Internet Engineering Task Force (IETF)
Request for Comments: 9663
Category: Informational
ISSN: 2070-1721

L. Colitti
Google, LLC
J. Linkova, Ed.
X. Ma, Ed.
Google

October 2024

Using DHCPv6 Prefix Delegation (DHCPv6-PD) to Allocate Unique IPv6 Prefixes per Client in Large Broadcast Networks

Abstract

This document discusses an IPv6 deployment scenario when individual nodes connected to large broadcast networks (such as enterprise networks or public Wi-Fi networks) are allocated unique prefixes via DHCPv6 Prefix Delegation (DHCPv6-PD), as specified in RFC 8415.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

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). Not all documents approved by the IESG are candidates for any level of Internet Standard; see 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/rfc9663.

Table of Contents

1. Introduction

Often, broadcast networks such as enterprise or public Wi-Fi deployments place many devices on a shared link with a single on-link prefix. This document describes an alternative deployment model where individual devices obtain prefixes from the network. This provides two important advantages.

First, it offers better scalability. Unlike IPv4, IPv6 allows hosts to have multiple addresses, and this is the case in most deployments (see Appendix A for more details). However, increasing the number of addresses introduces scalability issues on the network infrastructure. Network devices need to maintain various types of tables and hashes (Neighbor Cache on first-hop routers, Neighbor Discovery Proxy caches on Layer 2 devices, etc.). On Virtual eXtensible Local Area Network (VXLAN) networks [RFC7348], each address might be represented as a route. This means, for example, that if every client has 10 addresses instead of one, the network must support 10 times more routes, etc. If an infrastructure device's resources are exhausted, the device might drop some IPv6 addresses from the corresponding tables, while the address owner might still be using the address to send traffic. This leads to traffic being discarded and a degraded customer experience. Providing every host with one prefix allows the network to maintain only one entry per device, while still providing the device the ability to use an arbitrary number of addresses.

Second, this deployment model provides the ability to extend the network. In IPv4, a device that connects to the network can provide connectivity to subtended devices by using NAT. With DHCPv6 Prefix Delegation (DHCPv6-PD) [RFC8415], such a device can similarly extend the network, but unlike IPv4 NAT, it can provide its subtended devices with full end-to-end connectivity.

Another method of deploying unique prefixes per device is documented in [RFC8273]. Similarly, the standard deployment model in cellular IPv6 networks [RFC6459] provides a unique prefix to every device. However, providing a unique prefix per device is very uncommon in enterprise-style networks, where nodes are usually connected to broadcast segments such as VLANs and each link has a single on-link prefix assigned. This document takes a similar approach to [RFC8273], but allocates the prefix using DHCPv6-PD.

This document focuses on the behavior of the network. Host behavior is not defined in this document.

2. 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.

3. Terminology

Node: a device that implements IPv6 [RFC8200]

Host: any node that is not a router [RFC8200]

Client: a node that connects to a network and acquires addresses. The node may wish to obtain addresses for its own use, or it may be a router that wishes to extend the network to its physical or virtual subsystems, or both. It may be either a host or a router as defined by [RFC8200].

AP: (wireless) Access Point

DHCPv6 IA_NA: Identity Association for Non-temporary Addresses (Section 21.4 of [RFC8415])

DHCPv6 IA_PD: Identity Association for Prefix Delegation (Section 21.21 of [RFC8415])

DHCPv6-PD: DHCPv6 Prefix Delegation [RFC8415]; a mechanism to delegate IPv6 prefixes to clients.

ND: Neighbor Discovery [RFC4861]

NUD: Neighbor Unreachability Detection [RFC4861]

PIO: Prefix Information Option [RFC4862]

SLAAC: IPv6 Stateless Address Autoconfiguration [RFC4862]

4. Design Principles

Instead of all clients on a given link forming addresses from the same shared prefix assigned to that link, this deployment model operates as described below:

* A device acts as a DHCPv6-PD client and requests a prefix via DHCPv6-PD by sending an IA_PD request.

* The server delegates a prefix to the client and the delegated prefix is installed into the routing table of the first-hop router as a route pointing to the client’s link-local address. The first-hop router can act as a DHCPv6 relay and snoop DHCPv6 Reply messages from an off-link DHCPv6 server, or it can act as a DHCPv6 server itself. In both cases, it can install the route locally, and if the network is running a dynamic routing protocol, distribute the route or the entire prefix pool into the protocol.

* For the router and all other infrastructure devices, the delegated prefix is considered off-link, so traffic to that prefix does not trigger any ND packets, other than the minimum ND required to sustain Neighbor Unreachability Detection (NUD) for the client’s link-local address.

* The device can use the delegated prefix in various ways. For example, it can form addresses, as described in requirement WAA-7 of [RFC7084]. It can also extend the network, as described in [RFC7084] or [RFC7278].

An example scenario is shown in Figure 1. Note that the prefix lengths used in the example are /64 because that is the prefix length

currently supported by SLAAC and is not otherwise required by the proposed deployment model.

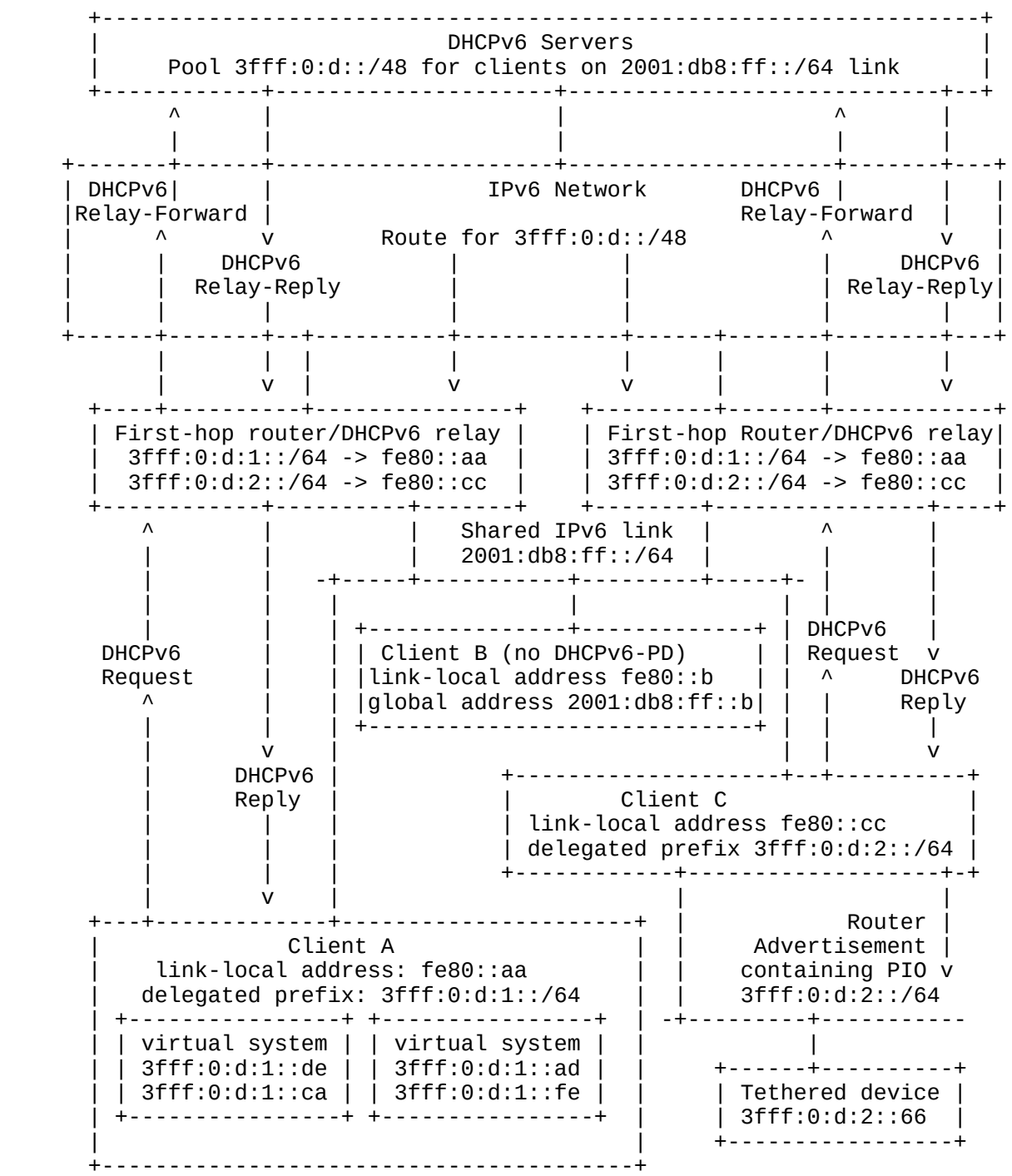

Figure 1: An Example Topology with Two First-Hop Routers

5. Applicability and Limitations

Delegating a unique prefix per client provides all the benefits of both SLAAC and DHCPv6 address allocation, but at the cost of greater address-space usage. This design would substantially benefit some networks (see Section 12) in which the additional cost of an additional service (such as DHCPv6 Prefix Delegation) and allocation of a larger amount of address space can easily be justified. Examples of such networks include but are not limited to:

* Large-scale networks where even three to five addresses per client might introduce scalability issues.

* Networks with a high number of virtual hosts, so physical devices require multiple addresses.

* Managed networks where extensive troubleshooting, device traffic logging, or forensics might be required.

In smaller networks, such as home networks or small enterprises with smaller address space and a lower number of clients, SLAAC is a simpler and often preferred option.

6. Routing and Addressing Considerations

6.1. Prefix Pool Allocation

One simple deployment model is to assign a dedicated prefix pool to each link. The prefixes from each link's pool are only issued to requesting clients on the link; if clients move to another link, they will obtain a prefix from the pool associated with the new link (see Section 9).

This is very similar to how address pools are allocated when using DHCP to assign individual addresses (e.g., DHCPv4 or DHCPv6 IA_NA), where each link has a dedicated pool of addresses, and clients on the link obtain addresses from the pool. In this model, the network can route the entire pool to the link's first-hop routers, and the routers do not need to advertise individual delegated prefixes into the network's dynamic routing protocol.

Other deployment models, such as prefix pools shared over multiple links or routers, are possible but are not described in this document.

6.2. First-Hop Router Requirements

In large networks, DHCPv6 servers are usually centralized and reached via DHCPv6 relays co-located with the first-hop routers. To delegate IPv6 prefixes to clients, the first hop routers need to implement DHCPv6 relay functions and meet the requirements defined in [RFC8987]. In particular, per Section 4.2 of [RFC8987], the first-hop router must maintain a local routing table that contains all prefixes delegated to clients.

With the first-hop routers performing DHCPv6 relay functions, the proposed design neither requires any subsequent relays in the path nor introduces any requirements (e.g., snooping) for such subsequent relays, if they are deployed.

To ensure that routes to the delegated prefixes are preserved even if a relay is rebooted or replaced, the operator MUST ensure that all relays in the network infrastructure support DHCPv6 Bulk Leasequery as defined in [RFC5460]. While Section 4.3 of [RFC8987] lists keeping active prefix delegations in persistent storage as an alternative to DHCPv6 Bulk Leasequery, relying on persistent storage has the following drawbacks:

* In a network with multiple relays, network state can change significantly while the relay is rebooting (new prefixes might be delegated or some prefixes might be expiring, etc).

* Persistent storage might not be preserved if the router is physically replaced.

Another mechanism for first-hop routers to obtain information about delegated prefixes is by using Active Leasequery [RFC7653], though this is not yet widely supported.

6.3. Topologies with Multiple First-Hop Routers

In a topology with redundant first-hop routers, all the routers need to relay DHCPv6 traffic, install the delegated prefixes into their routing tables and, if needed, advertise those prefixes to the network.

If the first-hop routers obtain information about delegated prefixes by snooping DHCPv6 Reply messages sent by the server, then all the first-hop routers must be able to snoop these messages. This is possible if the client multicasts the DHCPv6 messages it sends to the server. The server will receive one copy of the client message through each first-hop relay, and will reply unicast to each of them via the relay (or chain of relays) from which it received the message. Thus, all first-hop relays will be able to snoop the replies. Per Section 14 of [RFC8415], clients always use multicast unless the server uses the Server Unicast option to explicitly allow unicast communication ([RFC8415], Section 21.12). Therefore, in topologies with multiple first-hop routers, the DHCPv6 servers MUST be configured not to use the Server Unicast option. It should be noted that [RFC8415bis] deprecates the Server Unicast option precisely because it is not compatible with topologies with multiple first-hop relays.

To recover from crashes or reboots, relays can use Bulk Leasequery or Active Leasequery to issue a QUERY_BY_RELAY_ID with the ID(s) of the other relay(s), as configured by the operator. Additionally, some vendors provide vendor-specific mechanisms to synchronize state between DHCP relays.

6.4. On-Link Communication

For security reasons, some networks block on-link device-to-device traffic at Layer 2 to prevent communication between clients on the same link. In this case, delegating a prefix to each client doesn’t affect traffic flows, as all traffic is sent to the first-hop router anyway. Depending on the network security policy, the router may allow or drop the traffic.

If the network does allow peer-to-peer communication, the PIO for the on-link prefix is usually advertised with the L-bit set to 1 [RFC4861]. As a result, all addresses from that prefix are considered on-link, and traffic to those destinations is sent directly (not via routers). If such a network delegates prefixes to clients (as described in this document), then each client will consider other client’s destination addresses to be off-link, because those addresses are from the delegated prefixes and are no longer within the on-link prefix. When a client sends traffic to another client, packets will initially be sent to the default router. The router will respond with an ICMPv6 redirect message (Section 4.5 of [RFC4861]). If the client receives and accepts the redirect, then traffic can flow directly from device to device. Therefore, the administrator deploying the solution described in this document SHOULD ensure that the first-hop routers can send ICMPv6 redirects (the routers are configured to do so and the security policies permit those messages).

7. DHCPv6-PD Server Considerations

This document does not introduce any changes to the DHCPv6 protocol itself. However, for the proposed solution to work correctly, the DHCPv6-PD server needs to be configured as follows:

* The server MUST follow recommendations from [RFC8168] on processing prefix-length hints.

* The server MUST provide a prefix short enough for the client to extend the network to at least one interface and allow nodes on that interface to obtain addresses via SLAAC. The server MAY provide more address space to clients that ask for it, either by delegating multiple such prefixes, or by delegating a single prefix of a shorter length. It should be noted that [RFC8168] allows the server to provide a prefix shorter than the prefix-length hint value received from the client.

* If the server receives the same Solicit message from the same client multiple times through multiple relays, it MUST reply to all of them with the same prefix(es). This ensures that all the relays will correctly configure routes to the delegated prefixes.

* The server MUST NOT send the Server Unicast option to the client unless the network topology guarantees that no client is connected to a link with multiple relays (see Section 6.3).

* In order to ensure uninterrupted connectivity when a first-hop router crashes or reboots, the server MUST support Bulk Leasequery or Active Leasequery.

As most operators have some experience with IPv4, they can use a similar approach for choosing the pool size and the timers (such as T1 and T2 timers). In particular, the following factors should be taken into account:

* the expected maximum number of clients;

* the average duration of client connections;

* how mobile the clients are (a network where all clients are connected to a single wired VLAN might choose longer timers than a network where clients can switch between multiple wireless networks);

* how often clients are expected to reconnect to the network (for example, a corporate authenticated Wi-Fi network might be using longer timers than an open public Wi-Fi).

DHCPv6 servers that delegate prefixes can interface with Dynamic DNS infrastructure to automatically populate reverse DNS using wildcard records, similarly to what is described in Section 2.2 of [RFC8501]. Networks that also wish to populate forward DNS cannot do so automatically based only on DHCPv6 prefix delegation transactions, but they can do so in other ways, such as by supporting DHCPv6 address registration as described in [ADDR-NOTIFICATION].

Some additional recommendations driven by security and privacy considerations are discussed in Section 15 and Section 13.

8. Prefix Length Considerations

Delegating a prefix of sufficient size to use SLAAC allows the client to extend the network, providing limitless addresses to IPv6 nodes connected to it (e.g., virtual machines or tethered devices), because all IPv6 hosts are required to support SLAAC [RFC8504]. Additionally, even clients that support other forms of address assignment require SLAAC for some functions, such as forming dedicated addresses for the use of 464XLAT (see Section 6.3 of [RFC6877]).

At the time of writing, the only prefix size that will allow devices to use SLAAC is 64 bits. Also, as noted in [RFC7421], using an interface identifier (IID) shorter than 64 bits and a subnet prefix

longer than 64 bits is outside the current IPv6 specifications. Choosing longer prefixes would require the client and any connected system to use other address assignment mechanisms. This would limit the applicability of the proposed solution, as other mechanisms are not currently supported by many hosts.

For the same reasons, a prefix length of /64 or shorter is required to extend the network as described in [RFC7084] (see requirement L-2), and a prefix length of /64 is required to provide global connectivity for stub networks as per [SNAC-SIMPLE].

Assigning a prefix of sufficient size to support SLAAC is possible on large networks. In general, any network that numbers clients from an IPv4 prefix of length X (e.g., X=/18, X=/24) would require an IPv6 prefix of length X+32 (e.g., X=/40, X=/56) to provide a /64 prefix to every device. As an example, Section 9.2 of [RFC7934] suggests that even a very large network that assigns every single one of the 16 million IPv4 addresses in 10.0.0.0/8 would only need an IPv6 /40. A /40 prefix is a small amount of address space: there are 32 times more /40s in the current IPv6 unicast range 2000::/3 than there are IPv4 addresses. Existing sites that currently use a /48 prefix cannot support more than 64k clients in this model without renumbering, though many networks of such size have Local Internet Registry (LIR) status and can justify bigger address blocks.

Note that assigning a prefix of sufficient size to support SLAAC does not require that subtended nodes use SLAAC; they can use other address assignment mechanisms as well.

9. Client Mobility

As per Section 18.2.12 of [RFC8415], when the client moves to a new link, it MUST initiate a Rebind/Reply message exchange. Therefore, when the client moves between network attachment points, it would refresh its delegated prefix the same way it refreshes addresses assigned (via SLAAC or DHCPv6 IA_NA) from a shared on-link prefix:

* When a client moves from between different attachment points on the same link (e.g., roams between two APs while connected to the same wireless network or moves between two switchports belonging to the same VLAN), the delegated prefix does not change, and the first-hop routers have a route for the prefix with the nexthop set to the client link-local address on that link. As per requirement S-2 in Section 4.3 of [RFC8987], the DHCPv6-relays (the first-hop routers) MUST retain the route for the delegating prefix until the route is released or removed due to expiring DHCP timers. Therefore, if the client reconnects to the same link, the prefix doesn’t change.

* When a client moves to a different link, the DHCPv6 server provides the client with a new prefix, so the behavior is consistent with SLAAC or DHCPv6-assigned addresses, which are also different on the new link.

In theory, DHCPv6 servers can delegate the same prefix to the same client even if the client changes the attachment points. However, while allowing the client to keep the same prefix while roaming between links might provide some benefits for the client, it is not feasible without changing DHCPv6 relay behavior: after the client moves to a new link, the DHCPv6 relays would retain the route pointing to the client’s link-local address on the old link for the duration of DHCPv6 timers (see requirement S-2, Section 4.3 of [RFC8987]). As a result, the first-hop routers would have two routes for the same prefix pointing to different links, causing connectivity issues for the client.

It should be noted that addressing clients from a shared on-link prefix also does not allow clients to keep addresses while roaming between links, so the proposed solution is not different in that regard. In addition to that, different links often have different security policies applied (for example, corporate internal networks versus guest networks), hence clients on different links need to use different prefixes.

10. Antispoofing and SAVI Interaction

Enabling unicast Reverse Path Forwarding (uRPF) [RFC3704] on the first-hop router interfaces towards clients provides the first layer of defense against spoofing. A spoofed packet sent by a malicious client would be dropped by the router unless the spoofed address belongs to a prefix delegated to another client on the same interface. Therefore the malicious client can only spoof addresses already delegated to another client on the same link or another client's link-local address.

Source Address Validation Improvement (SAVI) [RFC7039] provides more reliable protection against address spoofing. Administrators deploying the proposed solution on SAVI-enabled infrastructure SHOULD ensure that SAVI perimeter devices support DHCPv6-PD snooping to create the correct binding for the delegated prefixes (see [RFC7513]). Using FCFS SAVI [RFC6620] to protect link-local addresses and create SAVI bindings for DHCPv6-PD assigned prefixes would prevent spoofing.

Some infrastructure devices do not implement SAVI as defined in [RFC7039]; instead, they perform other forms of address tracking and snooping for security or performance improvement purposes (e.g., ND proxy). This is very common behavior for wireless devices (such as access points and controllers). Administrators SHOULD ensure that such devices are able to snoop DHCPv6-PD packets so the traffic from the delegated prefixes is not dropped.

It should be noted that using DHCPv6-PD makes it harder for an attacker to select the spoofed source address. When all clients are using the same shared link to form addresses, the attacker might learn addresses used by other clients by listening to multicast Neighbor Solicitations and Neighbor Advertisements. In DHCPv6-PD environments, however, the attacker can only learn about other clients' global addresses by listening to multicast DHCPv6 messages, which are not transmitted so often, and may not be received by the client at all because they are sent to multicast groups that are specific to DHCPv6 servers and relays.

11. Migration Strategies and Co-existence with SLAAC Using Prefixes from the PIO

It would be beneficial for the network to explicitly indicate its support of DHCPv6-PD for connected clients.

* In small networks (e.g., home networks), where the number of clients is not too high, the number of available prefixes becomes a limiting factor. If every phone or laptop in a home network were to request a unique prefix suitable for SLAAC, the home network might run out of prefixes, if the prefix allocated to the Customer Premises Equipment (CPE) by its ISP is too long. For example, if an ISP delegates a /60, the CPE would only be able to delegate fifteen /64 prefixes to clients. So while the enterprise network administrator might want all phones in the network to request a prefix, it would be highly undesirable for the same phone to request a prefix when connecting to a home network.

* When the network supports both a unique prefix per client and a

PIO with A=1 as address assignment methods, it's highly desirable for the client NOT to use the PIO prefix to form global addresses and instead only use the prefix delegated via DHCPv6-PD. Starting both SLAAC using the PIO prefix and DHCPv6-PD, and then deprecating the SLAAC addresses after receiving a delegated prefix would be very disruptive for applications. If the client continues to use addresses formed from the PIO prefix, it would not only undermine the benefits of the proposed solution (see Section 12), but it would also introduce complexity and unpredictability in the source address selection. Therefore, the client needs to know what address assignment method to use and whether or not to use the prefix in the PIO, if the network provides the PIO with the 'A' flag set.

The deployment model described in this document does not require the network to signal support of DHCPv6-PD: for example, devices acting as compatible routers [RFC7084] will be able to receive prefixes via DHCPv6-PD even without such signaling. Also, some clients may decide to start DHCPv6-PD and acquire prefixes if they detect that the network does not provide addresses via SLAAC. To fully achieve the benefits described in this section, [PIO-PFLAG] defines a new PIO flag to signal that DHCPv6-PD is the preferred method of obtaining prefixes.

12. Benefits

The proposed solution provides the following benefits:

* Network device resources (e.g., memory) need to scale to the number of devices, not the number of IPv6 addresses. The first-hop routers have a single route per device pointing to the device's link-local address. This can potentially enable hardware cost savings; for example, if hardware such as wireless LAN controllers is limited to supporting only a specific number of client addresses, or in VXLAN deployments where each client address consumes one routing table entry.

* The cost of having multiple addresses is offloaded to the clients. Hosts are free to create and use as many addresses as they need without imposing any additional costs onto the network.

* If all clients connected to the given link support this mode of operation and can generate addresses from the delegated prefixes, there is no reason to advertise a common prefix assigned to that link in the PIO with the 'A' flag set. Therefore, it is possible to remove the global shared prefix from that link and the router interface completely, so no global addresses are on-link for the link. This would lead to reducing the attack surface for Neighbor Discovery attacks described in [RFC6583].

* DHCPv6-PD logs and routing tables obtained from first-hop routers provide complete information on IPv6 to MAC mapping, which can be used for forensics and troubleshooting. Such information is much less dynamic than the ND cache; therefore, it's much easier for an operator to collect and process it.

* A dedicated prefix per client allows the network administrator to create security policies per device (such as ACLs) even if the client is using temporary addresses. This mitigates one of the issues described in [IPv6-ADDRESS].

* Fate sharing: all global addresses used by a given client are routed as a single prefix. Either all of them work or none of them work, which makes failures easier to diagnose and mitigate.

* Lower level of multicast traffic: less Neighbor Discovery

[RFC4861] multicast packets, as the routers need to resolve only the clients' link-local addresses. Also, there is no Duplicate Address Detection (DAD) traffic except for the clients' link-local addresses.

* Ability to extend the network transparently. If the network delegates to the client a prefix of sufficient size to support SLAAC, the client can provide connectivity to other hosts, as is possible in IPv4 with NAT (e.g., by acting as an IPv6 Customer Edge (CE) router as described in [RFC7084]).

13. Privacy Considerations

If an eavesdropper or information collector is aware that a given client is using the proposed mechanism, then they may be able to track the client based on its prefix. The privacy implications of this are equivalent to the privacy implications of networks using stateful DHCPv6 address assignment: in both cases, the IPv6 addresses are determined by the server, either because the server assigns a full 128-bit address in a shared prefix, or because the server determines what prefix is delegated to the client. Administrators deploying the proposed mechanism can use similar methods to mitigate the impact as the ones used today in networks that use stateful DHCPv6 address assignment.

Except for networks (such as datacenter networks) where hosts do not need temporary addresses [RFC8981], the network SHOULD:

* Ensure that when a client requests a prefix, the prefix is randomly assigned and not allocated deterministically.

* Use short prefix lifetimes (e.g., hours) to ensure that when a client disconnects and reconnects it gets a different prefix.

* Allow the client to have more than one prefix at the same time. This allows the client to rotate prefixes using a mechanism similar to temporary addresses, but that operates on prefixes instead of on individual addresses. In this case, the prefix's lifetime MUST be short enough to allow the client to use a reasonable rotation interval without using too much address space. For example, if every 24 hours the client asks for a new prefix and stops renewing the old prefix, and the Valid Lifetime of delegated prefixes is one hour, then the client will consume two prefixes for one hour out of 24 hours, and thus will consume just under 1.05 prefixes on average.

14. IANA Considerations

This document has no IANA actions.

15. Security Considerations

A malicious (or just misbehaving) client might attempt to exhaust the DHCPv6-PD pool by sending a large number of requests with differing DHCP Unique Identifiers (DUIDs). To prevent a misbehaving client from denying service to other clients, the DHCPv6 server or relay MUST support limiting the number of prefixes delegated to a given client at any given time.

Networks can protect against malicious clients by authenticating devices using tokens that cannot be spoofed (e.g., 802.1x authentication) and limiting the number of link-local addresses or MAC addresses that each client is allowed to use. Note that this is not a new issue, as the same attack might be implemented using DHCPv4 or DHCPv6 IA_NA requests; in particular, while it is unlikely for clients to be able to exhaust an IA_NA address pool, clients using

IA_NA can exhaust other resources such as DHCPv6 and routing infrastructure resources such as server RAM, ND cache entries, Ternary Content-Addressable Memory (TCAM) entries, SAVI entries, etc.

A malicious client might request a prefix and then release it very quickly, causing routing convergence events on the relays. The impact of this attack can be reduced if the network rate-limits the amount of broadcast and multicast messages from the client.

Delegating the same prefix for the same client introduces privacy concerns. The proposed mitigation is discussed in Section 13.

Spoofing scenarios and prevention mechanisms are discussed in Section 10.

16. References

16.1. 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>.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <https://www.rfc-editor.org/info/rfc4193>.

[RFC5460] Stapp, M., "DHCPv6 Bulk Leasequery", RFC 5460, DOI 10.17487/RFC5460, February 2009, <https://www.rfc-editor.org/info/rfc5460>.

[RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses", RFC 6620, DOI 10.17487/RFC6620, May 2012, <https://www.rfc-editor.org/info/rfc6620>.

[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>.

[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>.

[RFC8168] Li, T., Liu, C., and Y. Cui, "DHCPv6 Prefix-Length Hint Issues", RFC 8168, DOI 10.17487/RFC8168, May 2017, <https://www.rfc-editor.org/info/rfc8168>.

[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>.

[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>.

[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>.

[RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", RFC 8981, DOI 10.17487/RFC8981, February 2021, <https://www.rfc-editor.org/info/rfc8981>.

[RFC8987] Farrer, I., Kottapalli, N., Hunek, M., and R. Patterson, "DHCPv6 Prefix Delegating Relay Requirements", RFC 8987, DOI 10.17487/RFC8987, February 2021, <https://www.rfc-editor.org/info/rfc8987>.

16.2. Informative References

[ADDR-NOTIFICATION] Kumari, W., Krishnan, S., Asati, R., Colitti, L., Linkova, J., and S. Jiang, "Registering Self-generated IPv6 Addresses using DHCPv6", Work in Progress, Internet-Draft, draft-ietf-dhc-addr-notification-13, 16 May 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-dhc-addr-notification-13>.

[IPv6-ADDRESS] Gont, F. and G. Gont, "Implications of IPv6 Addressing on Security Operations", Work in Progress, Internet-Draft, draft-ietf-opsec-ipv6-addressing-00, 2 June 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-opsec-ipv6-addressing-00>.

[PIO-PFLAG] Colitti, L., Linkova, J., Ma, X., and D. Lamparter, "Signaling DHCPv6 Prefix per Client Availability to Hosts", Work in Progress, Internet-Draft, draft-ietf-6man-pio-pflag-11, 4 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-6man-pio-pflag-11>.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <https://www.rfc-editor.org/info/rfc3704>.

[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>.

[RFC6459] Korhonen, J., Ed., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, DOI 10.17487/RFC6459, January 2012, <https://www.rfc-editor.org/info/rfc6459>.

[RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, DOI 10.17487/RFC6583, March 2012, <https://www.rfc-editor.org/info/rfc6583>.

[RFC7039] Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, Ed., "Source Address Validation Improvement (SAVI) Framework", RFC 7039, DOI 10.17487/RFC7039, October 2013, <https://www.rfc-editor.org/info/rfc7039>.

[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>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7421] Carpenter, B., Ed., Chown, T., Gont, F., Jiang, S., Petrescu, A., and A. Yourtchenko, "Analysis of the 64-bit Boundary in IPv6 Addressing", RFC 7421, DOI 10.17487/RFC7421, January 2015, <https://www.rfc-editor.org/info/rfc7421>.

[RFC7513] Bi, J., Wu, J., Yao, G., and F. Baker, "Source Address Validation Improvement (SAVI) Solution for DHCP", RFC 7513, DOI 10.17487/RFC7513, May 2015, <https://www.rfc-editor.org/info/rfc7513>.

[RFC7653] Raghuvanshi, D., Kinnear, K., and D. Kukrety, "DHCPv6 Active Leasequery", RFC 7653, DOI 10.17487/RFC7653, October 2015, <https://www.rfc-editor.org/info/rfc7653>.

[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>.

[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>.

[RFC8415bis] Mrugalski, T., Volz, B., Richardson, M., Jiang, S., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", Work in Progress, Internet-Draft, draft-ietf-dhc-rfc8415bis-05, 8 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-dhc-rfc8415bis-05>.

[RFC8501] Howard, L., "Reverse DNS in IPv6 for Internet Service Providers", RFC 8501, DOI 10.17487/RFC8501, November 2018, <https://www.rfc-editor.org/info/rfc8501>.

[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>.

[SNAC-SIMPLE] Lemon, T. and J. Hui, "Automatically Connecting Stub Networks to Unmanaged Infrastructure", Work in Progress, Internet-Draft, draft-ietf-snac-simple-05, 8 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-snac-simple-05>.

Appendix A. Multiple Addresses Considerations

While a typical IPv4 host normally has only one IPv4 address per interface, an IPv6 device almost always has multiple addresses assigned to its interface. At the very least, a host can be expected

to have one link-local address, one temporary address, and, in most cases, one stable global address. On a network providing NAT64 service, an IPv6-only host running the 464XLAT customer-side translator (CLAT) [RFC6877] would use a dedicated 464XLAT address, configured via SLAAC (see Section 6.3 of [RFC6877]), which brings the total number of addresses to four. Other common scenarios where the number of addresses per host interface might increase significantly include but are not limited to:

* Devices running containers or namespaces: each container or namespace would have multiple addresses as described above. As a result, a device running just a few containers in a bridge mode can easily have 20 or more IPv6 addresses on the given link.

* Networks assigning multiple prefixes to a given link: multihomed networks, networks using Unique Local IPv6 Unicast Addresses (ULA, [RFC4193]) and non-ULA prefixes together, or networks performing a graceful renumbering from one prefix to another.

[RFC7934] discusses this aspect and explicitly states that IPv6 deployments SHOULD NOT limit the number of IPv6 addresses a host can have. However, it has been observed that networks often do limit the number of on-link addresses per device, likely in an attempt to protect network resources and prevent DoS attacks.

The most common scenario of network-imposed limitations is ND proxy. Many enterprise-scale wireless solutions implement ND proxy to reduce the amount of broadcast and multicast downstream (AP to clients) traffic and provide SAVI functions. To perform ND proxy, a device usually maintains a table containing IPv6 and MAC addresses of connected clients. At least some implementations have hardcoded limits on how many IPv6 addresses per single MAC such a table can contain. When the limit is exceeded, the behavior is implementation dependent. Some vendors just fail to install an N+1 address to the table. Others delete the oldest entry for this MAC and replace it with the new address. In any case, the affected addresses lose network connectivity without receiving any implicit signal, with traffic being silently dropped.

Acknowledgements

Thanks to Harald Alvestrand, Nick Buraglio, Brian Carpenter, Tim Chown, Roman Danyliw, Gert Doering, David Farmer, Fernando Gont, Joel Halpern, Nick Hilliard, Bob Hinden, Martin Hunek, Erik Kline, Warren Kumari, David Lamparter, Andrew McGregor, Tomek Mrugalski, Alexandre Petrescu, Jurgen Schonwalder, Pascal Thubert, Ole Troan, Eric Vyncke, Eduard Vasilenko, Timothy Winters, Chongfeng Xie, and Peter Yee for the discussions, their input, and all contributions.

Authors' Addresses

Lorenzo Colitti
Google, LLC
Shibuya 3-21-3,
Japan
Email: lorenzo@google.com

Jen Linkova (editor)
Google
1 Darling Island Rd
Pyrmont New South Wales 2009
Australia
Email: furry13@gmail.com, furry@google.com

Xiao Ma (editor)
Google
Shibuya 3-21-3,
Japan
Email: xiaom@google.com