ï»¿

               Discovering PREF64 in Router Advertisements

Abstract

   This document specifies a Neighbor Discovery option to be used in
   Router Advertisements (RAs) to communicate prefixes of Network
   Address and Protocol Translation from IPv6 clients to IPv4 servers
   (NAT64) to hosts.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   https://www.rfc-editor.org/info/rfc8781.

Table of Contents

1. Introduction

   NAT64 [RFC6146] with DNS Extensions for Network Address Translation

from IPv6 clients to IPv4 servers (DNS64) [RFC6147] is a widely
deployed mechanism to provide IPv4 access on IPv6-only networks. In
various scenarios, the host must be aware of the NAT64 prefix in use
by the network. This document specifies a Neighbor Discovery
[RFC4861] option to be used in Router Advertisements (RAs) to
communicate NAT64 prefixes to hosts.

1.1. Requirements Language

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.

1.2. Terminology

PREF64 (or NAT64 prefix): An IPv6 prefix used for IPv6 address
    synthesis [RFC6146];

NAT64: Network Address and Protocol Translation from IPv6 clients to
    IPv4 servers [RFC6146];

Router Advertisement (RA): A message used by IPv6 routers to
    advertise their presence together with various link and Internet
    parameters [RFC4861];

DNS64: a mechanism for synthesizing AAAA records from A records
[RFC6147];

2. Use Cases for Communicating the NAT64 Prefix to Hosts

On networks employing NAT64, it is useful for hosts to know the NAT64
prefix for several reasons, including the following:

*   Enabling DNS64 functions on end hosts. In particular:

    -   Local DNSSEC validation (DNS64 in stub-resolver mode). As
        discussed in [RFC6147], Section 2, the stub resolver in the
        host "will try to obtain (real) AAAA RRs, and in case they are
        not available, the DNS64 function will synthesize AAAA RRs for
        internal usage." Therefore, to perform the DNS64 function, the
        stub resolver needs to know the NAT64 prefix. This is required
        in order to use DNSSEC on a NAT64 network.

    -   Trusted DNS server. AAAA synthesis is required for the host to
        be able to use a DNS server not provided by the network (e.g.,
        a DNS-over-TLS [RFC7858] or DNS-over-HTTPS [RFC8484] server
        with which the host has an existing trust relationship).

    -   Networks with no DNS64 server. Hosts that support AAAA
        synthesis and are aware of the NAT64 prefix in use do not need
        the network to perform the DNS64 function at all.

*   Enabling NAT64 address-translation functions on end hosts. For
    example:

    -   IPv4 address literals on an IPv6-only host. As described in
        [RFC8305], Section 7.1, IPv6-only hosts connecting to IPv4
        address literals can translate the IPv4 literal to an IPv6
        literal.

    -   464XLAT [RFC6877]. 464XLAT requires the host be aware of the
        NAT64 prefix.

3. Why Include the NAT64 Prefix in Router Advertisements?

Fate sharing: NAT64 requires routing to be configured. IPv6 routing
    configuration requires receiving an IPv6 RA [RFC4861]. Therefore,
    using RAs to provide hosts with the NAT64 prefix ensures that
    NAT64 reachability information shares the fate of the rest of the

network configuration on the host.

   Atomic configuration: Including the NAT64 prefix in the RA minimizes
      the number of packets required to configure a host. Only one
      packet (an RA) is required to complete the network configuration.
      This speeds up the process of connecting to a network that
      supports NAT64/DNS64. It also simplifies host implementation by
      removing the possibility that the host can have an incomplete
      Layer 3 configuration (e.g., IPv6 addresses and prefixes, but no
      NAT64 prefix).

   Updatability: It is possible to change the NAT64 prefix at any time,
      because when it changes, it is possible to notify hosts by sending
      a new RA.

   Deployability: All IPv6 hosts and networks are required to support
      Neighbor Discovery [RFC4861] so just a minor extension to the
      existing implementation is required. Other options, such as
      [RFC7225], require implementing other protocols (e.g., Port
      Control Protocol (PCP) [RFC7225]), which could be considered an
      obstacle for deployment.

4. Option Format

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |    Length     |    Scaled Lifetime    | PLC |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                                                               +
   |                  Highest 96 bits of the Prefix                |
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                   Figure 1: NAT64 Prefix Option Format

   Fields:

   Type: 8-bit identifier of the PREF64 option type (38)

   Length: 8-bit unsigned integer. The length of the option (including
      the Type and Length fields) is in units of 8 octets. The sender
      MUST set the length to 2. The receiver MUST ignore the PREF64
      option if the Length field value is not 2.

   Scaled Lifetime: 13-bit unsigned integer. The maximum time in units
      of 8 seconds over which this NAT64 prefix MAY be used. See
      Section 4.1 for the Scaled Lifetime field processing rules.

   PLC (Prefix Length Code): 3-bit unsigned integer. This field
      encodes the NAT64 Prefix Length defined in [RFC6052]. The PLC
      field values 0, 1, 2, 3, 4, and 5 indicate the NAT64 prefix length
      of 96, 64, 56, 48, 40, and 32 bits, respectively. The receiver
      MUST ignore the PREF64 option if the Prefix Length Code field is
      not set to one of those values.

   Highest 96 bits of the Prefix: 96-bit unsigned integer. Contains
      bits 0 - 95 of the NAT64 prefix.

4.1. Scaled Lifetime Processing

   It would be highly undesirable for the NAT64 prefix to have a
   lifetime shorter than the Router Lifetime, which is defined in
   Section 4.2 of [RFC4861] as a 16-bit unsigned integer. If the NAT64
   prefix lifetime is not at least equal to the default Router Lifetime,
   it might lead to scenarios in which the NAT64 prefix lifetime expires
   before the arrival of the next unsolicited RA. Therefore, the Scaled
   Lifetime encodes the NAT64 prefix lifetime in units of 8 seconds.
   The receiver MUST multiply the Scaled Lifetime value by 8 (for

example, by a logical left shift) to calculate the maximum time in
seconds the prefix MAY be used. The maximum lifetime of the NAT64
prefix is thus 65528 seconds. To ensure that the NAT64 prefix does
not expire before the default router, it is NOT RECOMMENDED to
configure default Router Lifetimes greater than 65528 seconds when
using this option. A lifetime of 0 indicates that the prefix SHOULD
NOT be used anymore.

By default, the value of the Scaled Lifetime field SHOULD be set to
the lesser of 3 x MaxRtrAdvInterval [RFC4861] divided by 8, or 8191.

Router vendors SHOULD allow administrators to specify nonzero
lifetime values that are not divisible by 8. In such cases, the
router SHOULD round the provided value up to the nearest integer that
is divisible by 8 and smaller than 65536, then divide the result by 8
(or perform a logical right shift by 3) and set the Scaled Lifetime
field to the resulting value. If a nonzero lifetime value that is to
be divided by 8 (or subjected to a logical right shift by 3) is less
than 8, then the Scaled Lifetime field SHOULD be set to 1. This last
step ensures that lifetimes under 8 seconds are encoded as a nonzero
Scaled Lifetime.

5. Usage Guidelines

This option specifies exactly one NAT64 prefix for all IPv4
destinations. If the network operator wants to route different parts
of the IPv4 address space to different NAT64 devices, this can be
accomplished by routing more specific subprefixes of the NAT64 prefix
to those devices. For example, suppose an operator is using the
[RFC1918] address space 10.0.0.0/8 internally. That operator might
want to route 10.0.0.0/8 through NAT64 device A, and the rest of the
IPv4 space through NAT64 device B. If the operator's NAT64 prefix is
2001:db8:a:b::/96, then the operator can route
2001:db8:a:b::a00:0/104 to NAT64 A and 2001:db8:a:b::/96 to NAT64 B.

This option may appear more than once in an RA (e.g., when gracefully
renumbering the network from one NAT64 prefix to another). Host
behavior with regard to synthesizing IPv6 addresses from IPv4
addresses SHOULD follow the recommendations given in Section 3 of
[RFC7050], limited to the NAT64 prefixes that have a nonzero
lifetime.

In a network (or a provisioning domain) that provides both IPv4 and
NAT64, it may be desirable for certain IPv4 addresses not to be
translated. An example might be private address ranges that are
local to the network/provisioning domain and that should not be
reached through the NAT64. This type of configuration cannot be
conveyed to hosts using this option, or through other NAT64 prefix
provisioning mechanisms such as [RFC7050] or [RFC7225]. This problem
does not apply in IPv6-only networks: the host in an IPv6-only
network does not have an IPv4 address and cannot reach any IPv4
destinations without the NAT64.

5.1. Handling Multiple NAT64 Prefixes

In some cases, a host may receive multiple NAT64 prefixes from
different sources. Possible scenarios include (but are not limited
to):

* the host is using multiple mechanisms to discover PREF64 prefixes
   (e.g., by using PCP [RFC7225]) and/or resolving an IPv4-only fully
   qualified domain name [RFC7050] in addition to receiving the
   PREF64 RA option);

* the PREF64 option presents in a single RA more than once;

* the host receives multiple RAs with different PREF64 prefixes on a
   given interface.

When multiple PREF64s are discovered via the RA PREF64 Option (either
the Option presents more than once in a single RA or multiple RAs are

received), host behavior with regard to synthesizing IPv6 addresses
from IPv4 addresses SHOULD follow the recommendations given in
Section 3 of [RFC7050], limited to the NAT64 prefixes that have a
nonzero lifetime.

When different PREF64s are discovered using multiple mechanisms,
hosts SHOULD select one source of information only. The RECOMMENDED
order is:

* PCP-discovered prefixes [RFC7225], if supported;

* PREF64s discovered via the RA Option;

* PREF64s resolving an IPv4-only fully qualified domain name
   [RFC7050]

Note: If the network provides PREF64s via both this RA Option and
[RFC7225], hosts that receive the PREF64 via the RA Option may choose
to use it immediately (before waiting for the PCP to complete);
therefore, some traffic may not reflect any more detailed
configuration provided by the PCP.

The host SHOULD treat the PREF64 as being specific to the network
interface it was received on. Hosts that are aware of Provisioning
Domain (PvD, [RFC7556]) MUST treat the PREF64 as being scoped to the
implicit or explicit PvD.

5.2. PREF64 Consistency

Section 6.2.7 of [RFC4861] recommends that routers inspect RAs sent
by other routers to ensure that all routers onlink advertise
consistent information. Routers SHOULD inspect valid PREF64 options
received on a given link and verify the consistency. Detected
inconsistencies indicate that one or more routers might be
misconfigured. Routers SHOULD log such cases to system or network
management. Routers SHOULD check and compare the following
information:

* set of PREF64s with a nonzero lifetime;

* set of PREF64s with a zero lifetime.

Routers that are aware of PvD ([RFC7556]) MUST only compare
information scoped to the same implicit or explicit PvD.

6. IANA Considerations

IANA has assigned a new IPv6 Neighbor Discovery Option type for the
PREF64 option defined in this document in the "IPv6 Neighbor
Discovery Option Formats" registry [IANA].

Description	Type
===============	======
PREF64 option	38

Table 1: New IANA
Registry Assignment

7. Security Considerations

Because RAs are required in all IPv6 configuration scenarios, on
IPv6-only networks, RAs must already be secured -- e.g., by deploying
an RA-Guard [RFC6105]. Providing all configuration in RAs reduces
the attack surface to be targeted by malicious attackers trying to
provide hosts with invalid configuration, as compared to distributing
the configuration through multiple different mechanisms that need to
be secured independently.

If a host is provided with an incorrect NAT64 prefix, the IPv6-only

host might not be able to communicate with IPv4-only destinations.
Connectivity to destinations reachable over IPv6 would not be
impacted just by providing a host with an incorrect prefix; however,
if attackers are capable of sending rogue RAs, they can perform
denial-of-service or man-in-the-middle attacks, as described in
[RFC6104].

The security measures that must already be in place to ensure that
RAs are only received from legitimate sources eliminate the problem
of NAT64 prefix validation described in Section 3.1 of [RFC7050].

8. References

8.1. Normative References

[IANA]   IANA, "Internet Control Message Protocol version 6
            (ICMPv6) Parameters",
            <https://www.iana.org/assignments/icmpv6-parameters>.

[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>.

[RFC4861]   Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
            "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
            DOI 10.17487/RFC4861, September 2007,
            <https://www.rfc-editor.org/info/rfc4861>.

[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>.

[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>.

[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

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

[RFC6104]   Chown, T. and S. Venaas, "Rogue IPv6 Router Advertisement
            Problem Statement", RFC 6104, DOI 10.17487/RFC6104,
            February 2011, <https://www.rfc-editor.org/info/rfc6104>.

[RFC6105]   Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J.
            Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105,
            DOI 10.17487/RFC6105, February 2011,
            <https://www.rfc-editor.org/info/rfc6105>.

[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>.

[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/info/rfc6877>.

   [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>.

   [RFC7556]   Anipko, D., Ed., "Multiple Provisioning Domain
               Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015,
               <https://www.rfc-editor.org/info/rfc7556>.

   [RFC7858]   Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D.,
               and P. Hoffman, "Specification for DNS over Transport
               Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May
               2016, <https://www.rfc-editor.org/info/rfc7858>.

   [RFC8305]   Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2:
               Better Connectivity Using Concurrency", RFC 8305,
               DOI 10.17487/RFC8305, December 2017,
               <https://www.rfc-editor.org/info/rfc8305>.

   [RFC8484]   Hoffman, P. and P. McManus, "DNS Queries over HTTPS
               (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018,
               <https://www.rfc-editor.org/info/rfc8484>.

Authors' Addresses

   Lorenzo Colitti
   Google
   Shibuya 3-21-3, Tokyo
   150-0002
   Japan

   Email: lorenzo@google.com

   Jen Linkova
   Google
   1 Darling Island Rd
   Pyrmont NSW 2009
   Australia

   Email: furry@google.com