Workgroup: Link-state routing
Internet-Draft:
draft-lamparter-lsr-v6ops-pd-aargh
Published: 26 July 2023
Intended Status: Experimental
Expires: 27 January 2024
Authors: D. Lamparter
NetDEF, Inc.

Prefix Dissemination for Semi-Automatic Addressing and Renumbering

Abstract

Between large enterprise networks that can reasonably use their own IPv6 address space and small home and office networks that do not utilize a complex routing topology, there is an intermediate space where a network may need to utilize a nontrivial routed topology but still connect to the internet in a plain "customer" role, with IPv6 address space being assigned over e.g. DHCPv6-PD [DHCPv6].

This poses a yet-unsolved issue that the prefix(es) assigned by the ISP may change, either frequently due to operational practice, or infrequently on some external events like loss of prefix assignment state. This change in prefix needs to propagate, at minimum, into [ADDRCONF] mechanisms, but frequently also other components like firewalls, naming systems, etc.

Table of Contents

1. Introduction and history
  1.1. Rationale and bootstrap considerations
  1.2. Non-goals
2. Requirements Language
3. Disseminated prefix concept
4. Example scenario
5. Using a disseminated prefix
6. Advertising a disseminated prefix
  6.1. Policy factors in disseminating a prefix
7. OSPFv3 transport
  7.1. Applicable Sub-TLVs
8. IS-IS transport
  8.1. Multi-Topology considerations
  8.2. Applicable Sub-TLVs
9. YANG model
10. Security Considerations
11. Privacy Considerations
12. IANA Considerations
13. References
  13.1. Normative References
  13.2. Informative References
Acknowledgements
Editing notes (TO BE REMOVED)
Author's Address

1. Introduction and history

Renumbering is a longstanding, non-trivial and well-known problem with IPv6 networks. Being driven entirely by real-world operational considerations and conditions - a "pure" IPv6 network in a perfect world need never renumber - has turned into somewhat of a sour note, given that (almost?) no one ever voluntarily renumbers.

There is a large body of applicable context on this topic. The Renumbering Still Needs Work [NEEDSWORK] document has gone as far as spawning a (since concluded) IETF working group, 6renum, dedicated to this topic. Output from this working group includes a Gap Analysis

[RFC7010] document, a Problem Statement [RFC6866], and Scenarios, Considerations and Methods [RFC6879]. Furthermore, the process of adding a new prefix in make-before-break manner before removing an old one was described in [RFC4192].

TBD: [RFC2894]

TBD: fill in homenet and autoconf context too / full-auto [RFC7695]

This document takes a "router-down" perspective without attempting to take all address management and policy out of the operator's hands. For perspective, this relates to above documents as follows:

- Generally, the problem considered here is renumbering routers (RFC5887 Section 3.1 [NEEDSWORK], RFC7010 Section 5.2 and 9.1 [RFC7010]).

- Only communicating available prefixes and making subnets or host addresses out of them is considered here. How to apply the results to an innumerable amount of entities that may need reconfiguring or flushing of state (RFC5887 Section 5.2 [NEEDSWORK]) is a giant of its own, e.g. [LEROY] discusses using macros to convene this. The [NETCONF] ecosystem is also relevant here, both to query prefix information from the routing system, as well as to apply it.

- The very heart of this document is challenging "Static Addresses Imply Static Prefixes" [RFC6866], by allowing static addressing with dynamic prefixes.

- RFC 7010 Section 6.3 (first bullet point, "Self-Contained Configuration in Individual Devices") [RFC7010] already documents the concept of "keywords or variables" that reduce configuration change impact from renumbering by reusing bits "defined as a value once". It proceeds to say that this "still means that every device needs to be individually updated". This document formalizes this very concept and provides a mechanism to automate updating this information.

1.1. Rationale and bootstrap considerations

There are many places and methods that could be used to communicate prefixes to be used in a network. Using the routing system to do this is a better choice than many others for several reasons:

1. At its most basic, to establish IPv6 reachability given a number of (working) lower layer links, two things are needed: addresses and routing information. When the routing domain has come up, the network should work, and if addresses are assigned statically, it hopefully will. This mechanism attempts to not

worsen this situation - the network should work after routing has come up.

2. Performing renumbering through configuration management systems can lead to the configuration changes themselves breaking the ability of the configuration management system to apply configuration. Even if make-before-break semantics are followed throughout, ordering constraints may exist - and be very hard to detect. For example, if new addresses are configured on the management system and some routers, the management system's network stack may decide to (in fact, is likely to) immediately use the new addresses in source address selection. If some firewall device has not been updated yet, it will block these connections. It may also block connections to itself. This is not an unsolvable problem, but it has both significant risk, as well as poor verifiability - even if it works in a test, ordering constraints can be subject to race conditions that may randomly show up in a later execution of the exact same renumbering event.

3. Providing alternate or fallback connectivity specifically for management can be costly. Particularly devices used in small to medium business scenarios - the exact target of this approach - may support a limited routing table size, and may not have any VRF support. Providing parallel ULA addressing doubles the resource need when added next to only one GUA prefix. Physically connecting out-of-band management ports can be very difficult with infrastructure cabling constraints.

4. If address configuration is kept in ephemeral state, this can result in bootstrap problems. Both IS-IS and OSPFv3 will start up on exclusively link-local addresses.

Aside from above applicability considerations, the OSPFv3 and IS-IS link state routing protocols provide discovery and flooding mechanisms that are very desirable for disseminating available prefixes. The limited size and expected count of this data is also a great match for the capabilities of these protocols.

1.2. Non-goals

This document does not attempt any kind of automatic prefix or address assignment in the "second half" of the prefix. It only attempts a mechanism to convey the "first half" of a prefix, and deal with changing it - while keeping the second half firmly under operator control.

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. Disseminated prefix concept

The behavior and encodings described in this document rely on the concept of a disseminated prefix. The semantics of such a prefix are as follows:

* There is a flow of information from "upstream" sources of assignment to "downstream" consumers of addressing information.

* Disseminated prefixes are created by operator configuration and policy, but this policy may not contain the actual prefix and instead reference another upstream source. In the primary case this document attempts to address, one or more prefixes will be picked up from one or more upstream ISPs. DHCPv6 Prefix Delegation and explicit assignment by the ISP are assumed to be the most common sources of a prefix. A randomly generated ULA prefix may also be used, though this would mainly be useful to just apply the same numbering that GUA prefixes receive. ULA prefixes should hopefully not need the renumbering feature.

* Disseminated prefixes can in theory be carried in any arbitrary way across the network. This document describes carrying them in [OSPFv3] or [IS-IS]. Care should be taken both when entering this information into these protocols as well as when passing it outside, to not create "information loops". This is not expected to be a problem since there is a clear upstream to downstream flow direction with disseminated prefixes.

* Prefixes (and host addresses) are carved out of a disseminated prefix for actual use. The bits "filled in" to make these assignments are always configured explicitly. This document does not suggest any kind of automatic numbering. Both prefix and host assignments are called "carve-outs" in the remainder of this document.

* There may be use cases for creating a disseminated prefix as a subnet of another disseminated prefix, but this is not expected to be a primary use case in this document. If this is done, the latter prefix MUST be longer than (covered by) the first prefix to establish a clear hierarchy.

*Disseminated prefixes have a lifetime, which may be inherited from their upstream source (e.g. DHCPv6), or be configured by the operator (e.g. for a static assignment.)

*Additional attributes can be placed on disseminated prefixes to influence how the prefix is (or is not) used. These attributes may be sourced along with the prefix itself (again, DHCPv6) or explicitly configured by the operator. A plain 32-bit opaque integer tag is specified in this document, other types of attributes may be specified.

4. Example scenario

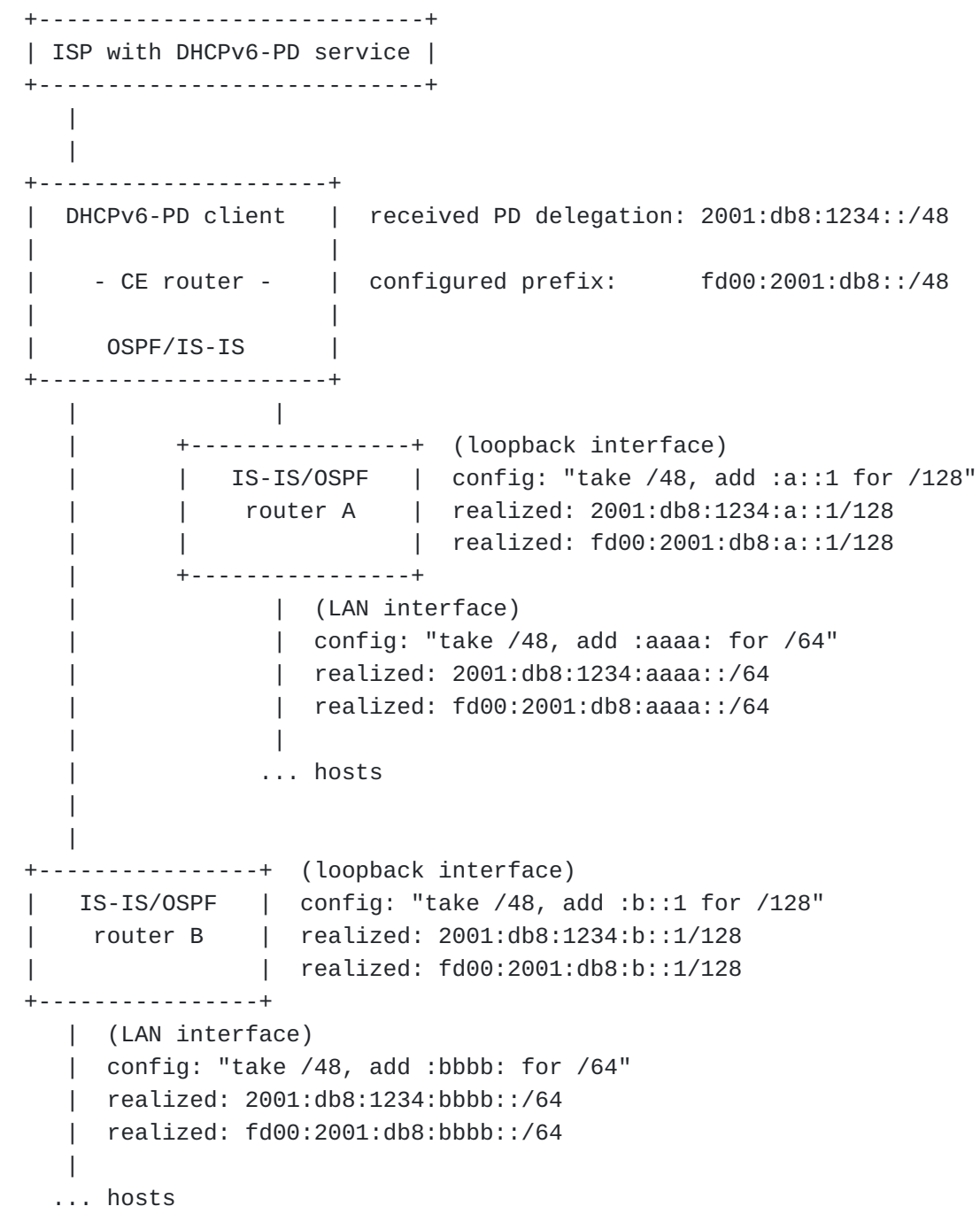


Figure 1: Basic prefix dissemination example

5. Using a disseminated prefix

First and most importantly, disseminated prefixes are not routing information. A system MUST NOT use disseminated prefix as input in any kind of path calculation. However, specific carve-outs MAY be used to configure routing functions, e.g. announce the carve-out into a routing protocol to create reachability. This is purely a function of automated configuration; routing protocols MUST NOT introduce any

special behavior for routing information sourced in this way even if the disseminated prefix was carried by the same routing protocol.

Wherever a disseminated prefix turns into actual prefixes or addresses used is considered a "consumer". This would likely happen on routers participating in the link state routing protocol, but disseminated prefix information MAY also be queried by additional applications (e.g. configuration management systems, name servers, firewalls, etc.) and then realized into use. Any system performing this step is equally a "consumer" and MUST follow the steps outlined below.

To make use of ("realize") a disseminated prefix, a carve-out MUST be created by explicit operator configuration. This requires input to fill in part of or all of the bits left unspecified in the prefix. This input consists of 4 pieces:

1. Minimum length of the disseminated prefix that is valid to make this carve-out from.

2. Target length of the carve-out to be created; this must be equal to or longer than the minimum length above. Common values are 64 for a subnet to be used for SLAAC and 128 for a host address. Other lengths are also possible e.g. to configure firewall rules for a range of networks or other subdivisions of the network.

3. Values for all bits of the prefix between minimum length and target length.

4. Optionally, additional restrictions/filtering of disseminated prefixes eligible to make this carve-out from, relying on additional attributes the disseminated prefix may carry, e.g. the opaque tag. Other metadata may also influence this filtering, e.g. reachability of the originating router, age of the disseminated prefix, or a limit on the number of disseminated prefixes used to realize a carve-outs.

To realize a carve-out, the consumer considers the operator's configuration against all disseminated prefixes available. The consumer MUST NOT arbitrarily limit the disseminated prefixes. In general, one configured carve-out can result in more than one actual prefix if more than one disseminated prefix is available. If there is a limit on the number of actual prefixes / addresses, the consumer MUST allow the operator to configure eligibility conditions and evaluate them against all disseminated prefix. Only if the limit is still not met, the consumer MUST then fall back to using the oldest disseminated prefix available. *(TBD: figure exact behavior.)*

The consumer MUST reevaluate carve-outs whenever a disseminated prefix becomes available, is no longer available, or has a change in

any of its attributes. There MAY be a minor delay or rate limiting on this behavior to protect against overloads or degenerate conditions.

A realized carve-out is ultimately a variable carrying a list of prefixes, made available for use wherever that variable is referenced. A system MAY limit the number of prefixes carried (e.g. to 1 prefix), but MUST support the empty case (no prefix) as it occurs when no no disseminated prefixes are available.

Realizing a carve-out MUST NOT unadvisedly create a disseminated prefix advertisement. Realizing carve-outs is a local process that does not create state in the network at large.

In order to fulfill the above requirements, any consumer residing outside the routing system itself MUST retrieve all disseminated prefix information from the routing system and MUST receive timely updates on change to it. Conversely, if routers provide methods of access to disseminated prefix information (not realized), they MUST include all such information with all of its metadata and attributes.

6. Advertising a disseminated prefix

As outlined in the disseminated prefix semantics, to create an advertisement always requires a configured policy to determine what to disseminate. A system MUST NOT create a disseminated prefix advertisement without operator policy.

A system MAY support consuming disseminated prefixes without support for creating advertisements. However, if capable of creating advertisement, a system SHOULD also support consuming them.

To aid debugging, any system capable of creating disseminated prefix advertisements SHOULD allow creating disseminated prefixes from operators. This also covers the use case where a prefix is statically assigned by an ISP and manually input by an operator.

In principle, it does not matter how some prefix becomes available to the system for disseminating, but the expectation is that this primarily happens through DHCPv6-PD or static assignment. Other methods may require additional considerations.

6.1. Policy factors in disseminating a prefix

The following considerations apply to systems capable of creating a disseminated prefix advertisement:

* There MUST be a method to constrain acceptable prefixes both in their network bits as well as their prefix length, e.g. a prefix-list style filter. (This does not apply to explicit, statically configured prefixes as that is already a constraint.)

*There MUST be some way to limit the number of prefixes disseminated, to prevent resource exhaustion security issues. If the limit is hit, the oldest known prefixes SHOULD be retained over newer ones to reduce churn.

*If a currently disseminated prefix is known to be withdrawn, its advertisement MUST be withdrawn in a timely manner.

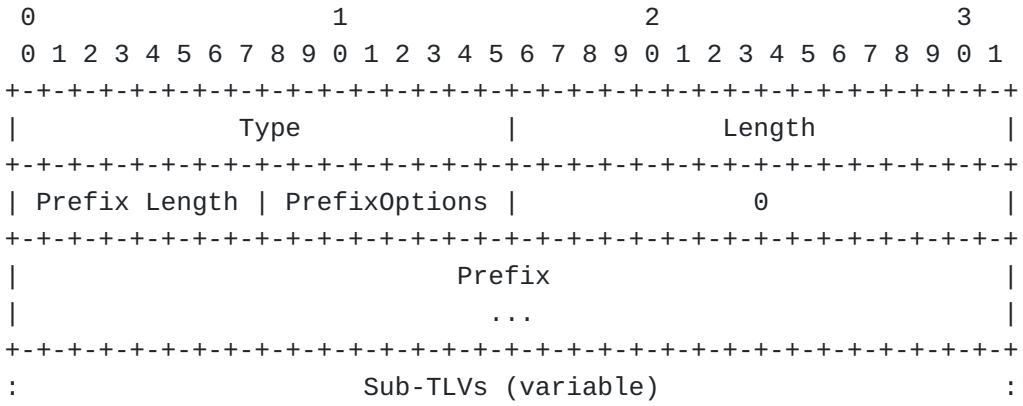
*If the prefix has known lifetimes, it MUST be possible to exclude prefixes below some specified remaining lifetime. This lifetime data MUST be propagated into the disseminated prefix data.

*Other metadata (e.g. from DHCPv6-PD) SHOULD be made available to filter on where useful, and SHOULD be carried in the disseminated prefix if possible.

7. OSPFv3 transport

TBD: which LSA to stick this in? Make a new one? Router Info (12)? Ext. Router LSA (33)? It's only given as a TLV below, but having it be its own LSA is probably better - in particular, with DHCPv6-PD it may need refreshing whenever the lifetime of the delegation is extended - this shouldn't cause routing system load. Needs discussion.

A disseminated prefix is transported in OSPFv3 using an Extended LSA TLV [OSPFv3-EXT] with the following format:



Type to be assigned (not sure if possible for experimental?)

Length *TBD*

Prefix Length, PrefixOptions and Prefix These fields behave as specified in [OSPFv3]. For the PrefixOptions field, all bits MUST be set to zero. *TODO: determine if some are applicable, possibly P and DN?*

TODO: flesh out, figure how exactly to carry lifetime (in TLV?), add sub-TLVs for e.g. DHCPv6 extra details

7.1. Applicable Sub-TLVs

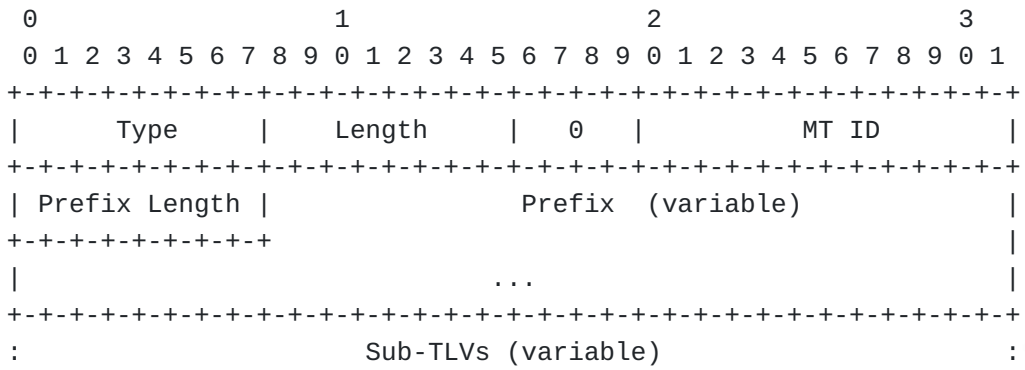
The following preexisting sub-TLVs semantically make sense when nested in the Prefix Dissemination TLV and SHOULD be supported in creating advertisements and filtering for realization:

(3) Route-Tag sub-TLV This sub-TLV may be used to carry operator-defined semantics on a disseminated prefix. Other uses of this sub-TLV limit it to one occurrence, for consistency this requirement is applied here too.

Other sub-TLVs MAY be supported to attach attributes to a disseminated prefix and filter upon them. Note that sub-TLVs present in a disseminated prefix (even explicitly listed above) MUST NOT have any effect on routing behavior by themselves. They also MUST NOT be copied or inherited into any use of realized addresses or prefixes, even if that use reflects back into OSPFv3.

8. IS-IS transport

A disseminated prefix is transported in IS-IS LSPs using a TLV with the following format:



Type to be assigned (not sure if possible for experimental?)

MT ID Topology ID as described in [IS-IS-MT]. There is no variant of this TLV without topology identifier; while this may waste 2 bytes, this trade-off was chosen against the alternative of creating two TLV types with and without MT ID. For applicable values see below.

Length *TBD*

Prefix Length and Prefix The Prefix Length field is given in bits, ranging from 0 to 128. As in RFC5308 Section 2 [IS-ISv6], the

prefix is "packed" and the number of octets used for the prefix is calculated by rounding up to the next byte boundary.

TODO: flesh out, figure how exactly to carry lifetime (maybe in TLV?), add sub-TLVs for e.g. DHCPv6 extra details.

NB: IS-IS TLV (for now) has 255 byte length limit (cf. current drafts on Multi-part / Big TLVs)

8.1. Multi-Topology considerations

Multi-Topology IS-IS is often used to allow separate topologies for IPv4 and IPv6, in which case IPv6 information is carried under MT ID #2. If IPv4 and IPv6 topologies are identical, MT ID #0 may be in use for IPv6.

Implementations of this specification MUST by default place disseminated prefix information in the same topology they use for IPv6 routing, and MUST only process information from that same topology. The topology used MAY be configurable. *TBD: other MT IDs? Exclude multicast and IPv4 specific ones?*

Implementations not supporting IS-IS multi-topology routing MUST use MT ID #0 for disseminated prefix TLVs and MUST ignore any received TLVs of this type with MT ID unequal zero.

8.2. Applicable Sub-TLVs

The following preexisting sub-TLVs semantically make sense when nested in the Prefix Dissemination TLV:

(1) 32-bit Administrative Tag Sub-TLV This sub-TLV may be used to carry operator-defined semantics on a disseminated prefix. The Sub-TLV may appear more than once.

(2) 64-bit Administrative Tag Sub-TLV Same as above.

TODO: (4) Prefix Attribute Flags *for R bit? - if yes then presumably 11 and 12 too (Source Router ID) - need to clear up L1/L2 behavior.*

Other sub-TLVs MAY be supported to attach attributes to a disseminated prefix and filter upon them. Note that any (even explicitly listed above) sub-TLVs present in a disseminated prefix MUST NOT have any effect on routing behavior by themselves. They also MUST NOT be copied or inherited into any use of realized addresses or prefixes, even if that use reflects back into IS-IS.

9. YANG model

well, this certainly needs one, other software will definitely want to pull prefix data out of the routing protocol...

10. Security Considerations

TBD

11. Privacy Considerations

TBD

12. IANA Considerations

TBD - needs codepoints for IS-IS and OSPFv3

13. References

13.1. Normative References

[IS-IS] ISO/IEC, "Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, Second Edition, 2002.

[IS-IS-MT] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <https://www.rfc-editor.org/info/rfc5120>.

[IS-ISv6] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <https://www.rfc-editor.org/info/rfc5308>.

[OSPFv3] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[OSPFv3-EXT] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and F. Baker, "OSPFv3 Link State Advertisement (LSA) Extensibility", RFC 8362, DOI 10.17487/RFC8362, April 2018, <https://www.rfc-editor.org/info/rfc8362>.

[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>.

[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>.

13.2. Informative References

[ADDRCONF] 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>.

[DHCPv6]
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>.

[LEROY] "Preparing network configurations for IPv6 renumbering", International Journal of Network Management, Volume 19, Issue 5, DOI 10.1002/nem.717, 2009, <https://doi.org/10.1002/nem.717>. http://inl.info.ucl.ac.be/system/files/dleroy-nem-2009.pdf

[NEEDSWORK] Carpenter, B., Atkinson, R., and H. Flinck, "Renumbering Still Needs Work", RFC 5887, DOI 10.17487/RFC5887, May 2010, <https://www.rfc-editor.org/info/rfc5887>.

[NETCONF] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC2894] Crawford, M., "Router Renumbering for IPv6", RFC 2894, DOI 10.17487/RFC2894, August 2000, <https://www.rfc-editor.org/info/rfc2894>.

[RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPv6 Network without a Flag Day", RFC 4192, DOI 10.17487/RFC4192, September 2005, <https://www.rfc-editor.org/info/rfc4192>.

[RFC6866] Carpenter, B. and S. Jiang, "Problem Statement for Renumbering IPv6 Hosts with Static Addresses in Enterprise Networks", RFC 6866, DOI 10.17487/RFC6866, February 2013, <https://www.rfc-editor.org/info/rfc6866>.

[RFC6879] Jiang, S., Liu, B., and B. Carpenter, "IPv6 Enterprise Network Renumbering Scenarios, Considerations, and

Methods", RFC 6879, DOI 10.17487/RFC6879, February 2013, <https://www.rfc-editor.org/info/rfc6879>.

[RFC7010] Liu, B., Jiang, S., Carpenter, B., Venaas, S., and W. George, "IPv6 Site Renumbering Gap Analysis", RFC 7010, DOI 10.17487/RFC7010, September 2013, <https://www.rfc-editor.org/info/rfc7010>.

[RFC7695] Pfister, P., Paterson, B., and J. Arkko, "Distributed Prefix Assignment Algorithm", RFC 7695, DOI 10.17487/RFC7695, November 2015, <https://www.rfc-editor.org/info/rfc7695>.

Acknowledgements

TBD, FILL IN

Editing notes (TO BE REMOVED)

This draft lives at https://github.com/eqvinox/pd-aargh

*-00: 2023-07-26 (IETF 117), initial revision.

Author's Address

David 'equinox' Lamparter
NetDEF, Inc.
San Jose,
United States of America

Email: equinox@diac24.net, equinox@opensourcerouting.org