomplete but non-empty address set scenario of section 4.1.4 could be caused by selective interference with DNS lookups. If in some situation this would be more harmful than complete DNS failure, it might be mitigated on the client side by refusing to act on an incomplete set, or on the server side by listing all addresses in A6 records with prefix length 0.

9. IANA Considerations

The A6 resource record has been assigned a Type value of 38.

10. Acknowledgments

The authors would like to thank the following persons for valuable discussions and reviews: Mark Andrews, Rob Austein, Jim Bound, Randy Bush, Brian Carpenter, David Conrad, Steve Deering, Francis Dupont, Robert Elz, Bob Fink, Olafur Gudmundsson, Bob Halley, Bob Hinden, Edward Lewis, Bill Manning, Keith Moore, Thomas Narten, Erik Nordmark, Mike O'Dell, Michael Patton and Ken Powell.

11. References

[AARCH] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[AGGR] Hinden, R., O'Dell, M. and S. Deering, "An IPv6 Aggregatable Global Unicast Address Format". RFC 2374, July 1998.

[BITLBL] Crawford, M., "Binary Labels in the Domain Name System", RFC 2673, August 1999.

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

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

[DNSIS] Mockapetris, P. V., "Domain names - implementation and specification", RFC 1035, November 1987.

[DNSSEC] Eastlake, D. 3rd and C. Kaufman, "Domain Name System Security Extensions", RFC 2535, March 1999.

[KWORD] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119.

[RENUM] Carpenter, B. and Y. Rekhter, "Renumbering Needs Work", RFC 1900, February 1996.

Ferguson, P. and H. Berkowitz, "Network Renumbering Overview: Why would I want it and what is it anyway?", RFC 2071, January 1997.

Carpenter, B., Crowcroft, J. and Y. Rekhter, "IPv4 Address Behaviour Today", RFC 2101, February 1997.

[TRANS] Gilligan, R. and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 1933, April 1996.

[TSIG] Vixie, P., Gudmundsson, O., Eastlake, D. 3rd and B. Wellington, "Secret Key Transaction Authentication for DNS

(TSIG)", work in progress.

12. Authors' Addresses

Matt Crawford	Christian Huitema	Susan Thomson
Fermilab	Telcordia	Telcordia
MS 368	MCC 1J236B	MCC 1C259B
PO Box 500	445 South Street	445 South Street
Batavia, IL 60510	Morristown, NJ 07960	Morristown, NJ 07960
USA	USA	USA
+1 630 840-3461	+1 201 829-4266	+1 201 829-4514
crawdad@fnal.gov	huitema@research.telcordia.com	set@research.telcordia.com