

Network Working Group
Internet-Draft
Updates: [7050](#) (if approved)
Intended status: Standards Track
Expires: November 23, 2017

S. Cheshire
D. Schinazi
Apple Inc.
May 22, 2017

Special Use Domain Name 'ipv4only.arpa'
draft-cheshire-sudn-ipv4only-dot-arpa-07

Abstract

The specification for how a client discovers its network's NAT64 prefix [[RFC7050](#)] defines the special name 'ipv4only.arpa' for this purpose, but declares it to be a non-special name in that specification's Domain Name Reservation Considerations section.

Consequently, despite the well articulated special purpose of the name, (at the time of writing) 'ipv4only.arpa' still does not appear as one of the names with special properties recorded in the Special-Use Domain Names registry.

This document formally declares the actual special properties of the name, and adds similar declarations for the corresponding reverse mapping names.

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 November 23, 2017.

Copyright Notice

Copyright (c) 2017 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.

1. Introduction

The specification for how a client discovers its network's NAT64 prefix [[RFC7050](#)] defines the special name 'ipv4only.arpa' for this purpose, but declares it to be a non-special name in that specification's Domain Name Reservation Considerations section.

Consequently, despite the well articulated special purpose of the name, (at the time of writing) 'ipv4only.arpa' still does not appear as one of the names with special properties recorded in the Special-Use Domain Names registry [[SUDN](#)].

This document formally declares the actual special properties of the name. This document also adds similar declarations for the corresponding reverse mapping names.

2. Specialness of 'ipv4only.arpa'

The hostname 'ipv4only.arpa' is peculiar in that it was never intended to be treated like a normal hostname.

A typical client never looks up the IPv4 address records for 'ipv4only.arpa', because it is already known, by specification [[RFC7050](#)], to have exactly two IPv4 address records, 192.0.0.170 and 192.0.0.171. No client ever has to look the name in order to learn those two addresses.

In contrast, clients often look up the IPv6 AAAA address records for 'ipv4only.arpa', which is contrary to general DNS expectations, given that it is already known, by specification [[RFC7050](#)], that no such IPv6 AAAA address records exist. And yet, clients expect to receive, and do in fact receive, positive answers for these IPv6 AAAA address records that are known to not exist.

This is clearly not a typical DNS name. In normal operation, clients never query for the two records that do in fact exist; instead they query for records that are known to not exist, and then get positive answers to those abnormal queries. Clients are using DNS to perform queries for this name, but they are certainly not using DNS to learn legitimate answers from the name's legitimate authoritative server. Instead, these clients have, in effect, co-opted the DNS protocol as an impromptu client-to-middlebox communication protocol, to communicate with the NAT64/DNS64 [[RFC6146](#)][[RFC6147](#)] gateway, if present, and request that it disclose the prefix it is using for IPv6 address synthesis.

It is this use of specially-crafted DNS queries as an impromptu client-to-middlebox communication protocol that makes the name 'ipv4only.arpa' most definitely a special name, and one that should be listed in IANA's registry along with other DNS names that have special uses [[SUDN](#)].

3. Consequences of 'ipv4only.arpa' previously being declared unspecial

As a result of the original specification [[RFC7050](#)] not formally declaring 'ipv4only.arpa' to have special properties, there was no mandate for any DNS software to treat this name specially. Consequently, queries for this name had to be handled normally, resulting in unnecessary queries to the authoritative 'arpa' name servers.

Having millions of devices around the world issue these queries generated pointless additional load on the authoritative 'arpa' name servers, which was completely unnecessary when the name 'ipv4only.arpa' is defined, by Internet Standard, to have exactly two IPv4 address records, 192.0.0.170 and 192.0.0.171, and no other records of any type.

Also, at times, for reasons that are as yet unclear, the authoritative 'arpa' name servers have been observed to be slow or unresponsive. The failures of these 'ipv4only.arpa' queries result in unnecessary failures of software that depends on them for DNS64 [[RFC6147](#)] address synthesis.

Even when the authoritative 'arpa' name servers are operating correctly, having to perform an unnecessary query to obtain an answer that is already known in advance can add precious milliseconds of delay for no reason.

A more serious problem occurs when a device is configured to use a recursive/caching DNS server other than the one it learned from the network. Typically a device joining a NAT64 network will learn the recursive/caching DNS server recommended for that network either via IPv6 Router Advertisement Options for DNS Configuration [[RFC6106](#)] or via DNS Configuration options for DHCPv6 [[RFC3646](#)]. On a NAT64 network it is essential that the client use the recursive/caching DNS64 server recommended for that network, since only that DNS64 server can be relied upon to know the appropriate prefix(es) to use for synthesizing IPv6 addresses that will be acceptable to the NAT64 server.

However, it is not uncommon for users to manually override their default DNS configuration because they wish to use some other public recursive resolver on the Internet, perhaps because they perceive their preferred recursive resolver to be faster, more reliable, or more trustworthy.

Another common scenario is the use of corporate VPN client software, which overrides the default configuration to divert DNS requests to the company's own private internal recursive resolver, because the

local network's recursive resolver will typically be unable to provide answers for the company's private internal host names. Similarly, the company's private internal recursive resolver may not be able to synthesize IPv6 addresses correctly for use with the local network's NAT64 server, because it is unlikely to be aware of the NAT64 prefix in use on the local network. It is clear that a single recursive resolver cannot meet both needs. The local network's recursive resolver cannot give answers for some company's private internal host names, and some company's private internal recursive resolver cannot give correctly synthesized IPv6 addresses suitable for the local network's NAT64 gateway.

The conflict here arises because DNS is being used for two unrelated purposes. The first purpose is retrieving data from a (nominally) global database -- generally retrieving the IP address(es) associated with a hostname. The second purpose is using the DNS protocol as a middlebox communication protocol, to interrogate the local network infrastructure to discover the IPv6 prefix(es) in use by the local NAT64 gateway for address synthesis.

(Possibly this problem could be solved if we could force all NAT64 gateways to use the same Well-Known Prefix for IPv6 address synthesis [[RFC6052](#)], but that would alleviate the need for 'ipv4only.arpa' altogether.)

This document leverages this operational experience to update the Domain Name Reservation Considerations section [[RFC6761](#)] of the earlier specification [[RFC7050](#)] with one that accurately lists the actual special properties of the name 'ipv4only.arpa' so that software can legitimately make appropriate performance and reliability optimizations.

4. Security Considerations

Hard-coding the known answers for 'ipv4only.arpa' queries in recursive/caching DNS servers reduces the risk of malicious devices intercepting those queries and returning incorrect answers, particularly in the case of recursive/caching DNS servers that do not perform DNSSEC validation.

One of the known concerns with DNS64 [[RFC6147](#)] is that it interferes with DNSSEC. DNSSEC may cryptographically assert that a name has no IPv6 AAAA records, while at the same time DNS64 address synthesis is contradicting this and claiming that IPv6 AAAA records do exist.

[Section 3](#) of the DNS64 specification [[RFC6147](#)] discusses this:

... DNS64 receives a query with the DO bit set and the CD bit set. In this case, the DNS64 is supposed to pass on all the data it gets to the query initiator. This case will not work with DNS64, unless the validating resolver is prepared to do DNS64 itself.

The NAT64 Prefix Discovery specification [[RFC7050](#)] provides the mechanism for the query initiator to learn the NAT64 prefix so that it can do its own validation and DNS64 synthesis as described above. With this mechanism the client can (i) interrogate the local NAT64/DNS64 gateway with an 'ipv4only.arpa' query to learn the IPv6 address synthesis prefix, (ii) query for the (signed) IPv4 address records itself, and then (iii) perform its own IPv6 address synthesis locally, combining the IPv6 address synthesis prefix learned from the local NAT64/DNS64 gateway with the secure DNSSEC-signed data learned from the global Domain Name System.

It is conceivable that over time, if DNSSEC is successful, the majority of clients could move to this validate-and-synthesize-locally model, which reduces the DNS64 machinery to the vestigial role of simply responding to the 'ipv4only.arpa' query to report the local IPv6 address synthesis prefix. In no case does the client care what answer(s) the authoritative 'arpa' name servers might give for that query. The 'ipv4only.arpa' query is being used purely as a local client-to-middlebox communication message.

This approach is even more attractive if it does not create an additional dependency on the authoritative 'arpa' name servers to answer a query that is unnecessary because the NAT64/DNS64 gateway already knows the answer before it even issues the query. Avoiding this unnecessary query improves performance and reliability for the client, and reduces unnecessary load for the authoritative 'arpa' name servers.

5. IANA Considerations

[Once published, this should say]

IANA has recorded the following names in the Special-Use Domain Names registry [[SUDN](#)]:

```
ipv4only.arpa.  
170.0.0.192.in-addr.arpa.  
171.0.0.192.in-addr.arpa.
```

IANA has recorded the following IPv4 addresses in the IPv4 Special-Purpose Address Registry [[SUV4](#)]:

```
192.0.0.170  
192.0.0.171
```

6. Domain Name Reservation Considerations

6.1. Conventions and Terminology Used in this Section

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this section are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [[RFC2119](#)].

6.2. ipv4only.arpa

The name 'ipv4only.arpa' is defined, by Internet Standard, to have two IPv4 address records with rdata 192.0.0.170 and 192.0.0.171.

When queried via a DNS64 [[RFC6147](#)] recursive/caching server, the name 'ipv4only.arpa' is also defined to have IPv6 AAAA records, with rdata synthesized from a combination of the NAT64 IPv6 prefix(es), and the IPv4 addresses 192.0.0.170 and 192.0.0.171. This can return more than one pair of IPv6 addresses if there are multiple NAT64 prefixes.

The name 'ipv4only.arpa' has no other DNS records of any type. There are no subdomains of ipv4only.arpa. All names falling below 'ipv4only.arpa' are defined to be nonexistent (NXDOMAIN).

The name 'ipv4only.arpa' is special to

- (a) client software wishing to perform DNS64 address synthesis,
- (b) APIs responsible for retrieving the correct information, and
- (c) the DNS64 recursive/caching server responding to such requests.

These three considerations are listed in items 2, 3 and 4 below:

1. Normal users should never have reason to encounter the 'ipv4only.arpa' domain name. If they do, they should expect queries for 'ipv4only.arpa' to result in the answers required by the specification [[RFC7050](#)]. Normal users have no need to know that 'ipv4only.arpa' is special.
2. Application software may explicitly use the name 'ipv4only.arpa' for NAT64/DNS64 address synthesis, and expect to get the answers required by the specification [[RFC7050](#)]. If application software encounters the name 'ipv4only.arpa' in the normal course of handling user input, the application software should resolve that name as usual and need not treat it in any special way.
3. Name resolution APIs and libraries MUST recognize 'ipv4only.arpa' as special and MUST give it special treatment. Regardless of any manual client DNS configuration, DNS overrides configured by VPN client software, or any other mechanisms that influence the choice of the client's recursive/caching DNS server address(es) (including client devices that run their own local recursive resolver and use the loopback address as their configured recursive/caching DNS server address) all queries for 'ipv4only.arpa' and any subdomains of that name MUST be sent to the recursive/caching DNS server learned from the network via IPv6 Router Advertisement Options for DNS Configuration [[RFC6106](#)] or via DNS Configuration options for DHCPv6 [[RFC3646](#)]. Because DNS queries for 'ipv4only.arpa' are actually a special middlebox communication protocol, it is essential that they go to the middlebox in question, and failure to honor this requirement would cause failure of the NAT64 Prefix Discovery mechanism [[RFC7050](#)].
4. For the purposes of this section, recursive/caching DNS servers fall into two categories. The first category is the traditional recursive/caching DNS servers that are in widespread use today. The second category is DNS64 servers, whose purpose is to synthesize IPv6 address records.

Traditional recursive/caching DNS servers SHOULD NOT recognize 'ipv4only.arpa' as special or give that name, or subdomains of that name, any special treatment. The rationale for this is that a traditional recursive/caching DNS server, such as built in to a home gateway, may itself be downstream of a DNS64 server.

Passing though the 'ipv4only.arpa' queries to the upstream DNS64 server will allow the correct NAT64 prefix to be discovered.

All DNS64 servers MUST recognize 'ipv4only.arpa' as special and MUST NOT attempt to look up NS records for it, or otherwise query authoritative DNS servers in an attempt to resolve this name. Instead, DNS64 servers MUST act as authoritative for this domain and generate immediate responses for all such queries.

DNS64 servers MUST generate the 192.0.0.170 and 192.0.0.171 responses for IPv4 address queries (DNS qtype "A"), the appropriate synthesized IPv6 address record responses for IPv6 address queries (DNS qtype "AAAA"), and a negative ("no error no answer") response for all other query types.

For all subdomains of 'ipv4only.arpa', DNS64 servers MUST generate immediate NXDOMAIN responses. All names falling below 'ipv4only.arpa' are defined to be nonexistent.

An example configuration for BIND 9 showing how to achieve the desired result is given in [Appendix A](#).

5. Traditional authoritative DNS server software need not recognize 'ipv4only.arpa' as special or handle it in any special way. Recursive/caching DNS servers SHOULD routinely act as authoritative for this name and return the results described above. Only the administrators of the 'arpa' namespace need to explicitly configure their actual authoritative name servers to be authoritative for this name and to generate the appropriate answers; all other authoritative name servers will not be configured to know anything about this name and will reject queries for it, as they would reject queries for any other name about which they have no information.
6. Generally speaking, operators of authoritative DNS servers need not know anything about the name 'ipv4only.arpa', just as they do not need to know anything about any other names they are not responsible for. Operators of authoritative DNS servers who are configuring their name servers to be authoritative for this name MUST understand that 'ipv4only.arpa' is a special name, with records rigidly specified by Internet Standard (generally this applies only to the administrators of the 'arpa' namespace).
7. DNS Registries/Registrars need not know anything about the name 'ipv4only.arpa', just as they do not need to know anything about any other name they are not responsible for. Only the administrators of the 'arpa' namespace need to be aware of this name's purpose and how it should be configured.

6.3. 170.0.0.192.in-addr.arpa and 171.0.0.192.in-addr.arpa

Since the IPv4 addresses 192.0.0.170 and 192.0.0.171 are defined to be special, and are listed in the IPv4 Special-Purpose Address Registry [[SUVv4](#)], the corresponding reverse mapping names in the in-addr.arpa domain are similarly special.

The name '170.0.0.192.in-addr.arpa' is defined, by Internet Standard, to have only a single DNS record, type PTR, with rdata 'ipv4only.arpa'.

The name '171.0.0.192.in-addr.arpa' is defined, by Internet Standard, to have only a single DNS record, type PTR, with rdata 'ipv4only.arpa'.

There are no subdomains of '170.0.0.192.in-addr.arpa' or '171.0.0.192.in-addr.arpa'. All names falling below these names are defined to be nonexistent (NXDOMAIN).

Practically speaking these two names are rarely used, but to the extent that they may be, they are special only to recursive/caching DNS servers as described in item 4 below:

1. Normal users should never have reason to encounter these two reverse mapping names. However, if they do, queries for these reverse mapping names should return the expected answer 'ipv4only.arpa'. Normal users have no need to know that these reverse mapping names are special.
2. Application software SHOULD NOT recognize these two reverse mapping names as special, and SHOULD NOT treat them differently. For example, if the user were to issue the Unix command "host 192.0.0.170" then the "host" command should issue the query as usual and display the result that is returned.
3. Name resolution APIs and libraries SHOULD recognize these two reverse mapping names as special and generate the required responses locally. For the names '170.0.0.192.in-addr.arpa' and '171.0.0.192.in-addr.arpa' PTR queries yield the result 'ipv4only.arpa'; all other query types yield a negative ("no error no answer") response. For all subdomains of these two reverse mapping domains, all queries yield an NXDOMAIN response. All names falling below these two reverse mapping domains are defined to be nonexistent.

This local self-contained generation of these responses is to avoid placing unnecessary load on the authoritative 'in-addr.arpa' name servers.

4. Recursive/caching DNS servers SHOULD NOT recognize these two reverse mapping names as special and SHOULD NOT, by default, give them any special treatment.
5. Traditional authoritative DNS server software need not recognize these two reverse mapping names as special or handle them in any special way.
As a practical matter, only the administrators of the '192.in-addr.arpa' namespace will configure their name servers to be authoritative for these names and to generate the appropriate answers; all other authoritative name servers will not be configured to know anything about these names and will reject queries for them as they would reject queries for any other name about which they have no information.
6. Generally speaking, operators of authoritative DNS servers need not know anything about these two reverse mapping names, just as they do not need to know anything about any other names they are not responsible for. Operators of authoritative DNS servers who are configuring their name servers to be authoritative for this name MUST understand that these two reverse mapping names are special, with answers specified by Internet Standard (generally this applies only to the administrators of the '192.in-addr.arpa' namespace).
7. DNS Registries/Registrars need not know anything about these two reverse mapping names, just as they do not need to know anything about any other name they are not responsible for. Only the administrators of the '192.in-addr.arpa' namespace need to be aware of the purpose of these two names.

6.3.1. ip6.arpa Reverse Mapping PTR Records

For all IPv6 addresses synthesized by a DNS64 recursive resolver, the DNS64 recursive resolver server is responsible for synthesizing the appropriate 'ip6.arpa' reverse mapping PTR records too, if it chooses to provide reverse mapping PTR records. The same applies to the synthesized IPv6 addresses corresponding to the IPv4 addresses 192.0.0.170 and 192.0.0.171.

Generally a DNS64 recursive/caching server synthesizes appropriate 'ip6.arpa' reverse mapping PTR records by extracting the embedded IPv4 address from the encoded IPv6 address, performing a reverse mapping PTR query for that IPv4 address, and then synthesizing a corresponding 'ip6.arpa' reverse mapping PTR record containing the same rdata.

In the case of synthesized IPv6 addresses corresponding to the IPv4 addresses 192.0.0.170 and 192.0.0.171, the DNS64 recursive/caching server does not issue reverse mapping queries for those IPv4 addresses, but instead, according to rule 3 above, immediately returns the answer 'ipv4only.arpa'.

In the case of a client that uses the 'ipv4only.arpa' query to discover the IPv6 prefixes in use by the local NAT64 gateway, and then proceeds to perform its own address synthesis locally (which has benefits such as allowing DNSSEC validation), that client MUST also synthesize 'ip6.arpa' reverse mapping PTR records for those discovered prefix(es), according to the rules above: When a client's name resolution APIs and libraries receive a request to look up an 'ip6.arpa' reverse mapping PTR record for an address that falls within one of the discovered NAT64 address synthesis prefixes, the software extracts the embedded IPv4 address and then, for IPv4 addresses 192.0.0.170 and 192.0.0.171, returns the fixed answer 'ipv4only.arpa', and for all other IPv4 addresses performs a reverse mapping PTR query for the IPv4 address, and then synthesizes a corresponding 'ip6.arpa' reverse mapping PTR record containing the same rdata.

7. Acknowledgements

Thanks to Jouni Korhonen, Teemu Savolainen, and Dan Wing, for devising the NAT64 Prefix Discovery mechanism [[RFC7050](#)], and for their feedback on this document. Thanks to Geoff Huston for his feedback on the draft, and to Erik Kline for pointing out that the in-addr.arpa names are special too. Thanks particularly to Lorenzo Colitti for an especially spirited hallway discussion at IETF 96 in Berlin, which lead directly to significant improvements in how this document presents the issues.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3646](#), DOI 10.17487/RFC3646, December 2003, <<http://www.rfc-editor.org/info/rfc3646>>.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", [RFC 6052](#), DOI 10.17487/RFC6052, October 2010, <<http://www.rfc-editor.org/info/rfc6052>>.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", [RFC 6106](#), DOI 10.17487/RFC6106, November 2010, <<http://www.rfc-editor.org/info/rfc6106>>.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", [RFC 6146](#), DOI 10.17487/RFC6146, April 2011, <<http://www.rfc-editor.org/info/rfc6146>>.
- [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](#), DOI 10.17487/RFC6147, April 2011, <<http://www.rfc-editor.org/info/rfc6147>>.
- [RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", [RFC 6761](#), DOI 10.17487/RFC6761, February 2013, <<http://www.rfc-editor.org/info/rfc6761>>.
- [RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", [RFC 7050](#), DOI 10.17487/RFC7050, November 2013, <<http://www.rfc-editor.org/info/rfc7050>>.

8.2. Informative References

- [SUDN] "Special-Use Domain Names Registry",
<<https://www.iana.org/assignments/special-use-domain-names/>>.
- [SUV4] "IANA IPv4 Special-Purpose Address Registry",
<<https://www.iana.org/assignments/iana-ipv4-special-registry/>>.

[Appendix A](#). Example BIND 9 Configuration

A BIND 9 recursive/caching DNS server can be configured to act as authoritative for the necessary DNS64 names as described below.

In `/etc/named.conf` the following line is added:

```
zone "ipv4only.arpa"          { type master; file "ipv4only"; };
```

The file `/var/named/ipv4only` is created with the following content:

```
$TTL 86400                ; Default TTL 24 hours
@ IN SOA nameserver.example. admin.nameserver.example. (
    2016052400            ; Serial
    7200                  ; Refresh ( 7200 = 2 hours)
    3600                  ; Retry   ( 3600 = 1 hour)
    15724800              ; Expire  (15724800 = 6 months)
    60                    ; Minimum
)
@ IN NS  nameserver.example.

@ IN A    192.0.0.170
@ IN A    192.0.0.171
@ IN AAAA 64:ff9b::192.0.0.170 ; If not using NAT64 Well-Known Prefix
@ IN AAAA 64:ff9b::192.0.0.171 ; Place actual prefix here
```

Authors' Addresses

Stuart Cheshire
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
USA

Phone: +1 408 974 3207
Email: cheshire@apple.com

David Schinazi
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
USA

Phone: +1 669 227 9921
Email: dschinazi@apple.com

