

Workgroup: v6ops  
Internet-Draft:  
draft-ietf-v6ops-ipv6-only-resolver-00  
Published: 23 October 2023  
Intended Status: Informational  
Expires: 25 April 2024  
Authors: (M. Yamamoto)  
The University of Tokyo/WIDE Project  
(Y. Toyota)  
Keio University/WIDE Project  
**IPv6-only Capable Resolvers Utilising NAT64**

## Abstract

This document outlines how IPv6-only iterative resolvers can use stateful NAT64 to establish communications with IPv4-only authoritative servers. It outlines a mechanism for translating the IPv4 addresses of authoritative servers to IPv6 addresses to initiate communications. Through the mechanism of IPv4-to-IPv6 address translation, these resolvers can operate in an IPv6-only network environment.

## 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

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 <https://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 25 April 2024.

## Copyright Notice

Copyright (c) 2023 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

- [1. Introduction](#)
- [2. Terminology](#)
- [3. Motivation and Problem Solved](#)
  - [3.1. Deployment Scenarios and Examples](#)
- [4. Solution with Existing Protocols](#)
  - [4.1. Finding an Authoritative Server with only IPv4 addresses](#)
  - [4.2. Generating IPv4-converted IPv6 Addresses](#)
    - [4.2.1. Obtaining the Pref64::](#)
    - [4.2.2. Performing Address Synthesis](#)
- [5. Security Considerations](#)
- [6. IANA Considerations](#)
- [7. Implementation Status](#)
- [8. References](#)
  - [8.1. Normative References](#)
  - [8.2. Informative References](#)
- [Acknowledgments](#)
- [Authors' Addresses](#)

## 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-to-IPv6 address translation [[RFC6052](#)] when 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-to-IPv6 address translation and utilising the stateful NAT64, it will be possible to access an IPv4-only authoritative server.

This document is meant to exemplify how existing tools can be used to allow IPv6-only resolvers to reach IPv4-only 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 IPv4-only resolvers.

## 2. Terminology

\*Iterative resolver: A DNS server that repeatedly makes non-recursive queries and follows referrals and/or aliases, as defined in DNS Terminology [[RFC8499](#)]

\*IPv6-only iterative resolvers: Iterative resolvers with only IPv6 connectivity.

\*IPv6/IPv4 translator: A 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 that either has only IPv4 connectivity or is registered with only an A record, making it accessible solely via 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, IPv6-only resolvers need to forward queries to a dual-stack recursive name server performing the iterative queries.

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 utilising NAT64 as an IPv6/IPv4 translation mechanism.

The NAT64/DNS64 mechanisms enable IPv6-only clients in an IPv6-only network to communicate with remote IPv4-only nodes. However, applications that rely upon IPv4 address literals instead of DNS names will fail (unless 464XLAT [[RFC6877](#)] is used). An iterative resolver is a service that uses literal IP addresses, so it cannot

use the DNS64. This problem can be solved by the iterative resolver converting IPv4 addresses to IPv6 addresses using the Pref64::/n NAT64 prefix and following the address translation algorithm in [RFC6052]. In doing so, an IPv6 packet conveying the query is directed to a stateful NAT64 function that converts the IPv6 packet 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 [RFC9386]. By operating an IPv6-only network and limiting IPv4 reachability to NAT64 functions, operators can 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 [RFC6147] are dual-stack name servers.

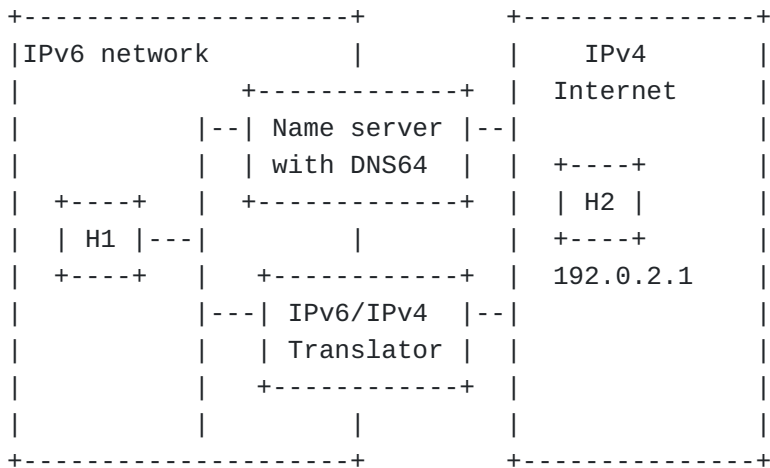


Figure 1: Example network setup of the use of DNS64 described in Section 7.1 of RFC6147

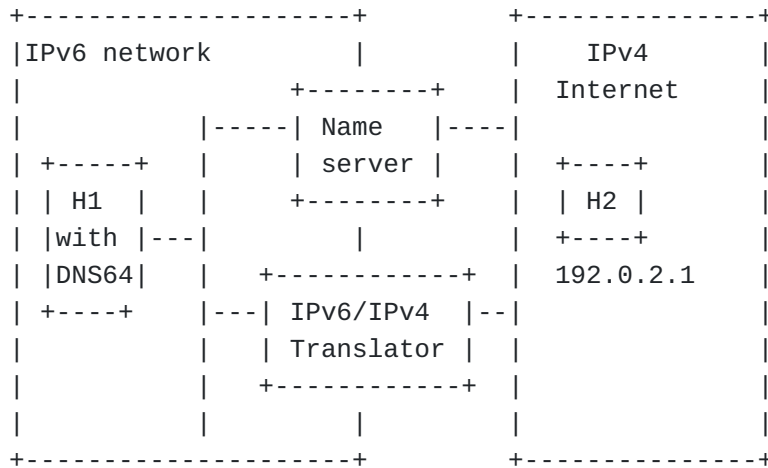


Figure 2: Example network setup of the use of DNS64 described in Section 7.2 of RFC6147

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 function.

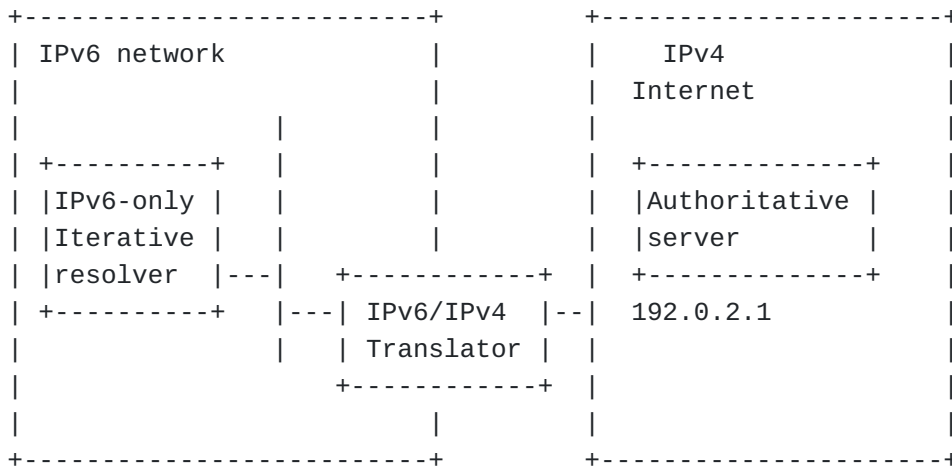


Figure 3: Network example referenced in this document with an IPv6-only iterative resolver

#### 4. Solution with Existing Protocols

This section describes the mechanism of an IPv6-only capable resolver utilising stateful NAT64. We assume that one or more IPv6/IPv4 translators [NAT64] are connecting an IPv6 network to an IPv4 network. The stateful NAT64 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 aligns with the dual-stack requirements of BCP91 [[RFC3901](#)].

#### **4.1. Finding an Authoritative Server with only IPv4 addresses**

Before initiating a query, an iterative resolver may prioritize authoritative servers with IPv6 addresses by sorting the SLIST data structure, as described in [[RFC1034](#)]. 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](#)]. With this the IPv6 packet carrying the query is routed to a stateful NAT64 function, which will convert the IPv6 packet with a destination IPv4-converted IPv6 address that matches the NAT64 prefix to an IPv4 packet. It is not recommended to synthesize IPv4 addresses of an authoritative server if it also has an IPv6 address.

#### **4.2. Generating IPv4-converted IPv6 Addresses**

##### **4.2.1. Obtaining the Pref64::**

The iterative resolver can obtain the Pref64::

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::RFC7050] or [[draft-hunek-v6ops-nat64-srv](#)] does not work because they require a resolver to work.

##### **4.2.2. Performing Address Synthesis**

The address translation algorithm 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.

#### **5. Security Considerations**

The use of NAT64 for address translation does not affect DNSSEC operations as no part of the DNS message is altered.

#### **6. IANA Considerations**

This document has no IANA actions.

## 7. Implementation Status

BIND has a work-in-progress branch available at:

[https://gitlab.isc.org/isc-projects/bind9/-/merge\\_requests/6334/commits](https://gitlab.isc.org/isc-projects/bind9/-/merge_requests/6334/commits)

Unbound has an implementation in source, available from below PR:

<https://github.com/NLnetLabs/unbound/pull/722>

Building from source or using a version after Version 1.19 will allow to use this function.

## 8. References

### 8.1. Normative References

- [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. Ihen, "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>>.

### 8.2. 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-05, 15 June 2023, <<https://datatracker.ietf.org/doc/html/draft-hunek-v6ops-nat64-srv-05>>.
- [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>>.
- [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/rfc/rfc6147>>.

**[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>>.

**[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/rfc/rfc8499>>.

**[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>>.

**[RFC9386]**

Fioccola, G., Volpato, P., Palet Martinez, J., Mishra, G., and C. Xie, "IPv6 Deployment Status", RFC 9386, DOI 10.17487/RFC9386, April 2023, <<https://www.rfc-editor.org/rfc/rfc9386>>.

**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](mailto:momoka.my6@gmail.com)  
Additional contact information:

The University of Tokyo/WIDE Project

Yasunobu Toyota  
Keio University/WIDE Project

Email: [yas-nyan@sfc.wide.ad.jp](mailto:yas-nyan@sfc.wide.ad.jp)  
Additional contact information:



Keio University/WIDE Project