intarea Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 27, 2017

L. Li
Y. Cui
C. Liu
J. Wu
Tsinghua University
F. Baker

J. Palet Martinez
Consulintel, S.L.

March 26, 2017

DHCPv6 Options for Discovery NAT64 Prefixes
draft-li-intarea-nat64-prefix-dhcp-option-01

Abstract

Several IPv6 transition mechanisms require the usage of stateless or stateful translators (commonly named as NAT64) able to allow IP/ICMP communication between IPv4 and IPv6 networks.

Those translators are using either a default Well-Known Prefix (WKP), and/or one or several additional Network Specific Prefix (NSP), which need to be configured into the nodes willing to use the translator. Different translators will likely have different IPv6 prefixes, to attract traffic to the correct translator. Thus, an automatic translator prefix discovery method is necessary.

This document defines a DHCPv6-based method to inform DHCPv6 clients the set of IPv6 and IPv4 prefixes it serves. This DHCPv6 option can be used by several transition mechanisms such as SIIT, 464XLAT, EAM.

Table of Contents

1. Introduction

Stateless IP/ICMP Translation (SIIT) [RFC7915] describes the basic translation mechanism (NAT64), which is actually used as the base for most of the related translation protocols.

Stateful NAT64 [RFC6146] describes how to allow IPv6-only clients to contact IPv4 servers using unicast UDP, TCP or ICMP.

464XLAT [RFC6877] describes an IPv4-over-IPv6 solution as one technique for IPv4 service extension and encouragement of IPv6 deployment. The 464XLAT architecture uses IPv4/IPv6 translation, described in [RFC6144], and standardized in [RFC6052], [RFC7915], and [RFC6146]. It encourages the IPv6 transition by making IPv4 service reachable across IPv6-only networks and providing IPv6 and IPv4 connectivity to single-stack IPv4 or IPv6 servers and peers. In the

464XLAT architecture, the CLAT (customer-side NAT46 translator) must determine which of potentially several PLAT (provider-side NAT64 translator) IPv6 prefix to use in order to send a packet to the PLAT with connectivity to its destination.

[RFC7050] describes a mechanism to learn the PLAT-side IPv6 prefix for protocol translation by DNS64 [RFC6147]. Although it supports multiple PLAT-side prefix by responding with multiple AAAA records to a DNS64 query, it does not support mapping IPv4 prefixes to IPv6 prefix, which would be required, for example, if one PLAT has connectivity to the general Internet following a default route, another has connectivity to a BGP peer, and a third has connectivity to a network using private addressing [RFC1918]. Therefore, in the scenario with multiple PLATs, [RFC7050] does not directly support destination-based IPv4 routing among PLATs; instead, the DNS64 database must contain equivalent information. It also requires the additional deployment of DNS64 service in customer-side networks, which is not required in 464XLAT deployment. Indeed, this scenario, which may become very common in wired access networks, has not even been considered by [RFC7051].

464XLAT is in fact, a usage case of Stateful NAT64.

Explicit Address Mappings for Stateless IP/ICMP Translation [RFC7757] extends SIIT with an Explicit Address Mapping (EAM) algorithm to facilitate stateless IP/ICMP translation between arbitrary (non-IPv4-translatable) IPv6 endpoints and IPv4.

This document proposes a method for the translator (NAT64) IPv6 prefix discovery based on DHCPv6, which is widely deployed and supported in customer networks. It defines two new DHCPv6 options for use by a DHCPv6 client to discover the translator IPv6 prefix(es). Also, the proposed mechanism can deal with the scenario with multiple independent DNS64 databases supporting separate translators.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. New DHCPv6 Option

3.1. NAT64 Prefix List Option Format

The NAT64 Prefix List Option is a container for NAT64 Prefix Option(s). A NAT64 Prefix List Option MAY contain multiple NAT64 Prefix Options.

The format of the NAT64 Prefix List Option is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   OPTION_NAT64_PREFIX_LIST    |       option-length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                      NAT64_PREFIX-options                     +
|                        (see Figure 2)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Format of NAT64 Prefix List Option

- o option-code: OPTION_NAT64_PREFIX_LIST (TBA1)
- o option-length: length of NAT64_PREFIX-options, specified in octets.
- o NAT64_PREFIX-options: one or more OPTION_NAT64_PREFIX options.

3.2. NAT64 Prefix Option Format

The NAT64 Prefix Option is encapsulated in the NAT64 Prefix List Option. This option allows the mapping of destination IPv4 address ranges (contained in the IPv4 Prefix List) to a NAT64 IPv6 prefix. If there is more than one such prefix, each prefix comes in its own option, with its associated IPv4 prefix list. In this way, the DHCPv6 client can select the NAT64 with the corresponding destination IPv4 address.

The format of the NAT64 Prefix Option is:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     OPTION_NAT64_PREFIX      |         option-length          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         NAT64-Type           |         NAT64-prelen           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         NAT64-prefix                          |
    |                       (variable length)                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         NAT64-suffix                          |
    |                       (variable length)                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                         (optional)                            .
    .             IPv4 Prefix List (variable length)                .
    .                       (see Figure 3)                          .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Format of NAT64 Prefix Option

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  IPv4-prelen  |            IPv4 Prefix (32 bits)              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    (cont.)    |  IPv4-prelen  |     IPv4 Prefix (32 bits)     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      IPv4 Prefix (cont.)      |              ...              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                              ...                              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Format of IPv4 Prefix List field

- option-code: OPTION_NAT64_PREFIX (TBA2)

- type-field: NAT64-Type (TBA3)

- option-length: 4 + length of NAT64-prefix + length of NAT64-suffix + length of IPv4 Prefix List, specified in octets.

- NAT64-prelen: length of NAT64-prefix, in octets. The values allowed are: 4, 5, 6, 7, 8 or 12, as indicated in Section 2.2 of [RFC6052].

- NAT64-prefix: The NAT64 IPv6 prefix to be used by the DHCPv6 client for the IPv6 address synthesis. MUST follow the specifications of [RFC6052].

- NAT64-suffix: The NAT64 suffix to be used by the DHCPv6 client for the IPv6 address synthesis, specified in Section 2.2 of [RFC6052]. The length of this field, in octets is 12 - NAT64-prelen. Not used in case of using the WKP (i.e., 64:ff9b::/96) or any other NSP (Network-Specific Prefix) which a length of 96 bits (/96).

- IPv4 Prefix List: This is an optional field. The format of the IPv4 Prefix List is shown in Figure 3. It is a list of zero or more IPv4 Prefixes. Each entry is formed by IPv4-prelen and IPv4 Prefix. The total length of the field is 5*number of IPv4 prefixes.

- IPv4-prelen: The length of the IPv4 Prefix.

- IPv4 Prefix: The destination-based IPv4 Prefix. The length is 4 octets.

4. Client Behavior

The client requests the OPTION_NAT64_PREFIX_LIST option using the Option Request option (ORO) in every Solicit, Request, Renew, Rebind, and Information-request message. The NAT64-Type field defines the mechanism being used. If the DHCPv6 server includes the OPTION_NAT64_PREFIX_LIST option in its response, the DHCPv6 client may use the contained NAT64-prefix to translate the destination IPv4 address into the destination IPv6 address.

When receiving the OPTION_NAT64_PREFIX option with IPv4 Prefix List, the DHCPv6 client MUST record the received IPv6 prefix and the corresponding IPv4 prefixes in IPv4 Prefix List. When receiving the OPTION_NAT64_PREFIX option without IPv4 Prefix List, the DHCPv6 client MUST treat the IPv6 prefix and the default IPv4 prefix 0.0.0.0/0 as one of the records.

If the DHCPv6 client loses contact with the DHCPv6 server, the DHCPv6 client SHOULD clear the prefix(es) it learned from the DHCPv6 server.

When translating the destination IPv4 address into the destination IPv6 address, DHCPv6 client MUST search an IPv4 routing database using the longest-match-first rule and select the IPv6 prefix offering that IPv4 prefix.

In order to make sure that the client has the right information, only a single container with all the current NAT64 Prefix List Option is allowed. When a new one is received, the client MUST clear the previous NAT64 Prefix List Option information and use only the new one.

5. Message Flow Illustration

The figure below shows an example of message flow for a Client learning IPv6 prefixes using DHCPv6.

In this example, two IPv6 prefixes are provided by the DHCPv6 server. The first IPv6 prefix is 2001:db8:122:300::/56, the corresponding IPv4 prefixes are 192.0.2.0/24 and 198.51.100.0/24. The second IPv6 prefix is 2001:db8:122::/48, the corresponding IPv4 prefix is 192.0.2.128/25.

When the DHCPv6 client receives the packet with destination IPv4 address 192.0.2.1, according to the rule of longest prefix match, the NAT64 with IPv6 prefix 2001:db8:122::/48 is chosen. In the same way, the NAT64 with IPv6 prefix 2001:db8:122::/48 is chosen.

```
   +---------------+                                +-----------------+
   | DHCPv6 Client |                                |  DHCPv6 server  |
   +---------------+                                +-----------------+
        |            DHCPv6 query for IPv6 prefix            |
        |-------------------------------------------------->|
        |   ORO with OPTION_NAT64_PREFIX_LIST                |
        |                                                    |
        |   DHCPv6 response with:                            |
        |      NAT64PREFIX{                                  |
        |         NAT64-v6-pre = 2001:db8:122:300::/56       |
        |         NAT64-v4-pre = 192.0.2.0/24                |
        |         NAT64-v4-pre = 198.51.100.0/24}            |
        |      NAT64PREFIX{                                  |
        |         NAT64-v6-pre = 2001:db8:122::/48           |
        |         NAT64-v4-pre = 192.0.2.128/25}             |
        |<--------------------------------------------------|
        |                                                    |
        |
        |                     +-----------------+   +-----------------+
        |                     |     NAT64 1     |   |     NAT64 2     |
        |                     +-----------------+   +-----------------+
        |                        NAT64-v6-pre =        NAT64-v6-pre =
        |                    2001:db8:122:300::/56  2001:db8:122::/48
        |                        NAT64-v4-pre =        NAT64-v4-pre =
        |                        192.0.2.0/24          192.0.2.128/25
        |                        198.51.100.0/24             |
        |                              |                     |
        |       Dest IPv4 addr:        |                     |
        |        192.0.2.1             |                     |
        |       Dest IPv6 addr:        |                     |
        |  2001:db8:122:300::c000:201  |                     |
        |----------------------------->|                     |
        |                              |                     |
        |                                                    |
        |     Dest IPv4 addr: 192.0.2.193                    |
        |     Dest IPv6 addr: 2001:db8:122::c000:2c1         |
        |-------------------------------------------------->|
```

Figure 4: Example of the process flow

6. Security Considerations

Considerations for security in this type of environment are primarily around the operation of the DHCPv6 protocol and the databases it uses.

In the DHCPv6 server, should the database be compromised, it will deliver incorrect data to its DHCPv6 clients. In the DHCPv6 client,

should its database be compromised by attack or polluted by an incorrect DHCPv6 server database, it will route data incorrectly. In both cases, the security of the systems and their databases in an operational matter, not managed by protocol.

However, the operation of the DHCPv6 protocol itself is also required to be correct - the server and its clients must recognize valid requests and reject invalid ones. Therefore, DHCPv6 exchanges MUST be secured as described in [RFC3315].

7. IANA Considerations

We request that IANA allocate two DHCPv6 option codes for use by OPTION_V6_PLATPREFIX_LIST and OPTION_V6_PLATPREFIX from the "Option Codes" table. Similarly, a request to IANA for assigning the NAT64-Type field codes. The following initial values are assigned in this document (values are 16-bit unsigned intergers).

Name	Value	RFC
Unspecified	0x00	RFC6052
SIIT	0x01	RFC7915
Stateful NAT64	0x02	RFC6146
EAM-SIIT	0x03	RFC7757

8. Acknowledgements

The authors will like to recognize the inputs from Tore Anderson in a previous version of this work. Mohamed Boucadair provided very significant inputs.

9. References

9.1. 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>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <http://www.rfc-editor.org/info/rfc3315>.

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

9.2. Informative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <http://www.rfc-editor.org/info/rfc1918>.

[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>.

[RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144, April 2011, <http://www.rfc-editor.org/info/rfc6144>.

[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>.

[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>.

[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, <http://www.rfc-editor.org/info/rfc7051>.

[RFC7757] Anderson, T. and A. Leiva Popper, "Explicit Address Mappings for Stateless IP/ICMP Translation", RFC 7757, DOI 10.17487/RFC7757, February 2016, <http://www.rfc-editor.org/info/rfc7757>.

[RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <http://www.rfc-editor.org/info/rfc7915>.

Authors' Addresses

Lishan Li
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-15201441862
Email: lilishan9248@126.com

Yong Cui
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

Cong Liu
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: gnocuil@gmail.com

Jianping Wu
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5983
Email: jianping@cernet.edu.cn

Fred Baker
Goleta, CA 93117
United States

Email: fredbaker.ietf@gmail.com

Jordi Palet Martinez
Consulintel, S.L.
La Navata - Galapagar 28420
Spain

Email: jordi.palet@consulintel.es