Workgroup: Network Working Group
Internet-Draft:
draft-boucadair-dnsop-prefix64-02
Updates: 6147 (if approved)
Published: 5 January 2022
Intended Status: Standards Track
Expires: 9 July 2022
Authors: M. Boucadair
Orange

An EDNS0 Option for Sharing Pref64::/n

Abstract

This document specifies an Extension Mechanisms for DNS (EDNS0) option to convey the IPv6 prefix used to build IPv4-converted IPv6 addresses. When conveyed in a DNS query, the option communicates the IPv6 prefix used in the network from which the query was originated. Such a network is assumed to enable a Network Address and Protocol Translation from IPv6 clients to IPv4 servers (NAT64) function. DNS64-capable servers will use that prefix to build synthesized AAAA records, rather than relying on a preconfigured prefix. When conveyed in a DNS reply, the option conveys the IPv6 prefix that is used by a DNS64-capable server to synthesized AAAA records. Such information helps to automatically detect mismatches between the local NAT64 configuration and the one enforced at the DNS64 server. Also, security-aware and validating hosts may use the new EDNS0 option to signal the presence of a NAT64 function. That signal is used by the DNS server to fill the additional section of the AAAA reply in order to supply A RRs of the target. Dual queries and delays are thus avoided.

This document updates RFC 6147.

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 9 July 2022.

Copyright Notice

Copyright (c) 2022 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

Network Address and Protocol Translation from IPv6 clients to IPv4 servers (NAT64) function [RFC6146] is widely deployed, especially in cellular networks. Such a function is solicited when an IPv6-only host communicates with an IPv4-only server. For that communication to take place, IPv4-only servers are represented in the IPv6 domain by synthesizing IPv6 addresses based on IPv4 addresses (called, IPv4-converted IPv6 addresses). The address translation algorithm is specified in [RFC6052]. In addition to an IPv4 address, this algorithm uses a dedicated IPv6 prefix as input. Such a prefix can be the Well-Known Prefix (i.e., 64:ff9b::/96) or a Network-Specific Prefix (NSP).

DNS64 [RFC6147] specifies a companion mechanism to represent IPv4-only servers in the IPv6 domains. Such a mechanism relies upon the same address translation algorithm as the one used by the NAT64 function. When both DNS64 and NAT64 are deployed in the same network, the same IPv6 prefix must be used to feed the address translation algorithm (Section 2 of [RFC6147]). A sample deployment

scenario is depicted in Figure 1. Note that no mechanism is supported to synchronize the prefix configured in both functions. In particular, there is no communication between the DNS64 and NAT64 functions.

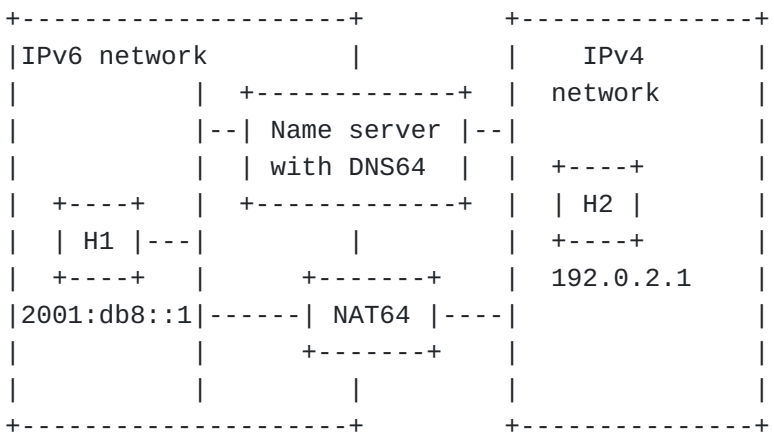

Figure 1: Sample Deployment (RFC6146)

In networks where DNS64 is enabled, some deployments use distinct IP addresses to reach the "normal" DNS server and the DNS64 server. This is used to demux queries issues by IPv6-only hosts from those from dual-stack hosts. The mechanism defined in this document allows to use the same DNS configuration for both IPv6-only and dual-stack hosts.

NAT64 does not require a DNS64 server to be enabled and, even if it is used, it does not mandate that it is enabled in the same network. As such, several public DNS64 servers are currently available for use over the Internet. However, these servers are restricted to the Well-Known Prefix. Users who decides to bypass their network-provisioned DNS64 server (e.g., including both trusted (access network, typically) and untrusted networks such as Airports) may experience connectivity issues if an NSP is used in their local networks (Section 4.4 of [RFC8683]). This document solves that issues by specifying a mechanism that allows to use any DNS64 server, not only the one hosted in the network that enables the NAT64.

If the IPv4 address of a remote IPv4-only server is known to an IPv6-only host (e.g., IPv4 literals, legacy DNS), the IPv6-only host can proceed with local address synthesis. For example, the stub resolver on the IPv6-only host tries to obtain (native) AAAA records, and if they are not found, the DNS64 function on the host will send a query for A records and then synthesize AAAA records. This behavior requires the host's stub-resolver to learn the prefix used for IPv6/IPv4 translation and synthesize AAAA records

accordingly. Many mechanisms were specified to discover such prefix, e.g.:

* [RFC7225] defines a new Port Control Protocol (PCP) option [RFC6887] to inform hosts about the Pref64::/n and suffix used by a NAT64 function.

* [RFC8781] specifies a Neighbor Discovery option used in Router Advertisements (RAs) to communicate NAT64 prefixes to hosts.

The reader may refer to [RFC7050][RFC7051] for an analysis on the issues related to the discovery of the Pref64::/n.

In some environments two DNS queries are issued even if the host is serviced using an IPv6-only connectivity (typically, AAAA followed by A). These two queries are sent sequentially, which introduces an extra delay when the target resource is IPv4-only. Such delay can be prevented owing to the mechanism specified in this document. As a side effect, the mechanism optimizes the load on DNS64 servers as only one query will be used instead of two.

This document updates [RFC6146] as it extends the DNS64 processing to also consider the supplied Pref64::/n in an EDNS0 option to synthesize AAA records. In particular statements such as "locally configured Pref64::/n" are updated to "locally configured Pref64::/n or Pref64::/n supplied in an EDNS0 PREFIX64 option". To that aim, this document leverages the aforementioned discovery mechanism to detect the presence of a NAT64 function.

In summary, the mechanism defined in this document is meant to:

* Provide a signal to indicate the support of NAT64 in a network.

* Allow a DNS64 server to service clients with distinct NAT64 prefixes.

* Avoid delays when both A and AAA queries are required.

* Optimize load on DNS server as only one query is generated rather that duplicating load when both AAA and A queries are required.

2. 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.

The reader should be familiar with terms and concepts defined in [RFC6052], [RFC6146], and [RFC6147]. Also, the document makes use of terms defined in [RFC8499].

"IPv6-only host" refers to a host with an IPv6-only connectivity.

3. Option Format

The format of the PREFIX64 EDNS0 option is shown in Figure 2. This format adheres to the guidelines specified in Section 6.1.2 of [RFC6891].

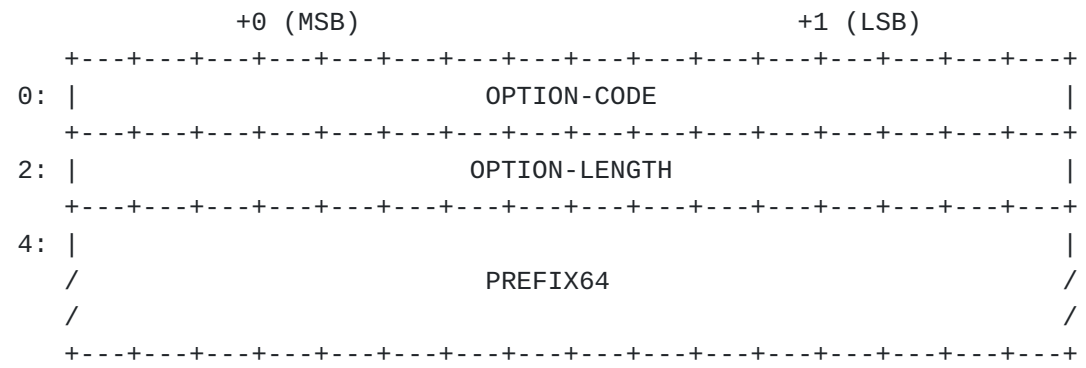

Figure 2: PREFIX64 EDNS0 Option Format

The description of the fields is as follows:

OPTION-CODE: MUST be set to TBA (Section 6).

OPTION-LENGTH: Size (in octets) of the enclosed Pref64::/n. Allowed values are: 0, 4, 5, 6, 7, 8, and 9.

The receiver MUST ignore the option if the OPTION-LENGTH is not set to one of those values.

When the value is set to 0, this indicates the presence of a NAT64 function in the network from which the query is generated.

PREFIX64: This field identifies the IPv6 unicast prefix to be used for constructing an IPv4-converted IPv6 address from an IPv4 address as specified in Section 2.2 of [RFC6052]. In such case, the prefix length MUST be 32, 40, 48, 56, 64, and 96 bits (i.e., OPTION-LENGTH must be set to 4, 5, 6, 7, 8, and 9) as specified in [RFC6052].

This prefix can be the Well-Known Prefix (i.e., 64:ff9b::/96) or a Network-Specific Prefix.

The address synthesis MUST follow the guidelines documented in [RFC6052].

4. Protocol Description

A stub-resolver on an IPv6-only host that discovers the presence of NAT64 inserts the PREFIX64 EDNS0 option in its AAAA queries. If the stub-resolver is on a multi-interfaced device, the Pref64::/n conveyed in the PREFIX64 EDNS0 option MUST be the one that is associated with the interface over which the DNS query is sent.

A stub-resolver that is prepared to handle A RRs enclosed in the additional section (e.g., security-aware and validating hosts) MAY insert a PREFIX64 EDNS0 option with an OPTION-LENGTH set to zero in its AAAA DNS queries. Such option is used by intermediate/ authoritative servers as a signal to include A RR in the additional section.

If a DNS server enables a DNS64 function, then the AAAA query is treated as in [RFC6147] with the exception that supplied valid Pref64::/n are used for synthesizing AAAA records. The reply MAY echo the PREFIX64 EDNS0 option.

A DNS forwarder MAY be configured to forward AAAA queries that carry an PREFIX64 EDNS0 option with non-null prefixes to a DNS64 server. Such queries are thus relayed to that DNS64 server. Upon receipt of such queries, the AAAA query is treated as in [RFC6147] with the exception that supplied valid Pref64::/n are used for synthesizing AAAA records. This prevents from exposing distinct IP addresses of DNS servers for "normal" DNS and DNS64 operations.

A DNS64 MAY be instructed to return the Pref64::/n that it uses when synthesizing AAAA records. If so, the DNS64 MUST include the PREFIX64 option in its replies that carry synthesized AAAA records. This is superior to the current situation where users have to check the documentation (when available) to determine the prefix used by a DNS64 server for address synthesis. Absent such checks, errors can be encountered to service IPv6-only hosts. The use of PREFIX64 option allows to automatically detect mismatches between the prefix used in the network (that is, the NAT64 function) and the one that is used by a DNS64 function.

A stub-resolver SHOULD determine whether the returned AAAA includes a native or IPv4-converted IPv6 by comparing the first bits of the IPv6 address with the local Pref64::/n. This check is also meant to determine that an on-path attacker has modified the PREFIX64 option.

5. Security Considerations

Generic EDNS0 security considerations are discussed in Section 8 of [RFC6891].

As discussed in Section 5.5 of [RFC6147], a security-aware and validating host has to perform the DNS64 function locally. This specification does not prevent that. The only enhancement is the receipt of A RRs in the additional section of AAAA replies.

6. IANA Considerations

This document requests IANA to assign the following new code from the "DNS EDNS0 Option Codes (OPT)" registry available at [DNS-OPT]:

Value	Name	Status	Reference
TBA	PREFIX64	Standard	[ThisDocument]

7. Acknowledgements

Thanks to Tiru Reddy for the comments.

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>.

[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/info/rfc6052>.

[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>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/

RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

[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

[DNS-OPT] IANA, "DNS EDNS0 Option Codes (OPT)", <http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-11>.

[RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <https://www.rfc-editor.org/info/rfc6887>.

[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>.

[RFC7051] Korhonen, J., Ed. and T. Savolainen, Ed., "Analysis of Solution Proposals for Hosts to Learn NAT64 Prefix", RFC 7051, DOI 10.17487/RFC7051, November 2013, <https://www.rfc-editor.org/info/rfc7051>.

[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/info/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/info/rfc8499>.

[RFC8683] Palet Martinez, J., "Additional Deployment Guidelines for NAT64/464XLAT in Operator and Enterprise Networks", RFC 8683, DOI 10.17487/RFC8683, November 2019, <https://www.rfc-editor.org/info/rfc8683>.

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

Author's Address

Mohamed Boucadair
Orange

35000 Rennes
France

Email: mohamed.boucadair@orange.com