Network Working Group
Internet-Draft
17 October 2002
Expires: 17 April 2003

Keith Moore
University of Tennessee

6to4 and DNS

draft-moore-6to4-dns-03.txt

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

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

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

Comments regarding this internet-draft should be sent to the author. Refer to the IETF web site at http://www.ietf.org/ for current contact information for IETF working groups. Please include the document identifier "draft-moore-6to4-dns-03" in any comments regarding this document.

This document supersedes both draft-moore-6to4-dns-02.txt and (more recently) draft-ietf-ngtrans-6to4-dns-00.txt

Abstract

This memo discusses several potential mechanisms for locating the DNS servers which provide "reverse lookup" of 6to4 addresses.

Please note that this is a preliminary draft which only attempts to outline possible means of solving the problem, for purpose of discussion. This version of the proposal is NOT rigorously specified, and the author does not claims significant expertise in DNS. Nevertheless, it is hoped that these proposals are sufficiently detailed to allow reviewers to make a first-order assessment of their viability.

The assistance of appropriate experts in drafting future revisions of these proposals would be most welcome.

1. Introduction

6to4 [1] defines a mechanism for allowing sites to communicate using IPv6 over the public IPv4 Internet. It does so by assigning a block of IPv6 addresses corresponding to any "public" (globally-scoped) IPv4 address, and a means of tunneling IPv6 traffic destined for such addresses over the IPv4 Internet. In this way, any site which is connected to the IPv4 Internet and which has at least one global IPv4 address assigned to it, can communicate with IPv6.

The advantage of 6to4 is that it decouples deployment of IPv6 by the core of the network (e.g. Internet Service Providers or ISPs) from deployment of IPv6 at the edges (e.g. customer sites), allowing each site or ISP to deploy IPv6 support in its own time frame according to its own priorities. With 6to4, the edges may communicate with one another using IPv6 even if one or more of their ISPs do not yet provide native IPv6 service. In addition, the principal cost of the 6to4 transition mechanism is borne by those who benefit from it.

However, the ability to perform so-called "reverse lookups" (lookups of IP addresses rather than domain names) in DNS requires that there be a delegation path for the IP address being queried, from the DNS root to the servers for the DNA zone which provides the PTR records for that IP address. For ordinary IPv6 addresses, the necessary DNS servers and records for IPv6 reverse lookups would be maintained by the each organization to which an address block is delegated; the delegation path of DNS records reflects the delegation of address blocks themselves. However, for IPv6 addresses beginning with the 6to4 address prefix, the DNS records would need to reflect IPv4 address delegation. Since the entire motivation of 6to4 is to decouple site deployment of IPv6 from infrastructure deployment of IPv6, such records cannot be expected to be present for a site using 6to4 - especially for a site whose ISP did not yet support IPv6 in any form.

This memo discusses several potential mechanisms for locating the DNS servers which are assumed to provide "reverse lookup" of 6to4 addresses.

1.1. Notation

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 RFC 2119.

The characters "{" and "}" are used to indicate protocol elements where literal DNS labels or addresses would appear in actual use; neither these delimiters nor the text appearing within are to be interpreted literally.

2. Design Goals

An ideal solution to this problem would have several characteristics:

- Easy to deploy.
- Minimal impact on existing software and operations.
- Reasonable efficiency for lookup of names corresponding to 6to4 addresses.
- Minimal effort in deployment of DNS support.
- Costs and effort borne primarily by those who immediately benefit.
- Does not adversely affect security of DNS queries.
- Any assumptions made by client or server software as to the location of authoritative DNS server(s) for reverse lookup of a 6to4 address, are made only if no explicit referral information is present.
- End-state is "normal" DNS operation with little or no additional overhead. No attempt has yet been made to establish relative importance of these goals.

3. Methods of Inferring Delegation Paths

The author has identified two methods of inferring delegation paths in the absence of explicit delegation information (NS records) for reverse lookups of IPv6 addresses in the DNS:

1. **Assume that the default DNS servers for lookup of a 6to4 address are the same servers that are responsible for reverse lookup of the** corresponding IPv4 address, OR

2. **Assume that the default DNS servers for lookup of a 6to4 address are reachable via some addresses which are derivable from the 6to4** prefix via a well-known algorithm.

While it might be possible to employ both of these methods, or use them in some combination, it seems better to choose one method or the

other.

In both methods, the actual PTR records for 6to4 addresses are explicitly maintained by the site to which that portion of 6to4 space is assigned (i.e. the site to whom the corresponding portion of IPv4 space - perhaps as little as a single IPv4 address - is assigned). This proposal does not make assumptions about, nor impose constraints on, the mapping between specific 6to4 addresses and specific host names.

3.1 6to4 NS records derived from IPv4 NS records

This method assumes that the default DNS servers for a zone that provides lookup of a 6to4 address are the same servers that are responsible for lookup of the corresponding IPv4 address. More specifically, for every NS resource record that refers queries for a portion of IN-ADDR.ARPA space to some set of DNS servers, we want to behave as if there were a similar NS record for the portion of IP6.ARPA space corresponding to those IPv4 addresses, in the absence of any explicit NS records for those names in IP6.ARPA space.

More precisely, for every resource record of the form:

```
{address-bits}.IN-ADDR.ARPA.  NS   some-domain.example.com.
```

we want to have the effect of also having a resource record of the form:

```
{address-bits}.2.0.0.2.IP6.ARPA. NS     some-domain.example.com.
```

unless the lookup for the IPv6 address can be fulfilled by a chain of explicit NS and PTR resource records. The following sections discuss various ways of producing the effect. The NS records so generated or assumed (by whatever means) are termed "pseudo-records" to distinguish them from explicitly-supplied NS records.

Note that due to the different ways of representing {address-bits} in DNS labels between IPv4 and IPv6, a transformation will be required. In particular, each label under IN-ADDR.ARPA that represents an octet of a v4 address must be transformed into two labels in IP6.ARPA each representing four bits (quartet?) of the v6 address prefix. In addition, the TTLs of the generated NS pseudo-records MUST NOT be larger than those of the NS records from which they were derived; in some cases it may be desirable to make them smaller.

This method has the advantage that 6to4 sites do not need to establish new DNS servers, nor to get those servers to answer to new addresses, in order to implement reverse lookup service for 6to4 addresses. It need only add the appropriate resource records to its existing DNS servers which perform those functions for IPv4.

However, this method only works for sites that already operate their own DNS servers that provide lookup for IPv4 addresses. In particular, sites with only a single IPv4 address may support a significant population of 6to4 users. However such a site is unlikely to be delegated authority to provide address lookup for its single IPv4 address, nor in most cases would such a site want to provide DNS servers for reverse lookup of a single IPv4 address.

3.2 6to4 NS records inferred from 6to4 prefix

This method assumes that the default DNS servers for lookup of a 6to4 address are reachable at a set of addresses which are derivable from the 6to4 prefix.

More formally, in the absence of any explicit NS resource records for the suffix {IPv4-address}.2.0.0.2.IP6.ARPA, resource records of the form

```
{IPv4-address}.2.0.0.2.IP6.ARPA. NS     {label}
{label}                     AAAA 2002:{IPv4-address}:{suffix1}
                                 AAAA 2002:{IPv4-address}:{suffix2}
                                 AAAA 2002:{IPv4-address}:{suffix3}
```

are inferred, where {suffix1}, {suffix2}, and {suffix3} are constant bit patterns that are to be determined. Here, {IPv4-address} is 32 bits of IPv4 address, {label} is a domain-name which is created for the purpose of associating a 6to4 address with its DNS servers (since NS records must refer to a DNS name rather than an IPv6 address). The 6to4 network then arranges for DNS servers to respond to queries that are sent to those IPv6 addresses.

Note that if a site uses more than one 6to4 prefix (because it has more than one IPv4 address assigned to it), its DNS servers which are responsible for reverse lookups will be required to accept queries at multiple addresses.

A variant of this method would be to define a single suffix for this purpose, rather than multiple suffixes, and to infer a single AAAA record rather than multiple AAAA records. Multiple servers could be supported by treating that single address as an anycast address. One difficulty with using anycast is in arranging for the hosts to respond to Neighbor Solicitation queries at those addresses only when the DNS servers on those hosts are correctly operating. Absent such a mechanism, client-based fail-over between separate addresses appears more reliable, if slower, than server selection by anycast.

4 Methods of adapting existing software to infer delegation paths

The following paragraphs detail several possible techniques which might allow existing platforms to infer these delegation paths with varying degrees of disruption. They are not mutually-exclusive; it is possible to employ more than of these techniques. Some of them are less attractive than others. At present the purpose of this document is to outline several possible approaches, and serve as a focal point of discussion, rather than to categorically recommend any particular approach.

Most of these implementation methods can be used with either method of inferring NS records - either deriving them from v4 NS records (section 3.1) or using well-known address suffixes (section 3.2).

4.1. Explicit delegation of NS records for 6to4 address lookup

This implementation method makes no changes to any DNS client or server software. Rather, it expects that the root servers, ISPs DNS servers, and the DNS servers of other organizations which delegate IPv4 address space, will be populated with NS records which refer lookup queries from 6to4 space.

Unless and until the assignee of the IPv4 address requested that new NS RRs be installed under {ipv4-address}.2.0.0.2.IP6.ARPA to point to the assignee's DNS servers, any changes made to the NS records under IN-ADDR.ARPA would also need to be reflected in the corresponding NS records under IP6.ARPA.

As stated above, this technique requires no software changes to either DNS server or client software. However, it would certainly require changes to the software used by registries, ISPs, and other networks, to maintain the DNS records needed to provide reverse lookups.

This implementation method may be used with either method of inferring NS records. In other words, either new NS records could be derived from existing NS records for IPv4 addresses, or new NS and AAAA records could be created assuming that servers would be established at one or more well-known suffixes within a 6to4 subnet prefix. In either case a site MUST be allowed to change the records associated with its 6to4 prefix after they are established.

This implementation method avoids kludges to DNS software but is assumed to be difficult to deploy, as it requires several different organizations (most with no previous relationship with, or direct responsibility to, sites) to explicitly support 6to4. Another problem is that in the absence of a complete chain of NS records from 2.0.0.2.IP6.ARPA to the site's DNS servers, lookups will still fail -

that is, every organization that delegates address space is required to cooperate before it will work.

4.2. Pseudo-records generated by DNS servers for the IPv4 zones

In this technique, the authoritative DNS servers for IP6.ARPA and its subdomains would be modified to return "pseudo-records" for any query for a name of the form "{something}.2.0.0.2.IP6.ARPA".

In particular,

- if the server had explicit resource records entirely matching the name of a query (the name ending in 2.0.0.2.IP6.ARPA), those records would be returned in response to that query and no pseudo-records would be returned;

- if the server had explicit NS records for a suffix of a name ending in "2.0.0.2.IP6.ARPA" (the suffix longer than "2.0.0.2.IP6.ARPA") that matched the name of a query, those records would be returned in response to that query and no pseudo-records would be returned;

- (if the method in section 3.1 were used) otherwise, if the IN-ADDR.ARPA zone had NS records matching {something}.IN-ADDR.ARPA, or matching any IPv4 address prefix of {something}.IN-ADDR.ARPA, NS pseudo-records corresponding to the longest matching prefixes would be returned. The pseudo-records so returned would be marked authoritative, and their TTLs would be no larger than the TTLs of the explicit records from which the pseudo-records were derived. (Note: It's difficult to see how the "longest match" could be done efficiently given that IPv4 address blocks are delegated on single-bit boundaries.)

- (if the method in section 3.2 were used) otherwise, the server would return an NS pseudo-record corresponding to the 6to4 prefix, which pointed to a label for which one or more AAAA pseudo-records containing the well-known address(es) for address lookups for addresses beginning with that prefix. The AAAA addresses would be returned as additional information in response to the query, but would necessarily also be obtainable from a separate AAAA or A6 query for any {label} returned in an NS pseudo-record.

This technique is assumed to be somewhat easier to deploy than the previous one, because it automates the generation of the pseudo-records and avoids the need for each organization that delegates IPv4 space to change its DNS maintenance procedures. However, it still requires changes to DNS servers, and it requires those organizations to upgrade their DNS servers to include those changes, before the changes will be useful. It also requires cooperation on behalf of the owner of the DNS

servers providing lookup for an IPv4 address, which might not be the same party that is using the corresponding 6to4 addresses.

4.3. Pseudo-records generated by DNS resolvers

In this technique, DNS servers which act as resolvers behave as if pseudo-records had been returned to them when some kinds of queries fail. In some cases they may return pseudo-records when a query fails.

When such a resolver received a query for a name that had a 2.0.0.2.IP6.ARPA suffix, it would first attempt to satisfy that query from its cache, or failing that, by forwarding the query to an upstream server. If that query failed due to a "no such domain" error, the resolver would then attempt to find the server for the {something}.2.0.0.2.IP6.ARPA name by (if the method in section 3.1 were used) issuing an NS query for {something}.IN-ADDR.ARPA, or (if the method in section 3.2 were used) inferring NS and AAAA records based on the 6to4 prefix derived from the IPv4 address.

If the method in section 3.1 were used, and the original query was for PTR records, and one or more NS records were found for {something}.IN-ADDR.ARPA, the resolver would then forward the original query for {something}.2.0.0.2.IP6.ARPA to one or more of those servers, and return the results from one of the forwarded queries if any were successful. If the original query was for NS records, and one or more NS records were found for {something}.IN-ADDR.ARPA, the resolver would then return the pseudo-records corresponding to the IN-ADDR.ARPA domains. Those pseudo-records would NOT be marked as authoritative, and the resolver would NOT cache those records.

Similarly, if the method in section 3.2 were used, the resolver would return NS and AAAA pseudo-records derived from the IPv6 address being queried.

Note that while the DNS resolver effectively behaves as if pseudo-records had been returned to it by other servers, it MUST NOT cache those pseudo-records. However, it MAY cache the actual NS or PTR records returned by those servers and use such cached data to generate additional pseudo-records.

This technique requires changes to DNS resolver software, and requires that sites using IPv6 and wishing to communicate with 6to4 sites, upgrade their DNS resolvers to include this change. However it does not require changes of IPv6 hosts.

4.4 Pseudo-records generated by DNS query libraries

In this technique, the run-time library used on a host by applications is modified to process DNS queries in the following manner:

If the name queried has a suffix of 2.0.0.2.IP6.ARPA, or if the query is otherwise intended to perform an address lookup (perhaps as a side effect), an attempt is first made to look up the address via normal means. If this attempt failed due to the lack of any delegation of the 6to4 prefix, NS and perhaps AAAA pseudo-records for the label of the NS are inferred according to sections 3.1 and/or 3.2 (whichever ends up being chosen). If this secondary query is successful, the original DNS query for the 6to4 address is re-issued to the servers which are authoritative for that IPv4 address; the result of the library call is determined from the response to that query.

This technique requires changes to DNS query libraries (and applications), and requires that hosts and/or applications using IPv6, and which wish to communicate with hosts and/or applications at 6to4 sites, upgrade their DNS libraries to include this change.

5. Author's Recommendations

For the purpose of facilitating discussion, the author tentatively recommends that the following combination of methods be used:

Locations of DNS servers to be used for address lookups should be obtained in the following manner:

- First, attempt to perform the lookup in the normal way used for any IPv6 address, by issuing a query for {address}.IP6.ARPA. If the result of this query is one or more PTR records, these results are used and the lookup is complete.

- Else, if the result of this query indicates that lookups for a prefix of the queried IPv6 address, greater than or equal to 48 bits in length, have explicitly been delegated, but the query could not be completed due to some error, the error is returned and the lookup is complete.

- Else, the method of inferring NS and AAAA records described in section 3.2 is used, with two or three well-known suffixes chosen rather than a single anycast address. Assigning two or three well-known suffixes rather than a single suffix allows a small site to provide redundant servers for reverse lookup without having to implement anycast.

This method is recommended both in preference to, and instead of, the method in section 3.1 because it is anticipated that many 6to4 sites will be using a single IPv4 address and will not have reverse lookup for that IPv4 address delegated to their name servers. (In other words, NS records delegating the reverse lookup of 32-bit IPv4 prefixes are assumed to be rare.)

Implementation of the above algorithm should be provided by both host-based DNS query libraries and (as a configuration option) by resolver servers. Thus, if either the host-based query library (for dynamically-linked applications) or the local resolver server has been upgraded to infer delegation of 6to4 addresses, applications on that host will be able to perform lookups of 6to4 addresses in the absence of explicit delegation.

This compromise largely preserves the favorable deployment characteristics of 6to4 - namely, that hosts and networks can use 6to4 without explicit support from the existing IPv4 network infrastructure. Implementing the algorithm in both query libraries in resolvers allows existing IPv6 hosts and applications to lookup 6to4 addresses without having to upgrade all of their hosts, while still allowing lookups for single hosts and small sites which cannot reconfigure their DNS resolver servers. However it does require that all IPv6 sites - not just those on 6to4 networks - upgrade their query libraries and/or resolvers if they wish to perform reverse lookups on 6to4 addresses.

Meanwhile, root servers, regional address registries, and ISPs are encouraged to populate and maintain the 2.0.0.2.IP6.ARPA zone to refer queries for 6to4 addresses to the same servers as are used to look up the corresponding IPv4 addresses in the IN-ADDR.ARPA zone and ISPs are encouraged to provide their customers who have statically allocated IPv4 addresses with the ability to establish new NS records for the corresponding portion of v6 space.

6. Security Considerations

The use of well-known address suffixes for DNS servers would allow hosts that could choose their own addresses to provide inverse name lookups in the absence of explicit delegation by the network administrators. For this reason, it is necessary to check for explicit delegation of address lookup service before using results obtained from queries to well-known addresses.

In addition, sites running 6to4 which do not provide address lookup service at each of the well-known address suffixes, should take measures to prevent ordinary hosts from assuming the role of DNS servers. For example, a site might make a decision to disallow those addresses being used by ordinary hosts and to filter any traffic originating from those

addresses which were not assigned to DNS servers.

Pseudo-records that are automatically derived from other DNS records cannot be signed using DNSSEC, even if the explicit records from which the pseudo-records are derived are signed. Since explicit records take precedence over pseudo-records, a host or application SHOULD NOT trust a signed NS record referring a query for some portion of IPv4 space as evidence of authoritative referral to the corresponding portion of 6to4 space unless it has evidence that there are no explicit records present for that portion of 6to4 space.

7. Author's Address

Keith Moore
University of Tennessee, Knoxville
1122 Volunteer Blvd, Suite 203
Knoxville TN, 37996-3450
USA
email: moore@cs.utk.edu

8. References

[1]. Carpenter, B., Moore, K. Connection of IPv6 Domains via IPv4 Clouds. RFC 3056, February 2001.

[2]. Crawford, M., Huitema, C., Thomson, S. DNS Extensions to Support IPv6 Address Aggregation and Renumbering. RFC 2874, July 2000.