Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 5, 2021

F. Templin, Ed.
Boeing Research & Technology
January 1, 2021

IPv6 Prefix Delegation and Multi-Addressing Models
draft-templin-v6ops-pdhost-27

Abstract

Requesting nodes typically acquire IPv6 prefixes from a prefix delegation service for the network. The requesting node can provision the prefix according to whether it acts as a router on behalf of any downstream networks and/or as a host on behalf of its local applications. In the latter case, the requesting node can use portions of the delegated prefix for its own multi-addressing purposes. This document therefore considers prefix delegation models for both the classic routing and various multi-addressing use cases.

Status of This Memo

This Internet-Draft will expire on July 5, 2021.

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 6
3. Multi-Addressing Considerations 6
4. Multi-Addressing Alternatives for Delegated Prefixes 7
5. Address Autoconfiguration Considerations 8
6. MLD/DAD Implications 8
7. Dynamic Routing Protocol Implications 9
8. IPv6 Neighbor Discovery Implications 9
9. Prefix Delegation Services 9
10. IANA Considerations 10
11. Security Considerations 10
12. Acknowledgements 10
13. References 11
13.1. Normative References 11
13.2. Informative References 12
Appendix A. Change Log 13
Author's Address 14

1. Introduction

IPv6 Neighbor Discovery (ND) is the process by which nodes on the link discover each other's presence as well as advertise and receive configuration information. IPv6 Prefix Delegation (PD) entails 1) the communication of a prefix from a delegation service to a requesting node, 2) a representation of the prefix in the network's Routing Information Base (RIB) and the first-hop router's Forwarding Information Base (FIB), and 3) a control messaging service to maintain prefix lifetimes. Following delegation, the prefix is available for the requesting node's exclusive use and is not shared with any other nodes. This document considers prefix delegation models and multiaddressing considerations for requesting nodes that act as a router on behalf of any downstream networks and/or as a host on behalf of their local applications.

For nodes that connect downstream-attached networks (e.g., a cellphone that connects a "tethered" Internet of Things (IoT), a laptop computer with a complex internal network of virtual machines, etc.), the classic routing model applies as shown in Figure 1:

```
                               .---------.
                            ,-(           )-.
                          (   +----------+   )
                         (    |Server 'S'|    )
                          (   +----------+   )
                           (   Network 'N'  )
                            `-(__________)-'
                                   |
                        +----------+----------+
                        | first-hop router 'F'|
                        +----------+----------+
                                   |
                     upstream link |
                                   |
                        +----------+----------+
                        |  upstream interface |
                        +---------------------+
                        |                     |
                        | requesting node 'R' |
                        |    (Prefix 'P')     |
                        |                     |
                        +--+-+--+-+--+-----+--+
                        |A1| |A2| |A3| ... |Aj|
                        +--+-+--+-+--+-----+--+
                        |downstream interfaces|
                        +----------+----------+
                                   |
          internal and/or external |
              downstream links     |
       X----+-------------+--------+----+---------------+---X
            |             |             |               |
       +---++-+--+   +---++-+--+   +---++-+--+     +---++-+--+
       |   |Ak|  |   |   |Al|  |   |   |Am|  |     |   |A*|  |
       |   +--+  |   |   +--+  |   |   +--+  |     |   +--+  |
       | host H1 |   | host H2 |   | host H3 | ... | host Hn |
       +---------+   +---------+   +---------+     +---------+

          <-------------- Downstream Network ------------->
```

Figure 1: Classic Routing Model

In the classic routing model, requesting node 'R' has one or more upstream interfaces and connects zero or more internal and/or external downstream networks. When 'R' requests a prefix delegation, the following sequence of events transpires:

- o Server 'S' located in network 'N' delegates prefix 'P' to requesting node 'R'.

- 'P' is injected into the RIB for 'N', and first hop router 'F' configures a FIB entry with 'R' as the next hop.

- R' receives 'P' and assigns zero or more addresses 'A(*)' taken from 'P' to its downstream interfaces

- 'R' advertises zero or more sub-prefixes taken from 'P' to hosts 'H(i)' on downstream networks.

- 'R' delegates zero or more sub-prefixes taken from 'P' to requesting nodes in downstream networks.

- 'R' acts as a router for hosts 'H(i)' on downstream networks and as a host on behalf of its local applications.

This document also considers the case when 'R' uses portions of 'P' for its own internal multi-addressing purposes. [RFC7934] provides Best Current Practice (BCP) motivations for the benefits of multi-addressing, while an operational means for providing nodes with multiple addresses is given in [RFC8273]. The following multi-addressing alternatives for delegated prefixes compliment this framework.

In a first alternative, when requesting node 'R' receives prefix 'P', it can assign addresses taken from 'P' to downstream virtual interfaces (e.g., a loopback) as shown in Figure 2:

```
                               x
                               |
                 upstream link |
                               |
                    +----------+----------+
                    |  upstream interface |
                    +---------------------+
                    |                     |
                    | requesting node 'R' |
                    |                     |
                    +--+-+--+-+--+-----+--+
                    |A1| |A2| |A3| ... |An|
                    +--+-+--+-+--+-----+--+
                    |  virtual interfaces |
                    +---------------------+
```

Figure 2: Address Assignment to Downstream Virtual Interfaces

In a second alternative, 'R' could assign IPv6 addresses taken from 'P' to the upstream interface over which the prefix was received as shown in Figure 3:

```
                                   x
                                   |
                     upstream link |
                                   |
                        +----------+----------+
                        |  upstream interface |
                        +--+-+--+-+--+-----+--+
                        |A1| |A2| |A3| ... |An|
                        +--+-+--+-+--+-----+--+
                        |                     |
                        | requesting node 'R' |
                        |                     |
                        +---------------------+
```

Figure 3: Upstream Interface Address Assignment

In a third alternative, 'R' could assign IPv6 addresses taken from 'P' to its local applications which appear as "psuedo" virtual interfaces as shown in Figure 4:

```
                                   x
                                   |
                     upstream link |
                                   |
                        +----------+----------+
                        |  upstream interface |
                        +---------------------+
                        |                     |
                        | requesting node 'R' |
                        |                     |
                        +--+-+--+-+--+-----+--+
                        |A1| |A2| |A3| ... |An|
                        +--+-+--+-+--+-----+--+
                        |     Applications    |
                        +---------------------+
```

Figure 4: Application Addressing Model

With these IPv6 PD-based multi-addressing considerations, the node can configure an unlimited supply of addresses to make them available for local applications without requiring coordination with other nodes on upstream interfaces. The following sections present considerations for nodes that employ IPv6 PD mechanisms.

2. Terminology

The terms "node", "host" and "router" are the same as defined in [RFC8200]. The terms Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS), Neighbor Advertisement (NA), Redirect and Prefix Information Option (PIO) are the same as defined in [RFC4861]. All other terminology in the normative references applies, while the following terms are defined within the context of this document:

shared prefix

an IPv6 prefix that may be advertised to more than one node on the link. The router that advertises the prefix must consider the prefix as on-link so that the IPv6 ND address resolution function will identify the correct neighbor for each packet.

individual prefix

an IPv6 prefix that is advertised to exactly one node on the link, where the node may be unaware that the prefix is individual and may not participate in prefix maintenance procedures. The router that advertises the prefix can consider the prefix as on-link or not on-link. In the former case, the router performs address resolution and only forwards those packets that match one of the node's configured addresses so that the node will not receive unwanted packets. In the latter case, the router can simply forward all packets matching the prefix to the node which must then drop any packets that do not match one of its configured addresses. An example individual prefix service is documented in [RFC8273].

delegated prefix

an IPv6 prefix that is explicitly conveyed to a node for its own exclusive use, where the node is an active participant in prefix delegation and maintenance procedures. The first-hop router simply forwards all packets matching the prefix to the requesting node. The requesting node associates the prefix with downstream and/or internal virtual interfaces (i.e., and not the upstream interface).

3. Multi-Addressing Considerations

IPv6 allows nodes to assign multiple addresses to a single interface. [RFC7934] discusses options for multi-addressing as well as use cases where multi-addressing may be desirable. Address configuration options for multi-addressing include StateLess Address AutoConfiguration (SLAAC) [RFC4862], Dynamic Host Configuration Protocol for IPv6 (DHCPv6) address configuration [RFC8415], manual configuration, etc.

Nodes configure addresses from a shared or individual prefix and assign them to the upstream interface over which the prefix was received. When the node assigns the addresses, it is required to use Multicast Listener Discovery (MLD) [RFC3810] to join the appropriate solicited-node multicast group(s) and to use the Duplicate Address Detection (DAD) algorithm [RFC4862] to ensure that no other node configures a duplicate address.

In contrast, a node that configures addresses from a delegated prefix can assign them without invoking MLD/DAD on an upstream interface, since the prefix has been delegated to the node for its own exclusive use and is not shared with any other nodes.

4. Multi-Addressing Alternatives for Delegated Prefixes

When a node receives a delegated prefix, it has many alternatives for provisioning the prefix to its local interfaces and/or downstream networks. [RFC7278] discusses alternatives for provisioning a prefix obtained by a User Equipment (UE) device under the 3rd Generation Partnership Program (3GPP) service model. This document considers the more general case when the node receives a delegated prefix explicitly provided for its own exclusive use.

When the node receives the prefix, it can distribute the prefix to internal (virtual) or external (physical) downstream networks and optionally configure addresses for itself on downstream interfaces. The node then acts as a router on behalf of its downstream networks.

The node could instead (or in addition) use portions of the delegated prefix for its own multi-addressing purposes. In a first alternative, the node can assign as many addresses as it wants from the prefix to downstream virtual interfaces.

In a second alternative, the node can assign as many addresses as it wants from the prefix to the upstream interface over which the prefix was received, but in normal practice does not assign the prefix itself (or subnets from the prefix) to the upstream interface. If the node assigned the prefix to the upstream interface, any neighbors on the upstream link receiving an RA could configure addresses from the prefix and a default route with the node as the next hop. This could create a loop where upstream link neighbors send packets to the node which in turn forwards them to another upstream link neighbor. Still, there may be cases where the node provides services for dependent neighbors on the upstream link that have no other means of connecting to the network. ([RFC8415] chose to remain silent on this subject since it is operational rather than functional in nature.)

In a third alternative, the node can assign addresses taken from the delegated prefix to its local applications. The applications themselves then serve as virtual interfaces. (Note that, in the future, the practice of assigning unique non-link-local IPv6 addresses to applications could obviate the need for transport protocol port numbers.)

In these multi-addressing cases, the node normally assigns the prefix itself to a virtual interface such as a loopback so that unused portions of the prefix are correctly identified as unreachable. The node then acts as a host on behalf of its local applications even though neighbors on the upstream link consider it as a router.

5. Address Autoconfiguration Considerations

Nodes autoconfigure addresses according to Section 6 of IPv6 Node Requirements [RFC8504].

Nodes that connect to a network that spans more than just a single LAN configure at least one non-link-local address, i.e., for network management and error reporting purposes.

Nodes recognize the Subnet Router Anycast address [RFC4291] for each delegated prefix. Therefore, the node's use of the Subnet Router Anycast address must be indistinguishable from the behavior of an ordinary router when viewed from the outside world.

6. MLD/DAD Implications

When a node configures addresses for itself from a shared or individual prefix (and when the interface variable 'DupAddrDetectTransmits' is non-zero [RFC4862]), the node performs MLD/DAD by sending multicast messages over the upstream interface to test whether there is another node on the link that configures a duplicate address. When there are many such addresses and/or many such nodes, this could result in substantial multicast traffic that affects all nodes on the link.

When a node configures addresses for itself from a delegated prefix and assigns them on downstream interfaces, it can configure as many addresses as it wants without performing MLD/DAD for any of the addresses over the upstream interface.

When a node configures addresses for itself from a delegated prefix and assigns them on the upstream interface over which the prefix was received, the node honors MLD/DAD procedures according to the interface's 'DupAddrDetectTransmits' variable.

7. Dynamic Routing Protocol Implications

Nodes that receive delegated prefixes can be configured to either participate or not participate in a dynamic routing protocol over the upstream interface. When there are many nodes on the upstream link, dynamic routing protocol participation might be impractical due to scaling limitations, and may also be exacerbated by factors such as node mobility.

Unless it participates in a dynamic routing protocol, the node initially has only a default route pointing to a neighbor via an upstream interface. This means that packets sent by the node over an upstream interface will initially go through a default router even if there is a better first-hop node on the link. The node may subsequently receive Redirect messages from the default router that identify a better first-hop.

8. IPv6 Neighbor Discovery Implications

According to [RFC4861], when a node receives a shared or individual prefix with "L=1" and has a packet to send to an IPv6 destination within the prefix, it is required to use the IPv6 ND address resolution function to resolve the link-layer address of a neighbor on the link that configures the address.

Also according to [RFC4861], when a node receives a shared or individual prefix with "L=0" and has a packet to send to an IPv6 destination within the prefix, it sends the packet to a default router since "L=0" makes no statement about on-link or off-link properties of the prefix.

When a node requires a delegated prefix, it acts as a simple host by sending RS messages over the upstream interface in the manner described in Section 4.2 of [RFC7084] and invokes prefix delegation services as discussed in Section 9. The node considers the upstream interface as a non-advertising interface [RFC4861], i.e., it does not send RA messages over the upstream interface. The node further does not perform the IPv6 ND address resolution function over the upstream interface, since the delegated prefix is by definition not associated with the upstream interface.

9. Prefix Delegation Services

Selection of prefix delegation services must be considered according to specific use cases. An example service is that offered by standard DHCPv6 Prefix Delegation [RFC8415]. Alternative services based on IPv6 ND messaging have also been proposed [I-D.templin-6man-dhcpv6-ndopt][I-D.naveen-slaac-prefix-management].

Other, non-router, mechanisms may exist, such as proprietary IPAMs, [I-D.ietf-anima-prefix-management] and [I-D.li-opsawg-address-pool-management-arch]. Requirements for extending an IPv6 /64 Prefix from a Third Generation Partnership Project (3GPP) Mobile Interface to a LAN Link are discussed in [RFC7278].

10. IANA Considerations

This document introduces no IANA considerations.

11. Security Considerations

Security considerations for IPv6 Neighbor Discovery [RFC4861] and any applicable PD mechanisms apply to this document. Nodes that manage their delegated prefixes such that MLD/DAD procedures are not needed on the upstream interface can avoid introducing multicast messaging congestion on the upstream link. Also, routers that delegate prefixes keep only a single neighbor cache entry for each prefix delegation recipient, meaning that the router's neighbor cache cannot be subject to address resolution-based resource exhaustion attacks.

For shared and individual prefixes, if the advertising router considers the prefix as on-link the IPv6 ND address resolution function will prevent unwanted IPv6 packets from reaching the node. For delegated prefixes and individual prefixes that are not considered on-link, the router delivers all packets that match the prefix to the node. In that case, the node may receive unwanted IPv6 packets via an upstream interface for which it has no matching configured address. The node then drops the packets and observes the ICMPv6 "Destination Unreachable - Address/Port unreachable" procedures discussed in [RFC4443].

The node may also receive IPv6 packets via an upstream interface that do not match any of the node's delegated prefixes. In that case, the node drops the packets and observes the ICMPv6 "Destination Unreachable - No route to destination" procedures discussed in [RFC4443]. Dropping the packets is necessary to avoid a reflection attack that would cause the node to forward packets received from an upstream interface via the same or a different upstream interface.

12. Acknowledgements

This work was motivated by discussions on the v6ops list. Mark Smith, Ricardo Pelaez-Negro, Edwin Cordeiro, Fred Baker, Ron Bonica, Naveen Lakshman, Ole Troan, Bob Hinden, Brian Carpenter, Joel Halpern, Albert Manfredi, Dusan Mudric, Paul Marks, Joe Touch, Alex

Petrescu, Lorenzo Colitti, Tatuya Jinmei and Naveen Kottapalli provided useful comments that have greatly improved the document.

This work is aligned with the NASA Safe Autonomous Systems Operation (SASO) program under NASA contract number NNA16BD84C.

This work is aligned with the FAA as per the SE2025 contract number DTFAWA-15-D-00030.

This work is aligned with the Boeing Commercial Airplanes (BCA) Internet of Things (IoT) and autonomy programs.

This work is aligned with the Boeing Information Technology (BIT) MobileNet program.

13. References

13.1. Normative References

[RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <https://www.rfc-editor.org/info/rfc3810>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[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>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

13.2. Informative References

[I-D.ietf-anima-prefix-management]
Jiang, S., Du, Z., Carpenter, B., and Q. Sun, "Autonomic IPv6 Edge Prefix Management in Large-scale Networks", draft-ietf-anima-prefix-management-07 (work in progress), December 2017.

[I-D.li-opsawg-address-pool-management-arch]
Li, C., Xie, C., Kumar, R., Fioccola, G., Xu, W., LIU, W., Ma, D., and J. Bi, "Coordinated Address Space Management architecture", draft-li-opsawg-address-pool-management-arch-01 (work in progress), July 2018.

[I-D.naveen-slaac-prefix-management]
Kottapalli, N., "IPv6 Stateless Prefix Management", draft-naveen-slaac-prefix-management-00 (work in progress), November 2018.

[I-D.templin-6man-dhcpv6-ndopt]
Templin, F., "A Unified Stateful/Stateless Configuration Service for IPv6", draft-templin-6man-dhcpv6-ndopt-10 (work in progress), June 2020.

[I-D.templin-6man-rio-redirect]
Templin, F. and j. woodyatt, "Route Information Options in IPv6 Neighbor Discovery", draft-templin-6man-rio-redirect-08 (work in progress), June 2019.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <https://www.rfc-editor.org/info/rfc7084>.

[RFC7278] Byrne, C., Drown, D., and A. Vizdal, "Extending an IPv6 /64 Prefix from a Third Generation Partnership Project (3GPP) Mobile Interface to a LAN Link", RFC 7278, DOI 10.17487/RFC7278, June 2014, <https://www.rfc-editor.org/info/rfc7278>.

[RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", BCP 204, RFC 7934, DOI 10.17487/RFC7934, July 2016, <https://www.rfc-editor.org/info/rfc7934>.

[RFC8273] Brzozowski, J. and G. Van de Velde, "Unique IPv6 Prefix per Host", RFC 8273, DOI 10.17487/RFC8273, December 2017, <https://www.rfc-editor.org/info/rfc8273>.

[RFC8504] Chown, T., Loughney, J., and T. Winters, "IPv6 Node Requirements", BCP 220, RFC 8504, DOI 10.17487/RFC8504, January 2019, <https://www.rfc-editor.org/info/rfc8504>.

Appendix A. Change Log

<< RFC Editor - remove prior to publication >>

Changes from -25 to -26:

o Version and reference update

Changes from -24 to -25:

o Version and reference update

Changes from -23 to -24:

o Version and reference update

Changes from -22 to -23:

o Changed DHCPv6 references to RFC8415. Deprecate RFC3315 and RFC3633.

o New text on assignment of addresses and prefixes on the upstream interface.

Changes from -21 to -22:

o Changes to address list comments contributed by Lorenzo Colitti, Tatuya Jinmei, Brian Carpenter and Fred Baker.

- Deleted section on ICMPv6 - now defer to normative reference [RFC4443].

- Discuss 'DupAddrDetectTransmits' variable implications under MLD/ DAD considerations.

Changes from -20 to -21:

- Re-worked classic routing model section

- Included multi-addressing case where addresses may be assigned to applications

- Removed strong/weak end system discussions

Changes from -19 to -20:

- figure 1 updates to show Server as being somewhere in the network

- Introductory material to show relation to other RFCs on multi-addressing

Changes from -18 to -19:

- added new section on Prefix Delegation Services

Changes from -17 to -18:

- re-worked discussion on the prefix delegation service in Section 1

- updated figures in Section 1

Changes from -16 to -17:

- added supporting text in the introduction to discuss the Delegating Router's relationship with the Requesting Router and with supporting intrastructure in the operator's network

- updated figures in introduction to include representation of operator's network

- added new section on Address Autoconfiguration Considerations

Author's Address

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org