

Behave WG
Internet-Draft
Intended status: Standards Track
Expires: July 28, 2012

T. Savolainen
Nokia
J. Korhonen
Nokia Siemens Networks
D. Wing
Cisco Systems
January 25, 2012

**Discovery of a Network-Specific NAT64 Prefix using a Well-Known Name
draft-ietf-behave-nat64-discovery-heuristic-05.txt**

Abstract

This document describes a method for detecting presence of DNS64 and for learning IPv6 prefix used for protocol translation on an access network without explicit support from the access network. The method depends on existence of a well-known IPv4-only domain name "ipv4only.arpa". The information learned enables applications and hosts to perform local IPv6 address synthesis and on dual-stack accesses avoid traversal through NAT64.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on July 28, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1. Introduction](#) [3](#)
- [2. Requirements and Terminology](#) [3](#)
 - [2.1. Requirements](#) [3](#)
 - [2.2. Terminology](#) [4](#)
- [3. Host behavior](#) [4](#)
 - [3.1. Learning NAT64 prefix securely by using DNSSEC](#) [5](#)
 - [3.1.1. Requirements for the network](#) [6](#)
 - [3.1.2. Host behavior](#) [6](#)
 - [3.2. Connectivity check](#) [7](#)
 - [3.2.1. No connectivity checks against ipv4only.arpa](#) [8](#)
 - [3.3. Alternative domain names](#) [8](#)
- [4. Operational considerations for hosting the IPv4-only well-known name](#) [9](#)
- [5. DNS\(64\) entity considerations](#) [9](#)
- [6. Exit strategy](#) [9](#)
- [7. Security Considerations](#) [9](#)
- [8. IANA Considerations](#) [10](#)
 - [8.1. About the IPv4 address for the well-known name](#) [10](#)
- [9. Acknowledgements](#) [11](#)
- [10. Normative References](#) [11](#)
- [Appendix A. Example of DNS record configuration](#) [11](#)
- [Authors' Addresses](#) [13](#)

1. Introduction

As part of the transition to IPv6 NAT64 [[RFC6146](#)] and DNS64 [[RFC6147](#)] technologies will be utilized by some access networks to provide IPv4 connectivity for IPv6-only hosts. The DNS64 utilizes IPv6 address synthesis to create local IPv6 presentations of peers having only IPv4 addresses, hence allowing DNS-using IPv6-only hosts to communicate with IPv4-only peers.

However, DNS64 cannot serve applications not using DNS, such as those receiving IPv4 address literals as referrals. Such applications could nevertheless be able to work through NAT64, provided they are able to create locally valid IPv6 presentations of peers' IPv4 addresses.

Additionally, DNS64 is not able to do IPv6 address synthesis for hosts running validating DNSSEC enabled resolvers, but instead the synthesis must be done by the hosts themselves. In order to perform IPv6 synthesis hosts have to learn the IPv6 prefix(es) used on the access network for protocol translation.

This document describes a best effort method for applications and hosts to learn the information required to perform local IPv6 address synthesis. An example application is a browser encountering IPv4 address literals in an IPv6-only access network. Another example is a host running validating security aware DNS resolver in an IPv6-only access network.

The knowledge of IPv6 address synthesis taking place may also be useful if DNS64 and NAT64 are present in dual-stack enabled access networks. In such cases hosts may choose to prefer IPv4 in order to avoid traversal through protocol translators.

It is important to notice that use of this approach will not result in as robust and good behaving system as an all-IPv6 system would be. Hence it is highly RECOMMENDED to upgrade to IPv6 and utilize the described method only as a short-term solution.

2. Requirements and Terminology

2.1. Requirements

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](#)].

2.2. Terminology

Well-Known IPv4-only Name (WKN): a fully qualified domain name, "ipv4only.arpa", well-known to have only A record.

Well-Known IPv4 Address: an IPv4 address that is well-known and mapped to the well-known name.

3. Host behavior

A host requiring information about presence of NAT64 and the IPv6 prefix used for protocol translation SHALL send a DNS query for AAAA records of a well-known IPv4-only fully qualified domain name: "ipv4only.arpa". The host MAY also need to perform DNS query for the A record of the well-known name in order to learn what is the IPv4 address of the well-known name. The host may perform this check in both IPv6-only and dual-stack access networks.

When sending AAAA query for the well-known name a host MUST set "Checking Disabled (CD)" bit to zero, as otherwise the DNS64 will not perform IPv6 address synthesis hence does not reveal the IPv6 prefix(es) used for protocol translation.

A DNS reply with one or more non-empty AAAA records indicates that the access network is utilizing IPv6 address synthesis. A host MUST look through all of the received AAAA records to collect all available prefixes. The prefixes may include Well-Known Prefix 64: ff9b::/96 [[RFC6052](#)] or one or more Network-Specific Prefixes. In the case of NSPs the host SHALL search for the IPv4 address of the well-known name inside of the received IPv6 addresses to determine the used address format.

An IPv4 address of the well-known name should be found inside synthetic IPv6 address at some of the locations described in [[RFC6052](#)]. If the searched IPv4 address is not found on any of the standard locations the network must be using different formatting. Developers may over time learn on IPv6 translated address formats that are extensions or alternatives to the standard formats. Developers MAY at that point add additional steps to the described discovery procedures. The additional steps are outside the scope of the present document.

The host should ensure a 32-bit IPv4 address value is present only once in an IPv6 address. In case another instance of the value is found inside the IPv6, the host shall repeat the search with another IPv4 address, if possible.

In the case only one IPv6 prefix was present in the DNS response: a host shall use that IPv6 prefix for both local synthesis and for detecting synthesis done by the DNS64 entity on the network.

In the case multiple IPv6 prefixes were present in the DNS response: a host SHOULD use all received prefixes when determining whether other received IPv6 addresses are synthetic. However, for selecting prefix for the local IPv6 address synthesis host MUST use the following prioritization order, of which purpose is to avoid use of prefixes containing suffixes reserved for the future [[RFC6052](#)]:

1. Use NSP having /96 prefix
2. Use WKP prefix
3. Use longest available NSP prefix

In the case of NXDOMAIN response or an empty AAAA reply: the DNS64 is not available on the access network, network filtered the well-known query on purpose, or something went wrong in the DNS resolution. All unsuccessful cases result in unavailability of a host to perform local IPv6 address synthesis. The host MAY periodically resend AAAA query to check if DNS64 has become available or possibly temporary problem cleared. The host MAY perform A query for the well-known name to learn whether the NAT64 prefix discovery framework is available at all (see [section 6](#) about Exit Strategy). The host MAY also continue monitoring DNS replies with IPv6 addresses constructed from WKP, in which case the host MAY use the WKP as if it were learned during the query for well-known name.

To save Internet's resources, if possible, a host should perform NAT64 discovery only when needed (e.g. when local synthesis is required, cached reply timeouts, new network interface is started, and so forth. Furthermore, the host SHOULD cache the replies it receives and honor TTLs.

3.1. Learning NAT64 prefix securely by using DNSSEC

If a node is using untrusted channel between itself and DNS64, or DNS64 entity itself is untrusted, it is possible for an attacker to influence node's NAT64 prefix detection procedures. This may result in denial-of-service, redirection, man-in-the-middle, or other attacks. To protect against these attacks the node may use DNSSEC, or communicate with trusted DNS64 over trusted channel. Significantly, DNSSEC must be configured by the NAT64 operator for the DNSSEC approach to work.

3.1.1. Requirements for the network

To support DNSSEC capable nodes to perform NAT64 prefix learning securely, the operator of the NAT64 device MUST perform the following configurations. In the case of multiple IPv6 prefixes being used in a network, e.g. for load-balancing purposes, it is for network administrators to decide whether a single NAT64's fully qualified domain name maps to multiple prefixes, or whether there will be dedicated FQDN per IPv6 prefix.

1. Have one or more fully qualified domain names for the NAT64 translators (NAT64 FQDN).
2. Each NAT64 FQDN MUST have one or more DNS AAAA resource records with each IPv6 address consisting of NAT64 prefix and 0's for the elements after the actual NAT64 prefix. Also, for the connectivity test each NAT64 FQDN MUST have one or more DNS A resource records with IPv4 address(es) of NAT64 Internet facing interfaces.
3. Each NAT64 prefix MUST HAVE PTR record that points to corresponding NAT64 FQDN.
4. Sign the NAT64 FQDNs' AAAA and A records with DNSSEC.
5. Have access network's authoritative nameservers to respond to DNS queries for the NAT64 FQDNs only when the queries have been originated from the network domain the NAT64 is serving. If the NAT64's AAAA records are made resolvable throughout the Internet, a possible misuse vector of the NAT64 prefixes and NAT64 FQDNs in other networks is enabled: an attacker in other access network may lure a host on that network to think it is configuring NAT64 prefix in secure manner, while in reality it is not as the node would be configuring NAT64 prefix in a network where the NAT64 prefix should not be used.

3.1.2. Host behavior

A DNSSEC capable node MUST use the following procedure to confirm the learned NAT64 prefix is legitimate:

1. Heuristically find out a NAT64 prefix candidate by making AAAA query for the "ipv4only.arpa" by following the procedure in [Section 3](#). This will return one or more AAAA resource records. For each of those AAAA resource records host wishes to use securely, the host performs the following steps.

2. Send DNS PTR query for the IPv6 address of the translator (for "ipv6.arpa"), using the prefix from the step 1 and 0 for the elements after the prefix length. This will return the NAT64 FQDN.
3. Send DNS AAAA query for the NAT64 FQDN.
4. Verify the DNS AAAA response matches the address obtained in step 1. It is possible that the NAT64 FQDN maps to multiple AAAA records, in which case the node has to check if any of the responses matches to the address obtained in step 1. The node must ignore other responses and not to use those for local IPv6 address synthesis.
5. Perform DNSSEC validation of the DNS AAAA response.

After the node has successfully performed the above five steps, the node can consider NAT64 prefix securely learned.

3.2. Connectivity check

After learning a NAT64 prefix, it can be useful to determine if that learned prefix is functional. It could be non-functional for a variety of reasons -- the heuristic failed to work as expected, the IPv6 path to the NAT64 is down, the NAT64 is down, or the IPv4 path beyond the NAT64 is down.

There are two general approaches to determine if the learned prefix is functional. The first approach is to perform a separate connectivity check. The second approach is to attempt to use the learned prefix for normal traffic. Each approach has some trade-offs (e.g., additional network traffic or possible user-noticeable delay), and implementations should carefully weigh which approach is appropriate for their application and the network.

The host MAY perform separate connectivity check by sending an ICMPv6 Echo Request to IPv6 address synthesized by combining discovered NAT64 prefix with an IPv4 address of the server used for the connectivity check. This will test the IPv6 path to the NAT64 and the NAT64's operation. It will not test the IPv4 path beyond the NAT64.

To perform connectivity check, the host does a PTR query of the NAT64's IPv6 prefix which returns a hostname. The host then does an A query of that hostname, which returns one or more A resource records, which are the IPv4-facing IP addresses of that NAT64. The host chooses the first one of these addresses and sends an ICMPv6 Echo Request to that address. If no response is received, the host

sends another ICMPv6 Echo Request, a second later. If still no response is received, it sends a third ICMPv6 Echo Request 3 seconds later. As the NAT64 will respond to ICMP Echo Requests sent to any of its IPv4 addresses, there is no purpose in attempting to send ICMP Echo Requests to the other IPv4 addresses. If an ICMPv6 Echo Response is received, the host knows the IPv6 path to the NAT64 and the NAT64 is functioning normally. If, after the three transmissions of the ICMPv6 Echo Request, no response is received, the host knows this NAT64 prefix is not functioning, and MAY choose a different NAT64 (if a different NAT64 is available) or choose to alert the user.

Alternatively, or additionally, the host MAY use implementation specific server other than the NAT64 for the connectivity check. This implementation specific server can be used as fallback server if the NAT64 does not respond or does not have A record.

To help hosts' connectivity checks NAT64 operators are RECOMMENDED to maintain DNS AAAA and A records for the NAT64 FQDN as is described in [Section 3.1.1](#) step 1, independently of possible DNSSEC support.

3.2.1. No connectivity checks against ipv4only.arpa

Clients MUST NOT send a connectivity check to the address returned in the ipv4only.arpa query. This is because, by design, no server will be operated on the Internet at that address as such. Similarly, network operators MUST NOT operate a server on that address. The reason this address isn't used for connectivity checks is that operators who neglect to operate a connectivity check server will allow that traffic towards the Internet where it will be dropped and cause a false negative connectivity check with the client (that is, the NAT64 is working fine, but the connectivity check fails because a server is not operating at ipv4-only.arpa on the Internet and a server is not operated by the NAT64 operator). Instead, for the connectivity check, an additional DNS resource record is looked up (specifically, the A record associated with the NAT64's hostname) and used for the connectivity check. This ensures that packets don't unnecessarily leak to the Internet and reduces the chance of a false negative connectivity check.

3.3. Alternative domain names

Some applications, operating systems, devices, or networks may find it advantageous to operate their own DNS infrastructure to perform a function similar to ipv4-only.arpa, but using a different resource record. The primary advantage is to ensure availability of the DNS infrastructure and ensure the proper configuration of the DNS record itself. DNS For example, a company named Example might have their

application query `ipv4-only.example.com`. Other than the different DNS resource record being queried, the rest of the operations are anticipated to be identical to the steps described in this document.

4. Operational considerations for hosting the IPv4-only well-known name

The authoritative name server for the well-known name shall have DNS record TTL set to a long value in order to improve effectiveness of DNS caching and robustness of the discovery procedure in general. The exact value depends on availability time for the used public IPv4 address, but should not be longer than one year.

The domain serving the well-known name must be signed with DNSSEC. See also Security Considerations section.

It is expected that volumes for well-known name related queries are roughly SOMETHING, TBD. The infrastructure required to serve well-known name is SOMETHING, TBD.

5. DNS(64) entity considerations

DNS(64) servers MUST NOT interfere or perform special procedures for the queries related to the well-known name until the time has arrived for the exit strategy to be deployed.

6. Exit strategy

A day will come when this tool is no longer needed. At that point best suited techniques for implementing exit strategy will be documented. In the global scope the exit strategy may include sending NXDOMAIN replies by the authoritative name server of the well-known name with a very long TTL.

A client implementation receiving NXDOMAIN response for the A query of the well-known name means SHOULD consider this tool as temporarily disabled.

7. Security Considerations

The security considerations follow closely those of [RFC6147](#) [[RFC6147](#)]. If an attacker manages to change the NAT64 prefix host discovers, the traffic generated by the host will be delivered to altered destination. This can result in either a denial-of-service (DoS) attack (if the resulting IPv6 addresses are not assigned to any

device), a flooding attack (if the resulting IPv6 addresses are assigned to devices that do not wish to receive the traffic), or an eavesdropping attack (in case the altered NSP is routed through the attacker).

The zone serving the well-known name has to be protected with DNSSEC, as otherwise it will be too attractive target for attackers who wish to alter hosts' NSP prefix discovery procedures.

A host SHOULD implement validating DNSSEC resolver for validating the A response of the well-known name query. A host without validating DNSSEC resolver SHOULD request validation to be performed by the used recursive DNS server.

For the secure NAT64 prefix discovery the access network SHOULD sign the NAT64 translator's fully qualified domain name, and make that DNS resolvable only from the network domain NAT64 is serving. Otherwise the NAT64 prefix may be used for attacks in other access networks. A host SHOULD use the algorithm described in [Section 3.1](#) in order to securely learn the NAT64 prefix.

8. IANA Considerations

According to procedures described in [RFC3172](#) this document requests IANA and IAB to reserve a second level domain from the .ARPA zone for the well-known domain name. The well-known domain name could be, for example, "ipv4only.arpa".

The well-known name also needs to map to one but preferably to two different public IPv4 addresses.

8.1. About the IPv4 address for the well-known name

The IPv4 address for the well-known name, if possible, should be chosen so that it is unlikely to appear more than once within an IPv6 address and also as easy as possible to find from within the synthetic IPv6 address. An address not listed in the [Section 3 of \[RFC5735\]](#) is required as otherwise DNS64 entity may not perform AAAA record synthesis. The address does not have to be routable or allocated to any node, as no communications are initiated to the IPv4 address.

Allocating two IPv4 addresses would improve the heuristics in cases where the primary IPv4 address' bit pattern appears more than once in the synthetic IPv6 address (NSP prefix contains the same bit pattern as the IPv4 address).

If no well-known IPv4 address is statically allocated for this method, the heuristic requires sending additional A query to learn the IPv4 address that is sought inside the received IPv6 address. Without knowing IPv4 address it is impossible to determine address format used by DNS64.

9. Acknowledgements

Authors would like to thank Andrew Sullivan, Washam Fan, Cameron Byrne, Zhenqiang Li, Dave Thaler, Peter Koch, and Christian Huitema for significant improvement ideas and comments.

10. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC5735] Cotton, M. and L. Vegoda, "Special Use IPv4 Addresses", [BCP 153](#), [RFC 5735](#), January 2010.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", [RFC 6052](#), October 2010.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", [RFC 6146](#), April 2011.
- [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", [RFC 6147](#), April 2011.

Appendix A. Example of DNS record configuration

The following BIND-style examples illustrate how A and AAAA records could be configured by NAT64 operator.

The examples use NAT64 prefix of 2001:db8::/96 and example.com domain.

The PTR record for reverse queries ([Section 3.1.1](#) bullet 3):

```
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.IP6.ARPA.  
@      IN      SOA    ns1.example.com. hostmaster.example.com. (  
                2003080800 12h 15m 3w 2h)  
      IN      NS     ns.example.com.  
  
      IN      PTR    nat64.example.com.
```

If the example.com does not use DNSSEC, the following configuration file could be used. Please note the nat64.example.com has both AAAA record with the NAT64 prefix and A record for the connectivity check ([Section 3.1.1](#) bullet 2).

```
example.com.  IN SOA  ns.example.com. hostmaster.example.com. (  
                2002050501 ; serial  
                100      ; refresh (1 minute 40 seconds)  
                200      ; retry (3 minutes 20 seconds)  
                604800   ; expire (1 week)  
                100      ; minimum (1 minute 40 seconds)  
                )
```

```
example.com.  IN NS   ns.example.com.
```

```
nat64.example.com.
```

```
      IN AAAA  2001:db8:0:0:0:0:0:0 nat64.example.com.  
      IN A    192.0.2.1
```

If the example.com does use DNSSEC, the following configuration file could be used for A and AAAA records:

```
example.com.  IN SOA  ns.example.com. hostmaster.example.com. (
    2002050501 ; serial
    100        ; refresh (1 minute 40 seconds)
    200        ; retry (3 minutes 20 seconds)
    604800    ; expire (1 week)
    100        ; minimum (1 minute 40 seconds)
)

example.com.  IN RRSIG SOA  5 2 100 20090803071330 (
    20090704071330 17000 example.com.
    TVgWsNQvsFmeNHAeccGi7+UI7KwcE9TXPuSvmV9yyJwo
    4FvHkxVC1H+98EtrmbR4c/XcdUzdfgn+q+lBqNsnbAit
    xFERwPxzxbX0+yeCdHbBjHe70u0c2Gc+CH6SbT2lKwVi
    iEx3ySqqNoVScoUyhRdnPV2A1LV0yd9GtG9mI4w= )

example.com.  IN NS  ns.example.com.
example.com.  IN RRSIG NS  5 2 100 20090803071330 (
    20090704071330 17000 example.com.
    Xuw7saDDi6+5Z7SmtC7FC2npP0iE8F9qMR87eA0egG0I
    B+xFx7pIogoVIDp0d1h3jqYivhblpCoDSBQb2oMbVy3B
    SX5cF0r7Iu/xKP8XrV4DjNiugpa+NnhEiARqG5uoPFbX
    4cYT51yNq70I5mJvvajJu7UjmdHl26ZlnK33xps= )

nat64.example.com.
    IN AAAA  2001:db8:0:0:0:0:0 nat64.example.com.
    IN RRSIG SOA  5 2 100 20090803071330 (
    20090704071330 17000 example.net.
    TVgWsNQvsFmeNHAeccGi7+UI7KwcE9TXPuSvmV9yyJwo
    4FvHkxVC1H+98EtrmbR4c/XcdUzdfgn+q+lBqNsnbAit
    xFERwPxzxbX0+yeCdHbBjHe70u0c2Gc+CH6SbT2lKwVi
    iEx3ySqqNoVScoUyhRdnPV2A1LV0yd9GtG9mI4w= )

nat64.example.com.
    IN A  192.0.2.1
```

Authors' Addresses

Teemu Savolainen
Nokia
Hermiankatu 12 D
FI-33720 Tampere
Finland

Email: teemu.savolainen@nokia.com

Jouni Korhonen
Nokia Siemens Networks
Linnoitustie 6
FI-02600 Espoo
Finland

Email: jouni.nospam@gmail.com

Dan Wing
Cisco Systems
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com