v6ops 山本 桃歌 (M. Yamamoto)
Internet-Draft The University of Tokyo/WIDE Project
Intended status: Informational 豊田 安信 (Y. Toyota)
Expires: 10 September 2023 Keio University/WIDE Project
9 March 2023

~~IPv6~~ IPv6-only ~~capable~~ Capable ~~resolver~~ Resolvers ~~utilising~~ Utilising NAT64

draft-momoka-v6ops-ipv6-only-resolver-01

Abstract

By performing ~~IPv4~~ IPv4-~~to~~ to-IPv6 translation, IPv6-only iterative resolvers can operate in an IPv6-only environment. When a specific DNS zone is only served by an IPv4-only authoritative server, the iterative resolver will translate ~~the~~ that IPv4 address to an IPv6 to ~~access~~ reach that~~the~~ authoritative server~~'s IPv4 address~~ via ~~stateful~~ a NAT64 function. This mechanism allows IPv6-only iterative resolvers to initiate communications to IPv4-only authoritative servers.

This document does not specify any new protocol extension but leverages existing tools.

Commenté [BMI1]: The proposed approach would apply for recursive resolvers as well.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the IPv6 Operations Working Group mailing list (v6ops@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/v6ops/.

Source for this draft and an issue tracker can be found at https://github.com/momoka0122y/draft-momoka-ipv6-only-resolver.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes how an IPv6-only iterative resolver can use stateful NAT64 [NAT64] to connect to an IPv4-only authoritative server by performing ~~IPv4~~ IPv4-~~to~~ to-IPv6 address translation [RFC6052] when ~~sending~~generating a query. When a specific DNS zone is only served by an IPv4-only authoritative server (which has only an A record), an IPv6-only iterative resolver cannot resolve that zone due to having no access to an IPv4 network. However, by performing ~~IPv4~~ IPv4-~~to~~ to-IPv6 address translation and utilizing the stateful NAT64, accessing an IPv4-only authoritative server will be possible.

This document does not require any protocol extensions.

This document is meant to exemplify how existing tools can be used to allow IPv6-only resolvers to reach IPv4-only upstream resolvers. DNS is thus seen as an application that uses NAT64.

The document focuses on the exchanges between iterative resolvers and authoritative resolvers but can be generalized to cover communications between IPv6-only recursive resolvers and upstream IPv4-only resolvers.

2. Terminology

* Iterative resolver: A DNS server that repeatedly makes non-recursive queries and follows referrals and/or aliases. The iterative resolution algorithm is described in Section 5.3.3 of [RFC1034].

Commenté [BMI2]: You may simply refer to RFC8499

* IPv6-only iterative resolvers: Iterative resolvers that only have IPv6 connectivity.

* IPv6/IPv4 translator: A ~~device~~ function that translates IPv6 packets to IPv4 packets and vice versa. It is only required that the communication initiated from the IPv6 side be supported.

* IPv4-only authoritative server: An authoritative server with only IPv4 connectivity, or an authoritative server with only an A record registered so it can only be accessed by IPv4.

3. Motivation and Problem Solved

An iterative resolver is one of the applications that require IPv4 connectivity. As stated in BCP91 [RFC3901], “every recursive name server SHOULD be either IPv4-only or dual stack.” This is because some authoritative servers do not support IPv6. As of 2023, even some of the most frequently queried authoritative servers cannot be accessed via IPv6. Without the utilization of an IPv6/IPv4 translation mechanism~~NAT64~~, IPv6-only resolvers need to forward queries to a dual-stack recursive name server performing the iterative queries.

Commenté [BMI3]: This text is about recursive servers.

The current situation where an iterative resolver cannot operate without IPv4 reachability may hinder the operation of a network's own iterative resolver in an IPv6-only network. Therefore, this document describes how iterative resolvers can be used without issues in IPv6-only networks by utilizing NAT64 as an IPv6/IPv4 translation mechanism.

The NAT64/DNS64 mechanisms enable~~s~~ IPv6-only clients in an IPv6-only network to communicate with remote IPv4-only nodes. However, applications that rely upon ~~using~~ IPv4 address literals ~~IPv4 addresses~~ instead of DNS names will fail (unless 464XLAT [RFC6877] is used). An iterative resolver cannot use the DNS64 because it is a service that uses literal IP addresses. This problem can be solved by the iterative resolver converting IPv4 addresses to IPv6 addresses ~~by adding~~using the Pref64::/n ~~prefix~~NAT64 prefix and following the address translation algorithm in [RFC6052]. In doing so, ~~and thus the~~ an IPv6 packet conveying the query is directed to a stateful NAT64 function~~gateway~~ that converts the IPv6 packet

Commenté [BMI4]: Just point to RFC6052 for the algo to build the address

Commenté [BMI5]: No need to respecify how addresses are built.

to an IPv4 packet. With this implementation, an iterative resolver can be operated even inside an IPv6-only network.

3.1. Deployment Scenarios and Examples

The deployment of IPv6-only networks is in progress, as demonstrated by [draft-xie-v6ops-framework-md-ipv6only-underlay]. By operating an IPv6-only network and limiting IPv4 reachability to NAT64 ~~devices~~functions, operators can ~~reduce~~ optimize IPv4 address usage and concentrate on IPv6 operations, which is generally believed to lower operational costs and optimize operations compared to a dual-stack environment.

In examples of past RFCs, name resolvers have always had an IPv4 address. For example, all three use cases for DNS64 in [RFC~~-~~6147] are dual-stack name servers.

```
+----------------------+         +----------------+
|IPv6 network          |         |     IPv4       |
|              +--------------+  |   Internet     |
|            |--| Name server |--|                |
|            |  | with DNS64  |  |   +-----+      |
|  +-----+   |  +--------------+  |   | H2  |      |
|  | H1  |---|          |         |   +-----+      |
|  +-----+   |   +-------------+  |   192.0.2.1    |
|            |---| IPv6/IPv4   |--|                |
|            |   | Translator  |  |                |
|            |   +-------------+  |                |
|            |          |         |                |
+----------------------+         +----------------+
```

Figure 1: Example network setup of the use of DNS64 described in ~~RFC6147~~ Section7.1 of RFC6147

```
+----------------------+         +----------------+
|IPv6 network          |         |     IPv4       |
|                 +---------+    |   Internet     |
|           |-----| Name    |----|                |
| +------+  |     | server  |    |   +-----+      |
| | H1   |  |     +---------+    |   | H2  |      |
| |with  |---|          |         |   +-----+      |
| |DNS64|   |   +-------------+  |   192.0.2.1    |
| +-----+   |---| IPv6/IPv4   |--|                |
|            |   | Translator  |  |                |
|            |   +-------------+  |                |
|            |          |         |                |
+----------------------+         +----------------+
```

Figure 2: Example network setup of the use of DNS64 described in Section7.2 of RFC6147 ~~Section7.2~~

However, it is necessary to consider the existence of an IPv6 single-stack full-service resolver. In this document we consider an IPv6-only network where the iterative resolver is inside the IPv6-only network and does not have an IPv4 address. This is to restrict IPv4 management to the NAT64 ~~device~~ function.

```
+---------------------------+        +-----------------------+
```

```
| IPv6 network               |          |   IPv4                 |
|                            |          |  Internet              |
|                  |         |          |                        |
| +----------+     |         |          |  +--------------+      |
| |IPv6-only |     |         |          |  |Authoritative |      |
| |Iterative |     |         |          |  |server        |      |
| |resolver  |---|    +------------+    |  +--------------+      |
| +----------+     |---| IPv6/IPv4  |--|  192.0.2.1             |
|                  |    | Translator |  |                        |
|                  |    +------------+  |                        |
|                            |          |                        |
+---------------------------+          +-----------------------+
```

Figure 3: Network example referenced in this document with an IPv6-only iterative resolver

4. Solution with ~~existing~~ Existing ~~protocols~~Protocols

This section ~~provides~~ describes the mechanism of an IPv6-only capable resolver utilizing stateful NAT64. We ~~will~~ assume that ~~we have~~ one or more IPv6/IPv4 translators ~~boxes~~ [NAT64] are connecting an IPv6 network to an IPv4 network. The stateful NAT64 ~~device~~ provides translation service and bridges the two networks, allowing communication between IPv6-only hosts and IPv4-only hosts. The IPv6-only capable resolver proposed in this document performs the IPv4 to IPv6 synthesis for the resolver to communicate with IPv4-only servers via stateful NAT64. By using stateful NAT64, this IPv6-only iterative resolver can be considered dual stack in the sense of BCP91 [RFC3901].

4.1. Finding an Authoritative ~~server~~ Server with ~~only~~ IPv4-only ~~addresses~~Addresses

Before ~~the~~ an iterative resolver sends queries to start a resolution, it may sort the SLIST data structure described in [RFC1034] to use the authoritative servers with IPv6 addresses first, and use servers with only an IPv4 address later. If the resolver finds only an A record for an authoritative server, the resolver should perform address synthesis to the IPv4 address of the authoritative server, converting IPv4 addresses to IPv6 by following the algorithm in [RFC6052] ~~adding the prefix Pref64::/n~~, so that the IPv6 packet carrying the query is ~~routed~~ forwarded to a ~~stateful~~ NAT64 ~~gateway~~function, which will convert~~s~~ the IPv6 packet with a destination IPv4-converted IPv6 address that matches the NAT64 prefix to an IPv4 packet. It is not recommended to ~~synthesise~~synthesize ~~an~~ IPv4 addresses of an authoritative server if it also has an IPv6 address.

Commenté [BMI6]: Just point to the address selection RFC: RFC6724.

4.2. ~~Generation~~ Generating ~~of the IPv6~~ IPv4-converted IPv6 ~~Representations of IPv4~~ Addresses

4.2.1. Obtaining the Pref64::/n of ~~the~~ a ~~stateful~~ NAT64

The iterative resolver can obtain the Pref64::/n used by the network's stateful NAT64 either by static configuration or by using a discovery mechanism~~s~~. Static configuration ~~may be~~is the most likely scenario~~, as the iterative resolver server may also serve as a DNS64 server~~.

The Port Control Protocol [RFC7225] or Router Advertisements [RFC8781] are two options available to the resolver if it wishes to use a discovery mechanism to find the Pref64::/n. Using the mechanisms described in [RFC7050] or [draft-hunek-v6ops-nat64-srv] may not work because they require a resolver to work.

4.2.2. Performing ~~the~~ Address Synthesis

The address translation algorithm ~~can be~~ is performed by following Section 2.3 of [RFC6052]. After the synthesis is done, the IPv6-only iterative resolver can send a query to the IPv4-converted IPv6 address.

4.3. Use of the I~~i~~terative ~~resolver~~ Resolver as DNS64

As the iterative resolver is used within an IPv6-only network, the server ~~can~~ may also ~~perform~~ provide ~~as~~ the DNS64 function [DNS64] when an AAAA record is queried from a ~~STUB~~ stub resolver but the ~~domain~~ target resource only has ~~an~~ A records.

5. Deployment Notes

TODO

6. Security Considerations

This algorithm does not change any part of the DNS message, just the packet type from IPv4 to IPv6 and the destination IP address from an IPv4 address to the ~~synthesised~~synthesized IPv6 address, so there should be no problems with DNSSEC.

Commenté [BMI7]: Which one ?

No algo is defined in this document.

7. IANA Considerations

This document has no IANA actions.

8. Implementation Status

BIND has a WIP branch.

https://gitlab.isc.org/isc-projects/bind9/-/merge_requests/6334/commits

Unbound has a PR from a contributor.

https://github.com/NLnetLabs/unbound/issues/721

9. References

9.1. Normative References

[DNS64] 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/rfc/rfc6147>.

[NAT64] 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/rfc/rfc6146>.

[RFC3901] Durand, A. and J. Ihren, "DNS IPv6 Transport Operational Guidelines", BCP 91, RFC 3901, DOI 10.17487/RFC3901, September 2004, <https://www.rfc-editor.org/rfc/rfc3901>.

[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, <https://www.rfc-editor.org/rfc/rfc6052>.

Commenté [BMI8]: This is not normative

Commenté [BMI9]: This is not normative.

9.2. Informative References

[draft-hunek-v6ops-nat64-srv] Huněk, M., "NAT64/DNS64 detection via SRV Records", Work in Progress, Internet-Draft, draft-hunek-v6ops-nat64-srv-04, 11 December 2022, <https://datatracker.ietf.org/doc/html/draft-hunek-v6ops-nat64-srv-04>.

[draft-xie-v6ops-framework-md-ipv6only-underlay] Xie, C., Ma, C., Li, X., Mishra, G. S., Boucadair, M., and T. Graf, "Framework of Multi-domain IPv6-only Underlay Networks and IPv4 as a Service", Work in Progress, Internet-Draft, draft-xie-v6ops-framework-md-ipv6only-underlay-05, 21 October 2022, <https://datatracker.ietf.org/doc/html/draft-xie-v6ops-framework-md-ipv6only-underlay-05>.

[ietf-v6ops-ipv6-deployment] Fioccola, G., Volpato, P., Martinez, J. P., Mishra, G. S., and C. Xie, "IPv6 Deployment Status", Work in Progress, Internet-Draft, draft-ietf-v6ops-ipv6-deployment-10, 1 December 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-v6ops-ipv6-deployment-10>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/rfc/rfc1034>.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <https://www.rfc-editor.org/rfc/rfc6877>.

[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/rfc/rfc7050>.

[RFC7225] Boucadair, M., "Discovering NAT64 IPv6 Prefixes Using the Port Control Protocol (PCP)", RFC 7225, DOI 10.17487/RFC7225, May 2014, <https://www.rfc-editor.org/rfc/rfc7225>.

[RFC8781] Colitti, L. and J. Linkova, "Discovering PREF64 in Router Advertisements", RFC 8781, DOI 10.17487/RFC8781, April 2020, <https://www.rfc-editor.org/rfc/rfc8781>.

Acknowledgments

TODO: acknowledge people.

Thank you for reading this draft.

Authors' Addresses

Momoka Yamamoto
The University of Tokyo/WIDE Project
Email: momoka.my6@gmail.com

Additional contact information:

山本 桃歌
The University of Tokyo/WIDE Project

Yasunobu Toyota
Keio University/WIDE Project
Email: yas-nyan@sfc.wide.ad.jp

Additional contact information:

豊田 安信
Keio University/WIDE Project