Stream: Internet Engineering Task Force (IETF)
RFC: 8880
Updates: 7050
Category: Standards Track
Published: August 2020
ISSN: 2070-1721

Authors: S. Cheshire *Apple Inc.* D. Schinazi *Google LLC*

RFC 8880
Special Use Domain Name 'ipv4only.arpa'

Abstract

NAT64 (Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers) allows client devices using IPv6 to communicate with servers that have only IPv4 connectivity.

The specification for how a client discovers its local network's NAT64 prefix (RFC 7050) defines the special name 'ipv4only.arpa' for this purpose. However, in its Domain Name Reservation Considerations section (Section 8.1), that specification (RFC 7050) indicates that the name actually has no particularly special properties that would require special handling.

Consequently, despite the well-articulated special purpose of the name, 'ipv4only.arpa' was not recorded in the Special-Use Domain Names registry as a name with special properties.

This document updates RFC 7050. It describes the special treatment required and formally declares the special properties of the name. It also adds similar declarations for the corresponding reverse mapping names.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

NAT64 (Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers) [RFC6146] allows client devices using IPv6 to communicate with servers that have only IPv4 connectivity.

DNS64 (DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers) [RFC6147] facilitates use of NAT64 by clients by generating synthesized IPv6 addresses for IPv4 servers that have no IPv6 address of their own, or by communicating the local network's NAT64 prefix to clients so that they can perform the IPv6 address synthesis themselves.

The specification for how a client discovers its local network's NAT64 prefix [RFC7050] defines the special name 'ipv4only.arpa' for this purpose, but in its Domain Name Reservation Considerations section (Section 8.1), that specification [RFC7050] indicates that the name actually has no particularly special properties that would require special handling and does not request IANA to record the name in the Special-Use Domain Names registry [SUDN].

Consequently, despite the well-articulated special purpose of the name, 'ipv4only.arpa' was not recorded in the Special-Use Domain Names registry [SUDN] as a name with special properties.

This omission was discussed in the document "Special-Use Domain Names Problem Statement" [RFC8244].

As a result of this omission, in cases where software needs to give this name special treatment in order for it to work correctly, there was no clear mandate authorizing software authors to implement that special treatment. Software implementers were left with the choice between not implementing the special behavior necessary for the name queries to work correctly or implementing the special behavior and being accused of being noncompliant with IETF DNS specifications.

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

1.1. Conventions and Terminology

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Reasons to Declare 'ipv4only.arpa' as Special

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

A typical client never has any reason to look up the IPv4 address records for 'ipv4only.arpa': no normal user is ever trying to view a website hosted at that domain name or trying to send email to an email address at that domain name. The name 'ipv4only.arpa' is already known, by IETF specification [RFC7050], to have exactly two IPv4 address records: 192.0.0.170 and 192.0.0.171. No client ever has to look up 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 IETF specification [RFC7050], that 'ipv4only.arpa' is an IPv4-only name, and it has no IPv6 AAAA address records. And yet, clients expect to receive, and do in fact receive, positive answers for these IPv6 AAAA address records that apparently should not exist.

This odd query behavior comes not because clients are using DNS to learn legitimate answers from the name's legitimate authoritative server, but because the DNS protocol has, in effect, been co-opted as an improvised client-to-middlebox communication protocol to look for a DNS64/NAT64 [RFC6147] [RFC6146] gateway and, if one is present, to request that it disclose the prefix it is using for IPv6 address synthesis.

This use of specially crafted DNS queries as an improvised client-to-middlebox communication protocol has a number of specific consequences, outlined below, which client software needs to take into account if the queries are to produce the desired results. This is particularly true when used on a multihomed host or when a VPN tunnel is in use. The name 'ipv4only.arpa' is most definitely a special name and needs to be listed in IANA's registry along with other DNS names that have special uses [SUDN].

3. Consequences of 'ipv4only.arpa' Not Being Declared Special

As a result of the original specification [RFC7050] not formally declaring 'ipv4only.arpa' to have special properties, there was no clear mandate for DNS software to treat this name specially. In particular, this lack of mandate for special treatment is relevant (a) to the name resolution APIs and libraries on client devices and (b) to DNS64 [RFC6147] implementations. These two aspects are discussed in more detail below.

3.1. Consequences for Name Resolution APIs and Libraries

A serious problem can occur with DNS64/NAT64 when a device is configured to use a recursive resolver other than the one it learned from the network.

A device joining a NAT64 network will learn the recursive resolver recommended for that network, typically via IPv6 Router Advertisement Options [RFC8106] or via DHCPv6 [RFC3646]. On a NAT64 network, it is essential that the client use the DNS64 recursive resolver recommended for that network, since only that recursive resolver can be relied upon to know the appropriate prefix(es) to use for synthesizing IPv6 addresses that will be acceptable to that NAT64 gateway.

However, it is becoming increasingly common for users to manually override their default DNS configuration because they wish to use some other public recursive resolver on the Internet, which may offer better speed, reliability, or privacy than the local network's default recursive resolver. At the time of writing, examples of widely known public recursive resolver services include Cloudflare Public DNS [DNS1], Google Public DNS [DNS8], and Quad9 [DNS9].

Another common scenario is the use of corporate or personal VPN client software. Both for privacy reasons and because the local network's recursive resolver will typically be unable to provide answers for the company's private internal host names, VPN client software usually overrides the local network's default configuration to divert some or all DNS requests so that they go to the company's own private internal recursive resolver instead of to the default recursive resolver the client learned from its local network. (The company's own private internal recursive resolvers typically have addresses that are themselves reached through the VPN tunnel, not via the public Internet.) As with the case described above of public recursive resolver services, the company's private internal recursive resolver cannot be expected to be able to synthesize IPv6 addresses correctly for use with the local network's NAT64 gateway, because the company's private internal recursive resolver is unlikely to be aware of the NAT64 prefix in use on the NAT64 network to which the client device is currently attached. It is clear that a single recursive resolver cannot meet both needs. The local network's recursive resolver cannot be expected to give answers for some unknown company's private internal host names, and a company's private internal recursive resolver cannot be expected to give correctly synthesized IPv6 addresses suitable for some unknown local network's NAT64 gateway.

Note that multiple NAT64 services may be simultaneously available to a client. For example, the local network may provide NAT64 service (to allow an IPv6-only client device to access IPv4-only Internet services), while at the same time, a corporate VPN may also provide NAT64 service (to allow a client connecting via an IPv6-only VPN tunnel to access IPv4-only corporate services). The NAT64 address synthesis prefixes for the two NAT64 services may be different. In this case, it is essential that the NAT64 address synthesis prefix used on the local network be the prefix learned from the local network's recursive resolver, and the NAT64 address synthesis prefix used on the VPN tunnel be the prefix learned from the VPN tunnel's recursive resolver.

The difficulty 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.

3.2. Consequences for DNS64 Implementations

As a result of there being no mandate for special treatment, queries for 'ipv4only.arpa' had to be handled normally, resulting in DNS64 gateways performing unnecessary queries to the authoritative 'arpa' name servers, both unnecessary IPv6 address record queries (DNS qtype "AAAA", always returning negative responses) and unnecessary IPv4 address record queries (DNS qtype "A", always returning the same positive responses).

Having DNS64 gateways around the world issue these queries generated additional load on the authoritative 'arpa' name servers, which was redundant when the name 'ipv4only.arpa' is defined, by IETF specification [RFC7050], to have exactly two IPv4 address records, 192.0.0.170 and 192.0.0.171, and no other IPv4 or IPv6 address records.

Also, at times, for reasons that remain 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 the DNS64 gateways and of the client devices that depend 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, affecting user experience on the client devices waiting for those synthesized replies.

4. Remedies

This document leverages operational experience to update the Domain Name Reservation Considerations section [RFC6761] of the earlier prefix discovery specification [RFC7050] with one that more accurately lists the actual special properties of the name 'ipv4only.arpa', so that software can legitimately implement the correct behavior necessary for better performance, better reliability, and correct operation.

These changes affect two bodies of software: (a) the name resolution APIs and libraries on client devices, and (b) DNS64 implementations.

The new special rules specified in this document for name resolution APIs and libraries state how they should select which recursive resolver to query to learn the IPv6 address synthesis prefix in use on a particular physical or virtual interface. Specifically, when querying for the name 'ipv4only.arpa', name resolution APIs and libraries should use the recursive resolver recommended by the network for the interface in question rather than a recursive resolver configured manually, a recursive resolver configured by VPN software, or a full-service recursive resolver running on the local host. Superficially, this might seem like a security issue, since the user might have explicitly configured the particular DNS resolver they wish to use, and rather than using that, the name resolution code ignores the user's stated preference and uses untrusted input received from the network instead. However, the 'ipv4only.arpa' query is not really a DNS query in the usual sense; even though it may look like a DNS query, it is actually an improvised client-to-middlebox communication protocol in disguise. For NAT64 to work at all, it is necessary for the interface on which NAT64 translation is being performed to tell the host the address of the DNS64 recursive resolver the host must use to learn the NAT64 prefix being used by that NAT64. This is typically done via IPv6 Router Advertisement Options for DNS Configuration [RFC8106] or via DNS Configuration options for DHCPv6 [RFC3646].

The new special rules specified in this document for DNS64 implementations recommend that they avoid performing run-time network queries for values that are known to be fixed by specification.

A useful property of the way NAT64 Prefix Discovery [RFC7050] was originally specified was that it allowed for incremental deployment. Even if existing DNS64 gateways, that were unaware of the special 'ipv4only.arpa' name, were already deployed, once IANA created the appropriate 'ipv4only.arpa' records, clients could begin to use the new facility immediately. Clients could send their special queries for 'ipv4only.arpa' to an ipv4only-unaware DNS64 gateway, and, as a side effect of its usual query processing (after a query to IANA's servers), the DNS64 gateway would then generate the correct synthesized response.

While this was a useful transition strategy to enable rapid adoption, it is not the ideal end situation. For better performance, better reliability, and lower load in IANA's servers, it is preferable for DNS64 gateways to be aware of the special 'ipv4only.arpa' name so that they can avoid issuing unnecessary queries. Network operators who wish to provide reliable, high-performance service to their customers are motivated to prefer DNS64 gateways that recognize the special 'ipv4only.arpa' name and apply the appropriate optimizations.

5. Security Considerations

One of the known concerns with DNS64 is that it conflicts with DNSSEC. If DNSSEC is used to assert cryptographically that a name has no IPv6 AAAA records, then this interferes with using DNS64 address synthesis to tell a client that those nonexistent 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 DNS64/NAT64 gateway (with an 'ipv4only.arpa' query) to learn the IPv6 address synthesis prefix, (ii) query for the (signed) IPv4 address records for the desired hostname and validate the response, and then (iii) perform its own IPv6 address synthesis locally, combining the IPv6 address synthesis prefix learned from the local DNS64/NAT64 gateway with the validated DNSSEC-signed data learned from the global Domain Name System.

It is conceivable that, over time, if DNSSEC adoption continues to grow, 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. At the time of publication, network operators have been observed "in the wild" deploying NAT64 service with DNS recursive resolvers that reply to 'ipv4only.arpa' queries but otherwise perform no other NAT64 address synthesis. 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 validate-and-synthesize-locally 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 DNS64/NAT64 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.

Hardcoding the known answers for 'ipv4only.arpa' IPv4 address record queries (DNS qtype "A") in recursive resolvers also reduces the risk of malicious devices intercepting those queries and returning incorrect answers. Because the 'ipv4only.arpa' zone has to be an insecure delegation (see below), DNSSEC cannot be used to protect these answers from tampering by malicious devices on the path.

With respect to the question of whether 'ipv4only.arpa' should be a secure or insecure delegation, we need to consider two paths of information flow through the network:

1. The path from the server authoritative for 'ipv4only.arpa' to the DNS64 recursive resolver
2. The path from the DNS64 recursive resolver to the ultimate client

The path from the authoritative server to the DNS64 recursive resolver (queries for IPv4 address records) need not be protected by DNSSEC, because the DNS64 recursive resolver already knows, by specification, what the answers are. In principle, if this were a secure delegation, and 'ipv4only.arpa' were a signed zone, then the path from the authoritative server to the DNS64 recursive resolver would still work, but DNSSEC is not necessary here. Run-time cryptographic signatures are not needed to verify compile-time constants. Validating the signatures could only serve to introduce potential failures into the system for minimal benefit.

The path from the DNS64 recursive resolver to the ultimate client (queries for IPv6 address records) *cannot* be protected by DNSSEC because the DNS64 recursive resolver is synthesizing IPv6 address answers and does not possess the DNSSEC secret key required to sign those answers.

Consequently, the 'ipv4only.arpa' zone **MUST** be an insecure delegation to give DNS64/NAT64 gateways the freedom to synthesize answers to those queries at will, without the answers being rejected by DNSSEC-capable resolvers. DNSSEC-capable resolvers that follow this specification **MUST NOT** attempt to validate answers received in response to queries for the IPv6 AAAA address records for 'ipv4only.arpa'. Note that the name 'ipv4only.arpa' has no use outside of being used for this special DNS pseudo-query used to learn the DNS64/NAT64 address synthesis prefix, so the lack of DNSSEC security for that name is not a problem.

The original NAT64 Prefix Discovery specification [RFC7050] stated, incorrectly:

> A signed "ipv4only.arpa." allows validating DNS64 servers (see [RFC6147] Section 3, Case 5, for example) to detect malicious AAAA resource records. Therefore, the zone serving the well-known name has to be protected with DNSSEC.

This document updates the previous specification [RFC7050] to correct that error. The 'ipv4only.arpa' zone **MUST** be an insecure delegation.

6. IANA Considerations

IANA has created an insecure delegation for 'ipv4only.arpa' to allow DNS64 recursive resolvers to create synthesized AAAA answers within that zone.

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 IANA IPv4 Special-Purpose Address Registry [SUv4]:

192.0.0.170

192.0.0.171

7. Domain Name Reservation Considerations

7.1. Special Use Domain Name 'ipv4only.arpa'

The name 'ipv4only.arpa' is defined, by IETF specification [RFC7050], to have two IPv4 address records with rdata 192.0.0.170 and 192.0.0.171.

When queried via a DNS64 recursive resolver [RFC6147], 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 IPv4 or IPv6 address records. 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 resolver 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 DNS64/NAT64 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.

Learning a network's NAT64 prefix is, by its nature, an interface-specific operation, and the special DNS query used to learn this interface-specific NAT64 prefix **MUST** be sent to the DNS recursive resolver address(es) the client learned via the configuration machinery for that specific client interface. The NAT64 prefix is a per-interface property, not a per-device property.

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 resolver address(es) (including client devices that run their own local recursive resolver and use the loopback address as their configured recursive resolver address), all queries for 'ipv4only.arpa' and any subdomains of that name **MUST** be sent to the recursive resolver learned from the network interface in question via IPv6 Router Advertisement Options for DNS Configuration [RFC8106], DNS Configuration options for DHCPv6 [RFC3646], or other configuration mechanisms. Because DNS queries for 'ipv4only.arpa' are actually a special middlebox communication protocol, it is essential that they go to the correct middlebox for the interface in question, and failure to honor this requirement would cause failure of the NAT64 Prefix Discovery mechanism [RFC7050].

One implication of this is that, on multihomed devices (devices that allow more than one logical or physical IP interface to be active at the same time, e.g., cellular data and Wi-Fi, or one physical interface plus a VPN connection), clients **MUST** use interface-aware name resolution APIs. On different (logical or physical) interfaces, different DNS64 answers may be received, and DNS64 answers are only valid for the interface on which they were received. On multihomed devices (including devices that support VPN), name resolution APIs that do not include interface parameters will not work reliably with NAT64. On single-homed devices, interface-unaware name resolution APIs are acceptable since when only one interface can be active at a time, there is no need to specify an interface.

DNSSEC-capable resolvers **MUST NOT** attempt to validate answers received in response to queries for the IPv6 AAAA address records for 'ipv4only.arpa' since, by definition, any such answers are generated by the local network's DNS64/NAT64 gateway, not the authoritative server responsible for that name.

4. For the purposes of this section, recursive resolvers fall into two categories. The first category is traditional recursive resolvers, which includes *forwarding* recursive resolvers, as commonly implemented in residential home gateways, and *iterative* recursive resolvers, as commonly deployed by ISPs. The second category is DNS64 recursive resolvers, whose purpose is to synthesize IPv6 address records. These may be *forwarding* DNS64 recursive resolvers or *iterative* DNS64 recursive resolvers, and they work in partnership with a companion NAT64 gateway to communicate the appropriate NAT64 address synthesis prefix to clients. More information on these terms can be found in the DNS Terminology document [RFC8499].

   Traditional forwarding recursive resolvers **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 forwarding recursive resolver, such as built in to a residential home gateway, may itself be downstream of a DNS64 recursive resolver. Passing through the 'ipv4only.arpa' queries to the upstream DNS64 recursive resolver will allow the correct NAT64 prefix to be discovered.

Traditional iterative recursive resolvers that are not explicitly configured to synthesize IPv6 prefixes on behalf of a companion NAT64 gateway need not recognize 'ipv4only.arpa' as special or take any special action.

Forwarding or iterative recursive resolvers that have been explicitly configured to perform DNS64 address synthesis in support of a companion NAT64 gateway (i.e., "DNS64 recursive resolvers") **MUST** recognize 'ipv4only.arpa' as special. The authoritative name servers for 'ipv4only.arpa' cannot be expected to know the local network's NAT64 address synthesis prefix, so consulting the authoritative name servers for IPv6 address records for 'ipv4only.arpa' is futile. All DNS64 recursive resolvers **MUST** recognize 'ipv4only.arpa' (and all of its subdomains) as special, and they **MUST NOT** attempt to look up NS records for 'ipv4only.arpa' or otherwise query authoritative name servers in an attempt to resolve this name. Instead, DNS64 recursive resolvers **MUST** act as authoritative for this zone, by generating immediate responses for all queries for 'ipv4only.arpa' (and any subdomain of 'ipv4only.arpa'), with the one exception of queries for the DS record. Queries for the DS record are resolved the usual way to allow a client to securely verify that the 'ipv4only.arpa' zone has an insecure delegation. Note that this exception is not expected to receive widespread usage, since any client compliant with this specification already knows that 'ipv4only.arpa' is an insecure delegation and will not attempt DNSSEC validation for this name.

DNS64 recursive resolvers **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 except DS.

For all subdomains of 'ipv4only.arpa', DNS64 recursive resolvers **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.

Note that this is *not* a locally served zone in the usual sense of that term [RFC6303] because this rule applies *only* to DNS64 recursive resolvers, not to traditional forwarding or iterative recursive resolvers.

5. Authoritative name server software need not recognize 'ipv4only.arpa' as special or handle it in any special way.
6. Generally speaking, operators of authoritative name 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. Only the administrators of the 'arpa' namespace need to be aware of this name's purpose and how it should be configured. In particular, 'ipv4only.arpa' **MUST** have the required records, and **MUST** be an insecure delegation, to allow DNS64

recursive resolvers to create synthesized AAAA answers within that zone. Making the 'ipv4only.arpa' zone a secure delegation would make it impossible for DNS64 recursive resolvers to create synthesized AAAA answers that will be accepted by DNSSEC validating clients, thereby defeating the entire purpose of the 'ipv4only.arpa' name.

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.

7.2. Names '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 IANA IPv4 Special-Purpose Address Registry [SUv4], 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 IETF specification [RFC7050], to have only one DNS record, type PTR, with rdata 'ipv4only.arpa'.

The name '171.0.0.192.in-addr.arpa' is defined, by IETF specification [RFC7050], to have only one 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 resolver APIs and libraries, as described in item 3 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 call the name resolution API or library 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 resolvers **SHOULD NOT** recognize these two reverse mapping names as special and **SHOULD NOT**, by default, give them any special treatment.

5. Authoritative name server software need not recognize these two reverse mapping names as special or handle them in any special way.
6. Generally speaking, most operators of authoritative name 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. Only the operators of the authoritative name servers for these two reverse mapping names need to be aware that these names are special, and require fixed answers specified by IETF specification.
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.

7.2.1. ip6.arpa Reverse Mapping PTR Records

For all IPv6 addresses synthesized by a DNS64 recursive resolver, the DNS64 recursive resolver 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 resolver 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 resolver 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.

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, <https://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, <https://www.rfc-editor.org/info/rfc3646>.

[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, <https://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, <https://www.rfc-editor.org/info/rfc6147>.

[RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, DOI 10.17487/RFC6761, February 2013, <https://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, <https://www.rfc-editor.org/info/rfc7050>.

[RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 8106, DOI 10.17487/RFC8106, March 2017, <https://www.rfc-editor.org/info/rfc8106>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[RFC6303] Andrews, M., "Locally Served DNS Zones", BCP 163, RFC 6303, DOI 10.17487/RFC6303, July 2011, <https://www.rfc-editor.org/info/rfc6303>.

[RFC8244] Lemon, T., Droms, R., and W. Kumari, "Special-Use Domain Names Problem Statement", RFC 8244, DOI 10.17487/RFC8244, October 2017, <https://www.rfc-editor.org/info/rfc8244>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

[SUDN] IANA, "Special-Use Domain Names", <https://www.iana.org/assignments/special-use-domain-names/>.

[SUv4] IANA, "IANA IPv4 Special-Purpose Address Registry", <https://www.iana.org/assignments/iana-ipv4-special-registry/>.

[DNS1] Cloudflare, "1.1.1.1 - The free app that makes your Internet safer.", <https://1.1.1.1/>.

[DNS8] Google, "Google Public DNS", <https://developers.google.com/speed/public-dns/>.

[DNS9] Quad9, "Internet Security and Privacy In a Few Easy Steps", <https://quad9.net/>.

Appendix A. Example BIND 9 Configuration

A BIND 9 recursive resolver 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 Well-Known Prefix
@ IN AAAA 64:ff9b::192.0.0.171 ; place chosen NAT64 prefix here
```

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 this document.

Thanks to Erik Kline for pointing out that the in-addr.arpa names are special, too.

Thanks to Mark Andrews for conclusively pointing out the reasons why the 'ipv4only.arpa' zone must be an insecure delegation in order for the NAT64 Prefix Discovery mechanism [RFC7050] to work and for many other very helpful comments.

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.

Thanks to Scott Bradner, Bernie Volz, Barry Leiba, Mirja Kuehlewind, Suresh Krishnan, Benjamin Kaduk, Roman Danyliw, Eric Vyncke, and the other IESG reviewers for their thoughtful feedback.

Thanks to Dave Thaler and Warren Kumari for generously helping shepherd this document through the publication process.

Authors' Addresses

Stuart Cheshire
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America
Phone: +1 (408) 996-1010
Email: cheshire@apple.com

David Schinazi
Google LLC
1600 Amphitheatre Parkway
Mountain View, California 94043
United States of America
Email: dschinazi.ietf@gmail.com