                    DNS64 Resolvers and Dual-Stack Hosts
                     draft-wing-behave-dns64-config-02

Abstract

   Some networks are expected to support IPv4-only, dual-stack, and
   IPv6-only hosts at the same time. Such networks also want to IPv6/
   IPv4 translation for the IPv6-only host so it can access servers on
   the IPv4 Internet. On such a network, the synthesized AAAA responses
   from a DNS64 can cause traffic to be translated. This document
   describes and analyzes several solutions to avoid that translation.

Status of this Memo

   This Internet-Draft is submitted to IETF in full conformance with the
   provisions of BCP 78 and BCP 79.

   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.

   This Internet-Draft will expire on August 16, 2010.

Table of Contents

1. Introduction

   In order to access IPv4 servers, an IPv6-only host needs to use an
   IPv6/IPv4 translator. Typically, the IPv6-only host performs a DNS
   query to a DNS64 recursive resolver, which synthesizes an AAAA when
   necessary. However, if a dual-stack host uses that same DNS64
   recursive resolver and normal address selection rules [RFC3484], the
   dual-stack host will send traffic through the IPv6/IPv4 translator
   when such traffic could have been sent using IPv4. Thus, as an
   optimization, it is desirable that a dual-stack host avoid IPv6/IPv4
   translation.

   Note: If the dual-stack host's IPv4 traffic is being NATted the
   difference is NAT44 versus NAT64, so the performance and scalability
   concern is nearly identical. However, at least one application
   breaks when translated between IP address families unless special
   measures are taken [I-D.ietf-behave-ftp64]. The IETF should decide
   if it is worthwhile to avoid NAT64 for dual-stack hosts that are
   connected to a network operating a DNS64.

      Note: Windows XP can only be configured with IPv4 DNS servers
      [XP-DNS]. This means a Windows XP host is always dual-stack and
      requires an IPv4 address in order to send its DNS queries. While
      it is possible to work around this issue by running BIND on the
      Windows XP device itself, this is complex. Thus, Windows XP
      should not be considered a viable operating system to join an
      IPv6-only network.

2. Terminology

   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 [RFC2119].

   "IPv4-only" means a host that has only IPv4 address(es) assigned to
   its interface(s). "Dual-stack" means a host that has an IPv4 address
   and an IPv6 address assigned to its interface(s). "IPv6-only" means
   a host that has only IPv6 address(es) assigned to its interface(s).

3. Procedures to support IPv6-only and dual-stack hosts with DNS64

   Several solutions are discussed in this section, in roughly the order
   of preference (as determined by the author).

3.1. IPv4-Mapped Address for 'normal' DNS server

It has been observed that some common operating systems, when
configured as dual-stack, will successfully use an IPv4-mapped
address (and send an IPv4 packet). But when configured as IPv6-only,
they will not successfully use an IPv4-mapped address (because they
lack an IPv4 address) [experiment].

We could take advantage of this by configuring the 'normal' DNS
server using an IPv4-mapped IPv6 address (that is, an IPv6 address
starting with ::ffff:/96), and configuring the DNS64 server using a
normal IPv6 address.

Because [RFC3646] says the DNS servers are used in the order listed,
a dual-stack host will use the 'normal' DNS server and an IPv6-only
host will be unable to use that 'normal' DNS server and will use the
next server on its list.

   Note: Non-compliant IPv6 stacks might send a packet to the IPv4-
   mapped IPv6 address (::ffff:c000:0201, using the example below).
   To deal with such non-compliant IPv6 implementations the network
   can filter (drop) traffic to that IPv6 address.

For example, a dual-stack host and an IPv6-only host would be
configured with the following DNS servers, in this order:

```
::ffff:192.0.2.1        # 'normal' DNS server (usable only by
                           dual-stack host.  Not usable by IPv6-only
                           host)
2001:0DB8:DDDD::1234   # DNS64 server
```

3.1.1. Host Transition

When transitioning from dual-stack to IPv6-only, nothing needs to
occur - the higher-priority DNS server (with the IPv4-mapped IPv6
address) will become inaccessible and the DNS client will failover to
the next-higher priority DNS server (which is the DNS64 server).

When transitioning from IPv6-only to dual-stack, nothing
automatically causes the host to start querying the 'normal' DNS
server. Thus, a host that transitions from IPv6-only to dual-stack
will continue to query the DNS64 until the host's stack re-
initializes.

3.1.2. Advantages and Disadvantages

Advantages:

o No change to hosts.

o On Linux systems, is not effective if the sysctl
   net.ipv6.bindv6only is set, as this causes dual-stack systems to
   not send packets to IPv4-mapped IPv6 addresses.

Disadvantages:

o Seems confusing to configure.

o Can we rely on IPv6-only hosts allowing IPv4-mapped IPv6 addresses
   to be configured as their DNS servers?

3.2. Disable DNS64 Functions in DNS Query

A dual-stack host does not need synthesized AAAA records, and does
not need to use the network's NAT64. So, a dual-stack host could
send a DNS query to a DNS64 requesting no synthesized AAAA records.
This allows both dual-stack and IPv6-only hosts to be configured with
a DNS64.

To request no AAAA synthesis, the DNS query sets both the DO bit ("I
understand DNSSEC") and the CD bit ("I will do DNSSEC validation").
When the DNS64 receives a query with those bits set, it does not
synthesize an AAAA response (because doing so would break DNSSEC
validation).

3.2.1. Host Transition

If host transitions from dual-stack to IPv6-only, it would need to
perform its own DNS64 function (which requires it know the prefix of
its NAT64) or would have to not set the DO and CD bits in DNS
queries. Likewise, if a host transitions from IPv6-only to dual-
stack, it needs to know to start sending DNS queries with the DO and
CD bits set.s

3.2.2. Advantages and Disadvantages

Advantages:

o Simple

Disadvantages:

o Requires dual-stack host support DNSSEC validation.

o Requires host support Section 3.2.1.

o   There are security implications if the DNS client is lying and is
    not, in fact, going to perform its own DNSSEC validation.

3.3. New DHCP option for 'normal' DNS server

Another approach, which requires modification of dual-stack hosts
which want to avoid the DNS64, is to introduce a new DHCP option.

This approach feels a little backwards at first. The idea is to
support unmodified hosts (which might be dual-stack but might be
IPv6-only) by placing DNS64 servers into the normal DHCPv6 option for
DNS servers [RFC3646]. Then, place the 'normal' DNS servers into a
new DHCPv6 option.

3.3.1. Host Transition

TBD.

3.3.2. Advantages and Disadvantages

Disadvantages:

o   If dual-stack hosts want to avoid NAT64, they need to be modified
    to understand this new DHCP option. If they aren't modified, they
    will use NAT64.

3.4. Modify Host's Address Selection Rules

The default address selection rules [RFC3484] prefer IPv6 over IPv4.
This means, for a dual-stack host, that IPv6 will be preferred (if
available) over IPv4. If a dual-stack host is configured to use a
DNS64 server, that DNS64 server will synthesize an AAAA response if
there is an A record. Thus, the dual-stack host will always use IPv6
if a DNS lookup was involved, even if IPv4 could have been used more
optimally.

   Note: If both a NAT44 and NAT64 are deployed on the same network,
   roughly the same inefficiency occurs (that is, NAT state is
   created). However, it is generally considered better to perform
   NAT44 than NAT64, because NAT64 translates between IP address
   families which can have side effects (e.g., FTP).

To avoid this, the host's default address selection rules [RFC3484]
can be modified so that IPv4 is preferred over the IPv6/IPv4
translator's prefix. At the same time, native IPv6 can still be
preferred over IPv4. This is accomplished by adding the network's
IPv6/IPv4 translator's prefix as the lowest Precedence in the address
selection rules.

If the IPv6/IPv4 translator's prefix is the IANA-assigned well-known
prefix (64:FF9B::/96, as assigned in
[I-D.ietf-behave-address-format]), this can be hard-coded or easily
scripted into the system startup. However, if the IPv6/IPv4
translator's prefix is a network-specific prefix (NSP, as described
in [I-D.ietf-behave-address-format]), the default address selection
rules can be modified only after the host learns its currently-
connected network's IPv6/IPv4 translator's prefix (e.g., using
[I-D.wing-behave-learn-prefix]).

On some operating systems, the address selection rules can be
configured using a command line utility (e.g., Windows, FreeBSD),
without new software in the host's IP stack. Other operating systems
are not as accommodating of this solution (see Section 3.4.2).

   Note: it may be desirable to create a standard to adjust a host's
   address selection rules based on the translator's prefix. This is
   a topic for the IPv6 maintenance working group [6man]. This
   automatic mechanism may involve modifications to the host's IP
   stack, depending on how the IETF chooses to standardize such a
   mechanism. FOR EXAMPLE, it may be useful to consider
   [I-D.wing-behave-learn-prefix] (which proposes using either DNS or
   DHCPv6) in conjunction with adjusting the host's address selection
   rules.

3.4.1. Host Transition

An IPv6-only and a dual-stack host can both be configured with the
same address selection rules (namely, both can add the network's
translator as the lowest Precedence). This is because the IPv6-only
host will never use IPv4 (because it lacks an IPv4 address) and will
thus fall through and use the IPv6 address synthesized by the DNS64
containing the IPv6/IPv4 translator's prefix (that is, as shown in
the examples, the IPv6-only host will use the Precedence 3 entry in
the default policy table). The dual-stack host, if it receives an
AAAA response, will prefer use IPv6; if it receives only an A
response, it will prefer to use IPv4 (using Precedence 10 for IPv4-
mapped addresses defined in Section 2.5.4 of [RFC2373]).

3.4.2. Limitations and Advantages

The following limitations are observed:

o OSX does not implement a [RFC3484] or [RFC3484]-like policy table.

o Some applications implement their own address selection rules,
   effectively ignoring the OS's address selection rules.

The following advantages are observed:

o Causes IPv4 to be preferred over IPv6/IPv4 translator addresses,
   even if DNS was not used to obtain the IPv4 or IPv6 address (e.g.,
   applications which do not use DNS).

3.4.3. Examples

For example, if a network is using the WKP 64:FF9B::/96
[I-D.ietf-behave-address-format] and a host is using the new default
policy table from [I-D.arifumi-6man-rfc3484-revise] (which added
Precedence 5 for Teredo), the host's new policy table would contain
one new entry with Precedence 3, as shown below:

```
    Prefix           Precedence Label
    ::1/128                 50    0     # localhost
    ::/0                    30    2     # IPv6 native
    2002::/16               20    3     # 6to4
    ::ffff:0:0/96           10    4     # IPv4-mapped
    2001::/32                5    5     # Teredo
    64:FF9B::/96             3    6     # 6/4 translator's prefix
```

As another example, if a network has the prefix 2001:0DB8::/32 and
the NAT64 is using the Network-Specific Prefix (NSP) 2001:0DB8:
AAAA::/96, and the host is using the new default policy table from
[I-D.arifumi-6man-rfc3484-revise] (which added Precedence 5 for
Teredo), the host's new policy table would contain one new entry with
Precedence 3, as shown below:

```
    Prefix           Precedence Label
    ::1/128                 50    0     # localhost
    ::/0                    30    2     # IPv6 native
    2002::/16               20    3     # 6to4
    ::ffff:0:0/96           10    4     # IPv4-mapped
    2001::/32                5    5     # Teredo
    2001:0DB8:AAAA::/96      3    6     # 6/4 translator's prefix
```

3.5. Use DHCP to Assign Appropriate DNS Server

Note: due to the limitations of this solution (see
Section 3.5.5), it may have little or no value.

To avoid unnecessary traffic through a translator, it is desirable to
configure IPv4-only and dual-stack hosts with a 'normal' DNS
recursive resolver.

However, it is necessary to configure IPv6-only hosts with a DNS64
[I-D.ietf-behave-dns64] recursive resolver so those hosts can use an

IPv6/IPv4 translator and access servers on the IPv4 Internet.

It is difficult to provide different DNS servers to those types of hosts, because there is no existing protocol that declares a host is IPv4-only, dual-stack, or IPv6-only.

This document describes how a network's DHCPv4 and DHCPv6 servers, combined with a client-identifiers [RFC4361] chosen by the host, can determine if a host is IPv4-only, dual-stack, or IPv6-only, and assign the correct DNS server according to that determination.

   Note: the DHCP mechanism described in this section have some
   overlap with the Multiple Interfaces Working Group [mif] and with
   split-zone DNS [I-D.savolainen-mif-dns-server-selection].

Both an IPv4-only host and a dual-stack host obtain an IPv4 network address. Today, hosts most commonly obtain an IPv4 address using DHCPv4 [RFC2131]. An IPv6-only host does not obtain an IPv4 address; however, it may be using DHCPv6 to obtain other information (e.g., NTP servers). The following procedure takes advantage of that difference to determine if a host is IPv4-only, dual-stack, or IPv6-only.

3.5.1. Host Requirements

The host has the following requirements:

1. if the host uses IPv4, it MUST use DHCPv4 to learn its IPv4
    address and its DNS server address(es); and,

2. if the host uses IPv6, it MUST use DHCPv6 to learn its IPv6 DNS
    resolver, using the Information-Request message described in
    Section 18.1.5 of [RFC3315] and using [RFC3646]; and,

3. the host MUST use client-identifiers [RFC4361] to identify itself
    to its DHCP server(s), and MUST use the same client-identifier
    for both DHCPv4 and DHCPv6

       Note: This last requirement is stronger than the SHOULD in
       Section 6.2 of [RFC4361]

    If the host does not support DHCP authentication, and acquires/
    releases its IPv4 address while keeping its IPv6 address, it MUST
    support the procedure described in Section 3.5.4; and,

4. the host MUST support the DHCP Information Refresh Time Option
    [RFC4242].

3.5.2. DHCPv4 and DHCPv6 Server Requirements

The DHCPv4 and DHCPv6 servers have the following requirements:

1. the DHCPv4 and DHCPv6 servers MUST be able to communicate with
    each other both client-identifiers [RFC4361] and if an IPv4
    address is assigned to that client-identifier; and,

2. If the DHCP server and the host support DHCP authentication, the
    DHCP server MUST support the procedure described in
    Section 3.5.4.

3. MUST support the DHCP Information Refresh Time Option [RFC4242].

3.5.3. DHCP Server Operation

If the DHCP server first receives a DHCPv4 request for a particular
client-identifier, it responds with the 'normal' DNS resolver. The
DHCPv6 server remembers that RFC4361 client identity and if the
DHCPv6 server sees a DHCPv6 request from that same client identity,
it responds to the DHCPv6 request with a 'normal' DNS resolver.

If the DHCP server first receives a DHCPv6 request for a particular
client-identifier, it responds with a short information refresh time
[RFC4242] (e.g., 30 seconds) and a DNS64 recursive resolver.

   Note-1: This means that during the short information refresh
   time, both a dual-stack host and an IPv6-only will have their DNS
   queries processed by the DNS64 recursive resolver. During that
   time, both the dual-stack host and the IPv6-only host will get
   connectivity to IPv4 servers, but the dual-stack host will use the
   IPv6/IPv4 translator until the information refresh time expires.

   Note-2: for discussion: Consider have DHCP server slightly delay
   (e.g., 100ms) responding to a DHCPv6 request. This gives a chance
   for the DHCPv4 request to be received, thus avoiding the issue
   described in Note-1.

After the short information refresh time, the DHCPv6 client will send
a new request. By that time, the DHCPv6 server will have either:

a. have seen a DHCPv4 request from the same RFC4361 host. This
    indicates the host supports dual-stack. The DHCP server should
    extend the DHCPv6 lease, and provide a 'normal' DNS server
    (instead of the DNS64 server).

b. have not seen a DHCPv4 request from the same RFC4361 host. This
    indicates the host is IPv6-only. The DHCP server should extend

the DHCPv6 lease and continue providing the same DNS64 server.

3.5.4. Host Transition

During natural evolution of a network or because of debugging/
troubleshooting, a host might transition between IPv4-only, dual-
stack, or IPv6-only. When the host acquires or releases its IPv4
address it transitions to needing a different DNS server; if the host
has an IPv4 address, it needs a 'normal' DNS server and if it does
not have an IPv4 address it needs a DNS64 server.

There are two transitions considered, where the host transitions:

1. from IPv6-only to IPv4-supporting (that is, IPv4-only or dual-
    stack),

2. from IPv4-supporting (that is, IPv4-only or dual-stack) to IPv6-
    only.

When doing (1), the DHCPv4 server will provide a 'normal' DNS server
(because the DHCPv4 server sees the same client-identifier as seen by
the DHCPv6 server). So case (1) is solved.

However, when doing (2), the host is giving up its IPv4 address and
is currently using a normal DNS server, but needs to be told to use a
DNS64 server instead. There are two mechanisms to provide that
function, based on the network and host's support of DHCP
authentication (Section 19.1.1 of [RFC3315])

1. with DHCP authentication: When a certain client identifier loses
    or acquires its IPv4 address and also has an IPv6 address, the
    DHCPv6 server MUST send a DHCP RECONFIGURE message [RFC3315] to
    the host and SHOULD include the Option Request option indicating
    the DNS server information has changed. The RECONFIGURE message
    triggers the host to send a new Information-Request message to
    the DHCPv6 server.

2. without DHCP authentication: the host, when keeping its IPv6
    address and releasing its IPv4 address, MUST also issue a new
    DHCPv6 Information-Request message to the DHCPv6 server.

In both cases, the Information-Request message causes the DHCPv6
server to reply with a DNS64 recursive resolver, as discussed in
Section 3.5.2.

3.5.5. Advantages and Disadvantages

   Advantages:

   o Dual-stack applications, which perform DNS lookups, will
      effectively avoid NAT64 when using the 'normal' DNS server.

   Disadvantages:

   o A network with mixed IPv4-only/dual-stack hosts and IPv6-only
      hosts needs to have a mix of DNS configurations for those hosts.
      Thus, mechanisms that advertise the same DNS servers to all hosts
      cannot be used on such networks (e.g., IPv6 router
      advertisements).

   o If separate networks operate DHCPv4 and DHCPv6 (e.g., as with
      Dual-Stack Lite where the ISP operates DHCPv4 and the customer
      premise router operates DHCPv6), it is likely impossible for the
      DHCPv4 and DHCPv6 servers to communicate necessary information
      with each other.

   o Windows does not support [RFC4361].

   o OSX does not support DHCPv6.

3.6. New DHCP option for DNS64 server

   Another approach, which requires modification of IPv6-only hosts
   which need to use the DNS64, is to introduce a new DHCP option.

   The idea is to support unmodified dual-stack hosts (which use the
   normal DNS server provided via [RFC3646]), but to modify IPv6-only
   hosts to look for the DNS64 server in a newly-defined DHCPv6 option.

3.6.1. Advantages and Disadvantages

   Disadvantages:

   o Requires modifying IPv6-only hosts, and without this modification
      they won't work at all with a DNS64.

4. Security Considerations

   TBD.

5. Acknowledgements

Thanks to Mohamed Boucadair, Marcelo Braun, Ralph Droms, Dave Thaler, Bernie Volz, and Andrew Yourtchenko for their review comments.

This document was produced using version 1.35pre1 of XML2RFC.

6. IANA Considerations

This document has no actions for IANA.

7. References

7.1. Normative References

[I-D.ietf-behave-dns64]
            Bagnulo, M., Sullivan, A., Matthews, P., and I. Beijnum,
            "DNS64: DNS extensions for Network Address Translation
            from IPv6 Clients to IPv4 Servers",
            draft-ietf-behave-dns64-05 (work in progress),
            December 2009.

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
            Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2131]   Droms, R., "Dynamic Host Configuration Protocol",
            RFC 2131, March 1997.

[RFC3315]   Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C.,
            and M. Carney, "Dynamic Host Configuration Protocol for
            IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3484]   Draves, R., "Default Address Selection for Internet
            Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC3646]   Droms, R., "DNS Configuration options for Dynamic Host
            Configuration Protocol for IPv6 (DHCPv6)", RFC 3646,
            December 2003.

[RFC4242]   Venaas, S., Chown, T., and B. Volz, "Information Refresh
            Time Option for Dynamic Host Configuration Protocol for
            IPv6 (DHCPv6)", RFC 4242, November 2005.

[RFC4361]   Lemon, T. and B. Sommerfeld, "Node-specific Client
            Identifiers for Dynamic Host Configuration Protocol
            Version Four (DHCPv4)", RFC 4361, February 2006.

7.2. Informative References

[6man]   IETF, "IPv6 Maintenance Working Group", 2009,
            <http://www.ietf.org/dyn/wg/charter/6man-charter>.

[I-D.arifumi-6man-rfc3484-revise]
            Matsumoto, A., Fujisaki, T., and R. Hiromi, "Things To Be
            Considered for RFC 3484 Revision",
            draft-arifumi-6man-rfc3484-revise-02 (work in progress),
            October 2009.

[I-D.ietf-behave-address-format]
            Huitema, C., Bao, C., Bagnulo, M., Boucadair, M., and X.
            Li, "IPv6 Addressing of IPv4/IPv6 Translators",
            draft-ietf-behave-address-format-04 (work in progress),
            January 2010.

[I-D.ietf-behave-ftp64]
            Beijnum, I., "IPv6-to-IPv4 translation FTP
            considerations", draft-ietf-behave-ftp64-00 (work in
            progress), December 2009.

[I-D.savolainen-mif-dns-server-selection]
            Savolainen, T., "DNS Server Selection on Multi-Homed
            Hosts", draft-savolainen-mif-dns-server-selection-01 (work
            in progress), October 2009.

[I-D.wing-behave-learn-prefix]
            Wing, D., "Learning the IPv6 Prefix of a Network's IPv6/
            IPv4 Translator", draft-wing-behave-learn-prefix-04 (work
            in progress), October 2009.

[RFC2373]   Hinden, R. and S. Deering, "IP Version 6 Addressing
            Architecture", RFC 2373, July 1998.

[XP-DNS]   Microsoft, "Windows XP: IPv6 configuration items", 5 2005,
            <http://www.microsoft.com/resources/documentation/windows/
            xp/all/proddocs/en-us/sag_ip_v6_imp_config_items.mspx>.

[experiment]
            Braun, "practical issues with using v4-mapped addresses
            for nat64", Jul 2008, <http://www.ietf.org/mail-archive/
            web/int-area/current/msg01476.html>.

[mif]   IETF, "Multiple Interfaces Working Group", 2009,
            <http://www.ietf.org/dyn/wg/charter/mif-charter>.

Author's Address

    Dan Wing
    Cisco Systems, Inc.
    170 West Tasman Drive
    San Jose, CA 95134
    USA

    Email: dwing@cisco.com