Learning the IPv6 Prefix of an IPv6/IPv4 Translator
draft-wing-behave-learn-prefix-02

Abstract

Some IPv6 applications obtain IPv4 address literals and want to
communicate with those IPv4 hosts through an IPv6/IPv4 translator. The IPv6 application can send an IPv6 packet through the translator if it knows the IPv6 prefix of the IPv6/IPv4 translator. In many IPv6/IPv4 translation deployments, that IPv6 prefix is not fixed; rather, the prefix is chosen by the network operator. This specification provides three methods for a host to learn the IPv6 prefix of its IPv6/IPv4 translator.

Table of Contents

1. Terminology ........................................ 3
2. Introduction ......................................... 3
3. Mechanisms to Learn the IPv6 Prefix and Length .......... 4
   3.1. Using DNS ................................... 4
   3.2. Using DHCP .................................. 5
   3.3. Using IPv6 Router Advertisements (RA) ............ 6
4. Authenticating the Learned Prefix ....................... 8
5. Security Considerations ............................. 9
6. IANA Considerations .................................. 9
7. Acknowledgements .................................... 10
8. References .......................................... 10
   8.1. Normative References ........................... 10
   8.2. Informative References ........................ 10
Appendix A. For future study .......................... 11
   A.1. multi-homed hosts ............................. 11
   A.2. Unicast and multicast translators ............... 11
Appendix B. Changes .................................. 12
   B.1. Changes from -01 to -02 ........................ 12
   B.2. Changes from -00 to -01 ........................ 12
Authors’ Addresses .................................. 12
1. Terminology

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].

AFT: Address Family Translator. A device that translates between IP address families.

DNS64: The function of synthesizing an AAAA record from an A record (also called "DNS rewriting" or "DNS-ALG"), described in [I-D.bagnulo-behave-dns64].

LIR Prefix: A prefix assigned to an IPv6/IPv4 translator that uses the same LIR (Local Internet Registry) prefix as assigned to the network by that network’s local Internet registry.

Edge router: The routers with some interfaces which attach to the same multicast link with some hosts.

2. Introduction

Certain applications, operating in certain translation scenarios, can benefit from knowing the IPv6 prefix of their IPv6/IPv4 translator. First, the host must be operating in an IPv6-initiated scenario with a local translator. The BEHAVE charter [Charter] describes these as Scenario 1, "IPv6 network to IPv4 Internet", and Scenario 5, "An IPv6 network to an IPv4 network". Learning the prefix is useful for both stateful translation and stateless translation.

With those scenarios, the IPv6 host usually performs a DNS AAAA query which is processed by a DNS64 server. The DNS64 server generates a synthetic AAAA response, when necessary. This synthetic AAAA response contains the prefix of the IPv6/IPv4 translator. When the IPv6 host sends a packet to that address returned in the AAAA response, the packet is routed to the translator which translates it to IPv4. This functionality is transparent to the IPv6 host, for the most part.

Second, the IPv6 application obtains an IPv4 address literal via a means outside of DNS, and wants to communicate with that IPv4 address. So far, the authors have identified the following applications which can benefit knowing the IPv6 prefix of the host’s IPv6/IPv4 translator:

- host-based DNSSEC validation (Section 4.3 of [I-D.bagnulo-behave-dns64])
o BitTorrent (Section 2.2 of [I-D.wing-behave-nat64-referrals])

o multicast translation (Section 4 of [I-D.venaas-behave-mcast46])

o URI schemes with host IPv4 address literals rather than domain names (e.g., http://192.0.2.1, ftp://192.0.2.1, imap://192.0.2.1, ipp://192.0.2.1)).

o In the absence of an FTP ALG in the IPv6/IPv4 translator, IPv6 FTP clients that connect to an IPv4 FTP server which does not support EPSV [RFC2428].

When an IPv6/IPv4 translator is used with an LIR prefix (rather than the well-known prefix), it is necessary for such applications to learn the IPv6 prefix (and length) of the translator so that the application can create an IPv6 packet that will be routed to the translator and be translated to IPv4.

3. Mechanisms to Learn the IPv6 Prefix and Length

Both the IPv6 prefix of the translator and the prefix length of the translator need to be learned. With that information, the application can generate an appropriate IPv6 address that will be routed to the translator for the translator to process.

The host can learn the necessary information using DNS, DHCP, or Router Alert, as described in the following sections.

Issue: If a conflict exists between DNS, DHCP, or RA, which should take precedence? Should we choose one mechanism?

3.1. Using DNS

This specification defines a new U-NAPTR [RFC4848] application to discover the translator’s IPv6 prefix and length. The input domain name is the exact same as would be used for a reverse DNS lookup, derived from the host’s IPv6 in the ".ip6.arpa." tree and follows the construction rules in Section 2.5 of [RFC3596]. This is shortened to 20 labels (representing a /64 network prefix) and, if DNS returns an error is shortened to 16 labels (representing a /48 network prefix).

If an IPv6/IPv4 translator is present on the network, the successful result of one of those queries will produce a NAPTR record with the desired service tag "TRANSLATE64:" which contains the IPv6 prefix and prefix length of the translator, separated by a "/" (the same syntax as specified in Section 2.3 of [RFC4291]).
For example, a host with the IP address 2001:db8:1:2:3:4:567:89ab would first send an NAPTR query for 
3.0.0.0.2.0.0.0.1.0.0.0.8.b.d.0.1.0.0.2.IP6.ARPA (20 elements, 
representing a /64 network prefix). If that fails (returns 
NXDOMAIN), it would send an NAPTR query for 
2.0.0.0.1.0.0.0.8.b.d.0.1.0.0.2.IP6.ARPA (16 elements, representing a 
/48 network prefix).

Note: Both /64 and /48 prefix lengths are shown in this version 
of the document for illustrative purposes. The number of elements 
of this query will depend on the prefix length(s) defined by the 
BEHAVE working group for a translator. If the BEHAVE working 
group decides that all translators will have a certain prefix 
length, then only one DNS query is sent.

If the host needs to authenticate the prefix it just learned (e.g., 
because the host is running a DNSSEC validator) the host performs the 
additional authentication steps described in Section 4.

3.2. Using DHCP

A new DHCP option, OPTION_AFT_PREFIX_DHCP, is defined. It contains 
the IPv6 prefix and its length.
option-code: OPTION_AFT_PREFIX_DHCP (TBD)
option-length: 17
prefix-length: Length for this prefix in bits
IPv6-prefix: An IPv6 prefix

Figure 1: DHCP option OPTION_AFT_PREFIX_DHCP

In order to conserve space, it is RECOMMENDED that only the significant bits of the IPv6 prefix be sent in the DHCP option.

If the host needs to authenticate the prefix it just learned (e.g., because the host is running a DNSSEC validator) the host performs the additional authentication steps described in Section 4.

3.3. Using IPv6 Router Advertisements (RA)

The IPv6 prefix and the prefix length can be advertised to IPv6 hosts by routers in Router Advertisement (RA) messages [RFC4861].
In the scenario where the IPv6 hosts attach to the same multicast link as the translator (i.e., IPv6-A in Figure 2), the translator can transmit the IPv6 prefix and the length to IPv6 hosts in the RA messages advertised periodically or in the responses to valid Router Solicitation messages.

In the scenarios where IPv6 hosts are attached to a different multicast link than the translator (i.e., IPv6-B in Figure 2), the IPv6 prefix and the prefix length could be manually configured for edge routers in the IPv6 domain such as router-B. Either the translator can use the IGP (Interior Gateway Protocol, e.g., OSPF, ISIS, RIP) to announce the IPv6 prefix and the prefix length to edge routers in the IPv6 domain. Thus edge routers can transfer the IPv6 prefix and the length to IPv6 hosts in the RA messages advertised periodically or in the responses to valid Router Solicitation messages.

This mechanism requires extensions to both the RA message of Neighbor Discovery protocol [RFC4861] and IGP allowing the IPv6 prefix and prefix length to be communicated to IPv6 hosts or routers.

In the extension of RA messages, a new option type, OPTION_AFT_PREFIX_RA, is defined. It contains the IPv6 prefix and its length.
Extending OSPF requires defining a new TLV type, TLV_AFT_PREFIX, of Router Information LSA (Link State Advertisement) to transfer the IPv6 prefix and the prefix length. The format of TLV_AFT_PREFIX is the same as OPTION_AFT_PREFIX_DHCP of Section 3.2.

Extending other IGPs (e.g., ISIS, RIP) will be discussed in a future version of this document.

If the host needs to authenticate the prefix it just learned (e.g., because the host is running a DNSSEC validator) the host performs the additional authentication steps described in Section 4.

4. Authenticating the Learned Prefix

In some cases (e.g., a host performing DNSSEC validation), the host needs to authenticate the translator’s IPv6 prefix learned via one of the mechanisms described earlier. To allow such authentication the operator of the translator first creates a PTR record for the translator (with 0’s for the elements after the translator’s IPv6 prefix) which points to a hostname. The hostname has a signed AAAA record for the same 0-padded IPv6 address returned by the PTR query. Once those configuration steps are done, a host can validate the translator’s IPv6 prefix by performing the following steps:

a. The host sends a DNS PTR query for the IPv6 address of the translator (for "ipv6.arpa"), using 0 for the elements after the prefix length. This will return the fully-qualified hostname of that translator device.
b. Verify the full-qualified hostname is on the host’s configured list of authorized translators (e.g., seattle.translator.example.net).

c. Send a DNS AAAA query for that hostname.

d. Verify the AAAA response matches the IPv6 address obtained in step 1.

e. Perform DNSSEC validation of the AAAA response.

For example, if the translator’s IPv6 prefix length is /48, the host would send a PTR query for 2.0.0.1.0.0.0.0.0.0.0.0.1.2.3.4.IP6.ARPA which would return a hostname, seattle.translator.example.net. The host verifies that seattle.translator.example.net is on its configured list of authorized translators, as maintained in a text file. The host sends an AAAA query for seattle.translator.example.net and verifies the AAAA response contains the same IPv6 address. The host then validates the DNSSEC signature for seattle.translator.example.net.

5. Security Considerations

After learning the IPv6 prefix of its translator by following the procedures in this specification, the IPv6 host will utilize this information for subsequent actions (e.g., sending a packet to it, or using that information to synthesize DNS records or to perform DNSSEC validation). If an attacker provides a fraudulent IPv6 to the IPv6 host, the attacker can become on-path for traffic to/from that IPv6 host and preform passive or active eavesdropping or traffic analysis. To protect against this attack, it is RECOMMENDED that IPv6 hosts be configured with the names of authorized translators and RECOMMENDED that IPv6 hosts uses DNSSEC to validate that name matches the IPv6 prefix learned via DNS, DHCPv6 or RA message, as described in Section 4.

6. IANA Considerations

A new DHCPv6 option, OPTION_AFT_PREFIX_DHCP, and RA option, OPTION_AFT_PREFIX_RA, needs to be assigned by IANA.

A new TLV type, TLV_AFT_PREFIX, of Router Information LSA for OSPF needs to be assigned by IANA.

The new NAPTR Application Service tag "TRANSLATE64" is registered with IANA.
7. Acknowledgements

This draft was fostered by discussion on the 46translation mailing list and at the v4v6 Interim in Montreal. Special thanks to Iljitsch van Beijnum, Andrew Sullivan, Marcelo Bagnulo Braun, Fred Baker, and Xing Li for their comments and dialog.

The mechanism to perform a shortened NAPTR query was described first by Martin Thomson [I-D.thomson-geopriv-res-gw-lis-discovery].

Thanks to Ralph Droms for his help with DHCPv6. Thanks to John Schnizlein for improving the DNS learning algorithm. Thanks to Keith Moore and Scott Brim for suggesting HTTP IPv4 address literals.

8. References

8.1. Normative References


8.2. Informative References


[I-D.thomson-geopriv-res-gw-lis-discovery]
Appendix A.  For future study

A.1.  Multi-homed hosts

A multi-homed host may have different translation devices available on each of its networks, and can learn those via DNS, DHCP, or RA.

When using DNS to learn the translator’s prefix (Section 3.1) or using DNS to authenticate the translator prefix (Section 4), it is possible a split horizon DNS exists. Such a split DNS requires the host to query the DNS server associated with that network prefix as described in [I-D.savolainen-mif-dns-server-selection].

A.2.  Unicast and multicast translators

It may be necessary to use different prefixes for unicast, any source multicast (ASM), and source-specific multicast (SSM) (Section 2 of [I-D.venaas-behave-mcast46]).
Appendix B. Changes

B.1. Changes from -01 to -02

- provided another method of using RA message for a host to learn its translator’s IPv6 prefix and length
- added IPv4 address literals in URIs and multicast as benefactors for learning the translator’s prefix.
- added FTP interworking using PASV
- clarified which Scenarios this applies to, and that this is for stateful and stateless.

B.2. Changes from -00 to -01

- made clearer this is for NAT64 prefix (changed title and some text).
- changed from querying for "_aft_prefix" TXT record to querying ipv6.arpa NAPTR record.
- BitTorrent is another application that benefits from knowing the NAT64 prefix; previously only DNSSEC was listed.
- changed to standards track.

Authors’ Addresses

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA  95134
USA

Email:  dwing@cisco.com

Xuewei Wang
Huawei Technologies Co.,Ltd
No.9 Xinxi Rd.
Beijing, Hai-Dian District  100085
P.R. China

Email:  wangxuewei@huawei.com
Xiaohu Xu
Huawei Technologies Co., Ltd
No.9 Xinxi Rd.
Beijing, Hai-Dian District  100085
P.R. China

Email:  xuxh@huawei.com