Stream: Internet Engineering Task Force (IETF)
RFC: 8947
Category: Standards Track
Published: December 2020
ISSN: 2070-1721
Authors: B. Volz, *Cisco*; T. Mrugalski, *ISC*; CJ. Bernardos, *UC3M*

RFC 8947
Link-Layer Address Assignment Mechanism for DHCPv6

Abstract

In certain environments, e.g., large-scale virtualization deployments, new devices are created in an automated manner. Such devices may have their link-layer addresses assigned in an automated fashion. With sufficient scale, the likelihood of a collision using random assignment without duplication detection is not acceptable. Therefore, an allocation mechanism is required. This document proposes an extension to DHCPv6 that allows a scalable approach to link-layer address assignments where preassigned link-layer address assignments (such as by a manufacturer) are not possible or are unnecessary.

Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8947.

Copyright Notice

Table of Contents

1. Introduction

There are several deployment types that deal with a large number of devices that need to be initialized. One of them is a scenario where virtual machines (VMs) are created on a massive scale. Typically, the new VM instances are assigned a link-layer address, but random assignment does not scale well due to the risk of a collision (see Appendix A.1 of [RFC4429]). Another use case is Internet of Things (IoT) devices (see [RFC7228]). The huge number of such devices could strain the IEEE's available Organizationally Unique Identifier (OUI) global address space. While there is typically no need to provide global link-layer address uniqueness for such devices, a link-layer assignment mechanism allows for conflicts to be avoided inside an administrative domain. For those reasons, it is desired to have some form of mechanism that would be able to assign locally unique Media Access Control (MAC) addresses.

This document proposes a new mechanism that extends DHCPv6 operation to handle link-layer address assignments.

Since DHCPv6 [RFC8415] is a protocol that can allocate various types of resources (non-temporary addresses, temporary addresses, prefixes, as well as many options) and has the necessary infrastructure to maintain such allocations (numerous server and client implementations, large deployed relay infrastructure, and supportive solutions such as leasequery and failover), it is a good candidate to address the desired functionality.

While this document presents a design that should be usable for any link-layer address type, some of the details are specific to IEEE 802 48-bit MAC addresses [IEEEStd802]. Future documents may provide specifics for other link-layer address types.

IEEE 802 originally set aside half of the 48-bit MAC address space for local use (where the Universal/Local (U/L) bit is set to 1). In 2017, IEEE published an amendment [IEEEStd802c] that divides this space into quadrants with differentiated address rules. More details are in Appendix A.

IEEE is also developing protocols and procedures for assignment of locally unique addresses (IEEE 802.1CQ). This work may serve as an alternative protocol for assignment. For additional background, see [IEEE-P802.1CQ-Project].

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

The DHCP terminology relevant to this specification from [RFC8415] applies here. The following definitions either modify those definitions as to how they are used in this document or define new terminology used herein.

address — Unless specified otherwise, a link-layer (or MAC) address, as specified in [IEEEStd802]. The address is typically six octets long, but some network architectures may use different lengths.

address block — A number of consecutive link-layer addresses. An address block is expressed as a first address plus a number that designates the number of additional (extra) addresses. A single address can be represented by the address itself and zero extra addresses.

client — A node that is interested in obtaining link-layer addresses. It implements the basic DHCP mechanisms needed by a DHCP client, as described in [RFC8415], and supports the new options specified in this document (IA_LL and LLADDR). The client may or may not support IPv6 address assignment and prefix delegation, as specified in [RFC8415].

IA_LL — Identity Association for Link-Layer Address, an identity association (IA) used to request or assign link-layer addresses. See Section 11.1 for details on the IA_LL option.

LLADDR — Link-layer address option that is used to request or assign a block of link-layer addresses. See Section 11.2 for details on the LLADDR option.

server — A node that manages link-layer address allocation and is able to respond to client queries. It implements basic DHCP server functionality, as described in [RFC8415], and supports the new options specified in this document (IA_LL and LLADDR). The server may or may not support IPv6 address assignment and prefix delegation as specified in [RFC8415].

4. Deployment Scenarios

This mechanism is designed to be generic and usable in many deployments, but there are two scenarios it attempts to address in particular: (i) proxy client mode and (ii) direct client mode.

4.1. Scenario: Proxy Client Mode

This mode is used when an entity acts as a DHCP client that requests that available DHCP servers assign one or more addresses (an address block) for the DHCP client to then assign to the final end devices to use. Large-scale virtualization is one application scenario for proxy client mode. In such environments, this entity is often called a "hypervisor" and is frequently required to spawn new VMs. The hypervisor needs to assign new addresses to those machines. The

hypervisor does not use those addresses for itself, but rather it uses them to create new VMs with appropriate addresses. It is worth pointing out the cumulative nature of this scenario. Over time, the hypervisor is likely to increase its address use. Some obsolete VMs will be deleted; their addresses are potentially eligible for reuse by new VMs.

4.2. Scenario: Direct Client Mode

This mode can be used when an entity acts as a DHCP client that requests that available DHCP servers assign one or more addresses (an address block) for its own use. This usage scenario is related to IoT (see Section 1). Upon first boot, for each interface, the device uses a temporary address, as described in [IEEEStd802.11] and IEEE 802.1CQ [IEEE-P802.1CQ-Project], to send initial DHCP packets to available DHCP servers wherein the device requests a single address for that network interface. Once the server assigns an address, the device abandons its temporary address and uses the assigned (leased) address.

Note that a client that operates as above that does not have a globally unique link-layer address on any of its interfaces **MUST NOT** use a link-layer-based DHCP Unique Identifier (DUID). For more details, refer to Section 11 of [RFC8415].

Also, a client that operates as above may run into issues if the switch it is connected to prohibits or restricts link-layer address changes. This may limit where this capability can be used or may require the administrator to adjust the configuration of the switch(es) to allow a change in address.

5. Mechanism Overview

In the scenarios described in Section 4, the protocol operates in fundamentally the same way. The device requesting an address, acting as a DHCP client, will send a Solicit message with an IA_LL option to all available DHCP servers. That IA_LL option **MUST** include an LLADDR option specifying the link-layer-type and link-layer-len, and it may include a specific address or address block as a hint for the server. Each available server responds with either a Reply message with committed address(es) (if Rapid Commit was requested and honored) or an Advertise message with offered address(es). The client selects a server's response, as governed by [RFC8415]. If necessary, the client sends a Request message; the target server will then assign the address(es) and send a Reply message. Once a Reply is received, the client can start using those address(es).

Normal DHCP mechanisms are in use. The client is expected to periodically renew the addresses as governed by T1 and T2 timers and to stop using the address once the valid lifetime expires. Renewals can be administratively disabled by the server sending "infinity" as the T1 and T2 values (see Section 7.7 of [RFC8415]). An administrator may make the address assignment permanent by specifying use of the "infinity" valid lifetime, as defined in Section 7.7 of [RFC8415].

The client can release addresses when they are no longer needed by sending a Release message (see Section 18.2.7 of [RFC8415]).

Figure 9 in [RFC8415] shows a timeline diagram of the messages exchanged between a client and two servers for the typical life cycle of one or more leases.

Confirm and Information-request messages are not used in link-layer address assignment. Decline should technically never be needed, but see Section 12 for one situation where this message is needed.

Clients implementing this mechanism **SHOULD** use the Rapid Commit option, as specified in Sections 5.1 and 18.2.1 of [RFC8415], to obtain addresses with a two-message exchange when possible.

Devices supporting this proposal **MAY** support the reconfigure mechanism, as defined in Section 18.2.11 of [RFC8415]. If supported by both server and client, the reconfigure mechanism allows the administrator to immediately notify clients that the configuration has changed and triggers retrieval of relevant changes immediately, rather than after the T1 timer elapses. Since this mechanism requires implementation of Reconfiguration Key Authentication Protocol (see Section 20.4 of [RFC8415]), small-footprint devices may choose not to support it.

6. Design Assumptions

One of the essential aspects of this mechanism is its cumulative nature, especially in the hypervisor scenario. The server-client relationship does not look like other DHCP transactions in the hypervisor scenario. In a typical environment, there would be one server and a rather small number of hypervisors, possibly even only one. However, over time, the number of addresses requested by the hypervisor(s) will increase as more VMs are spawned.

Another aspect crucial for efficient design is the observation that a single client acting as hypervisor will likely use thousands of addresses. An approach similar to what is used for IPv6 address or prefix assignment (IA container with all assigned addresses listed, one option for each address) would not work well. Therefore, the mechanism should operate on address blocks rather than single values. A single address can be treated as an address block with just one address.

The DHCP mechanisms are reused to a large degree, including message and option formats, transmission mechanisms, relay infrastructure, and others. However, a device wishing to support only link-layer address assignment is not required to support full DHCP. In other words, the device may support only assignment of link-layer addresses but not IPv6 addresses or prefixes.

7. Information Encoding

A client **MUST** send an LLADDR option encapsulated in an IA_LL option to specify the link-layer-type and link-layer-len values. For link-layer-type 1 (Ethernet) and 6 (IEEE 802 Networks), a client sets the link-layer-address field to:

1. All zeroes if the client has no hint as to the starting address of the unicast address block. This address has the IEEE 802 individual/group bit set to 0 (individual).

2. Any other value to request a specific block of address starting with the specified address.

Encoding information for other link-layer-types may be added in the future by publishing an RFC that specifies those values.

A client sets the extra-addresses field to either 0 for a single address or the size of the requested address block minus 1.

A client **MUST** set the valid-lifetime field to 0 (this field **MUST** be ignored by the server).

8. Requesting Addresses

The addresses are assigned in blocks. The smallest block is a single address. To request an assignment, the client sends a Solicit message with an IA_LL option inside. The IA_LL option **MUST** contain an LLADDR option, as specified in Section 7.

The server, upon receiving an IA_LL option, inspects its content and may offer an address or addresses for each LLADDR option according to its policy. The server **MAY** take into consideration the address block requested by the client in the LLADDR option. However, the server **MAY** choose to ignore some or all parameters of the requested address block. In particular, the server may send either a different starting address or a smaller number of addresses than requested. The server sends back an Advertise message with an IA_LL option containing an LLADDR option that specifies the addresses being offered. If the server is unable to provide any addresses, it **MUST** return the IA_LL option containing a Status Code option (see Section 21.13 of [RFC8415]) with status set to NoAddrsAvail.

Note that servers that do not support the IA_LL option will ignore the option and not return it in Advertise (and Reply) messages. Clients that send IA_LL options **MUST** treat this as if the server returned the NoAddrsAvail status for these IA_LL option(s).

The client waits for available servers to send Advertise responses and picks one server, as defined in Section 18.2.9 of [RFC8415]. The client then sends a Request message that includes the IA_LL container option with the LLADDR option copied from the Advertise message sent by the chosen server.

The client **MUST** process the address block(s) returned in the Advertise, rather than what it included in the Solicit message, and may consider the offered address block(s) in selecting the Advertise message to accept. The server may offer a smaller number of addresses or different addresses from those requested. A client **MUST NOT** use resources returned in an Advertise message except to select a server and in sending the Request message to that server; resources are only useable by a client when returned in a Reply message.

Upon reception of a Request message with the IA_LL container option, the server assigns the requested addresses. The server allocates a block of addresses according to its configured policy. The server **MAY** assign a different block or smaller block size than requested in the Request message. The server then generates and sends a Reply message back to the client.

Upon receiving a Reply message, the client parses the IA_LL container option and may start using all provided addresses. It **MUST** restart its T1 and T2 timers using the values specified in the IA_LL option.

The client **MUST** use the address block(s) returned in the Reply message, which may be a smaller block(s) or may have a different address(es) than requested.

A client that has included a Rapid Commit option in the Solicit message may receive a Reply in response to the Solicit message and skip the Advertise and Request message steps above (see Section 18.2.1 of [RFC8415]).

A client that changes its link-layer address on an interface **SHOULD** follow the recommendations in Section 7.2.6 of [RFC4861] to inform its neighbors of the new link-layer address quickly.

9. Renewing Addresses

Address renewals follow the normal DHCP renewals processing described in Section 18.2.4 of [RFC8415]. Once the T1 timer elapses, the client starts sending Renew messages with the IA_LL option containing an LLADDR option for the address block being renewed. The server responds with a Reply message that contains the renewed address block. The server **MUST NOT** shrink or expand the address block. Once a block is assigned and has a non-zero valid lifetime, its size, starting address, and ending address **MUST NOT** change.

If the requesting client needs additional addresses (e.g., in the hypervisor scenario because addresses need to be assigned to new VMs), it **MUST** send an IA_LL option with a different Identity Association IDentifier (IAID) to create another "container" for more addresses.

If the client is unable to renew before the T2 timer elapses, it starts sending Rebind messages, as described in Section 18.2.5 of [RFC8415].

10. Releasing Addresses

The client may decide to release a leased address block. A client **MUST** release the block in its entirety. A client releases an address block by sending a Release message that includes an IA_LL option containing the LLADDR option for the address block to release. The Release transmission mechanism is described in Section 18.2.7 of [RFC8415].

Note that if the client is releasing the link-layer address it is using, it **MUST** stop using this address before sending the Release message (as per [RFC8415]). In order to send the Release message, the client **MUST** use another address (such as the one originally used to initiate DHCPv6 to provide an allocated link-layer address).

11. Option Definitions

This mechanism uses an approach similar to the existing mechanisms in DHCP. There is one container option (the IA_LL option) that contains the actual address or addresses, represented by an LLADDR option. Each LLADDR option represents an address block, which is expressed as a first address with a number that specifies how many additional addresses are included.

11.1. Identity Association for Link-Layer Addresses Option

The Identity Association for Link-Layer Addresses option (the IA_LL option) is used to carry an IA_LL, the parameters associated with the IA_LL, and the address blocks associated with the IA_LL.

The format of the IA_LL option is:

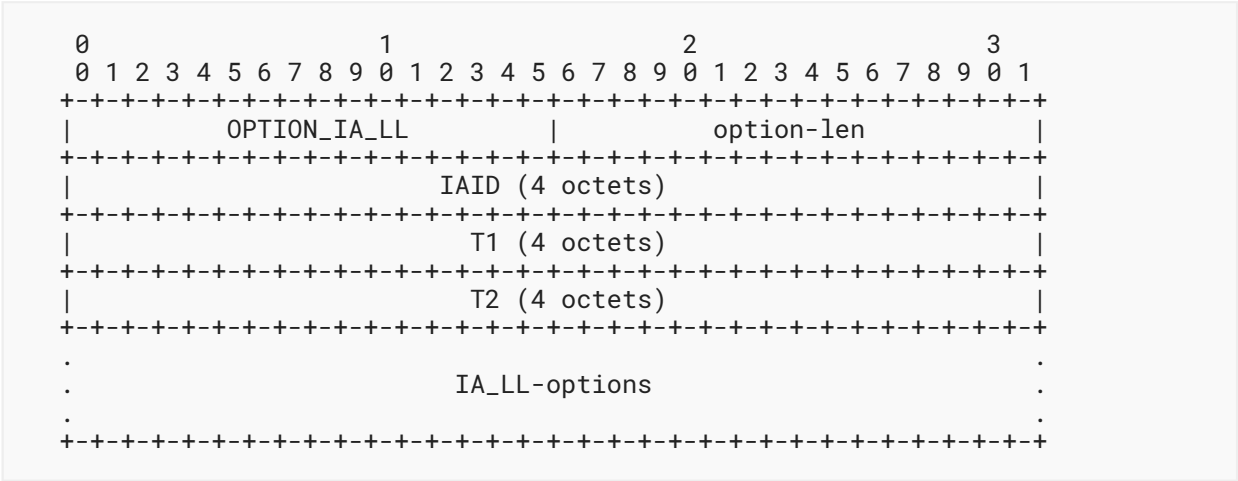

Figure 1: IA_LL Option Format

option-code OPTION_IA_LL (138).

option-len 12 + length of IA_LL-options field.

IAID The unique identifier for this IA_LL; the IAID must be unique among the identifiers for all of this client's IA_LLs. The number space for IA_LL IAIDs is separate from the number space for other IA option types (i.e., IA_NA, IA_TA, and IA_PD). A 4-octet field containing an unsigned integer.

T1 The time interval after which the client should contact the server from which the addresses in the IA_LL were obtained to extend the valid lifetime of the addresses assigned to the IA_LL; T1 is a time duration relative to the current time expressed in units of seconds. A 4-octet field containing an unsigned integer.

T2	The time interval after which the client should contact any available server to extend the valid lifetime of the addresses assigned to the IA_LL; T2 is a time duration relative to the current time expressed in units of seconds. A 4-octet field containing an unsigned integer.
IA_LL-options	Options associated with this IA_LL. A variable-length field (12 octets less than the value in the option-len field).

An IA_LL option may only appear in the options area of a DHCP message. A DHCP message may contain multiple IA_LL options (though each must have a unique IAID).

The status of any operations involving this IA_LL is indicated in a Status Code option (see Section 21.13 of [RFC8415]) in the IA_LL-options field.

Note that an IA_LL has no explicit "lifetime" or "lease length" of its own. When the valid lifetimes of all of the addresses in an IA_LL have expired, the IA_LL can be considered to be expired. T1 and T2 are included to give servers explicit control over when a client recontacts the server about a specific IA_LL.

In a message sent by a client to a server, the T1 and T2 fields **MUST** be set to 0. The server **MUST** ignore any values in these fields in messages received from a client.

In a message sent by a server to a client, the client **MUST** use the values in the T1 and T2 fields for the T1 and T2 times, unless those values in those fields are 0. The values in the T1 and T2 fields are the number of seconds until T1 and T2.

As per Section 7.7 of [RFC8415], the value 0xffffffff is taken to mean "infinity" and should be used carefully.

The server selects the T1 and T2 times to allow the client to extend the lifetimes of any address block in the IA_LL before the lifetimes expire, even if the server is unavailable for some short period of time. Recommended values for T1 and T2 are .5 and .8 times the shortest valid lifetime of the address blocks in the IA that the server is willing to extend, respectively. If the "shortest" valid lifetime is 0xffffffff ("infinity"), the recommended T1 and T2 values are also 0xffffffff. If the time at which the addresses in an IA_LL are to be renewed is to be left to the discretion of the client, the server sets T1 and T2 to 0. The client **MUST** follow the rules defined in Section 14.2 of [RFC8415].

If a client receives an IA_LL with T1 greater than T2, and both T1 and T2 are greater than 0, the client discards the IA_LL option and processes the remainder of the message as though the server had not included the invalid IA_LL option.

The IA_LL-options field typically contains one or more LLADDR options (see Section 11.2). If a client does not include an LLADDR option in a Solicit or Request message, the server **MUST** treat this as a request for a single address and that the client has no hint as to the address it would like.

11.2. Link-Layer Addresses Option

The Link-Layer Addresses option is used to specify an address block associated with an IA_LL. The option must be encapsulated in the IA_LL-options field of an IA_LL option. The LLaddr-options field encapsulates those options that are specific to this address block.

The format of the Link-Layer Addresses option is:

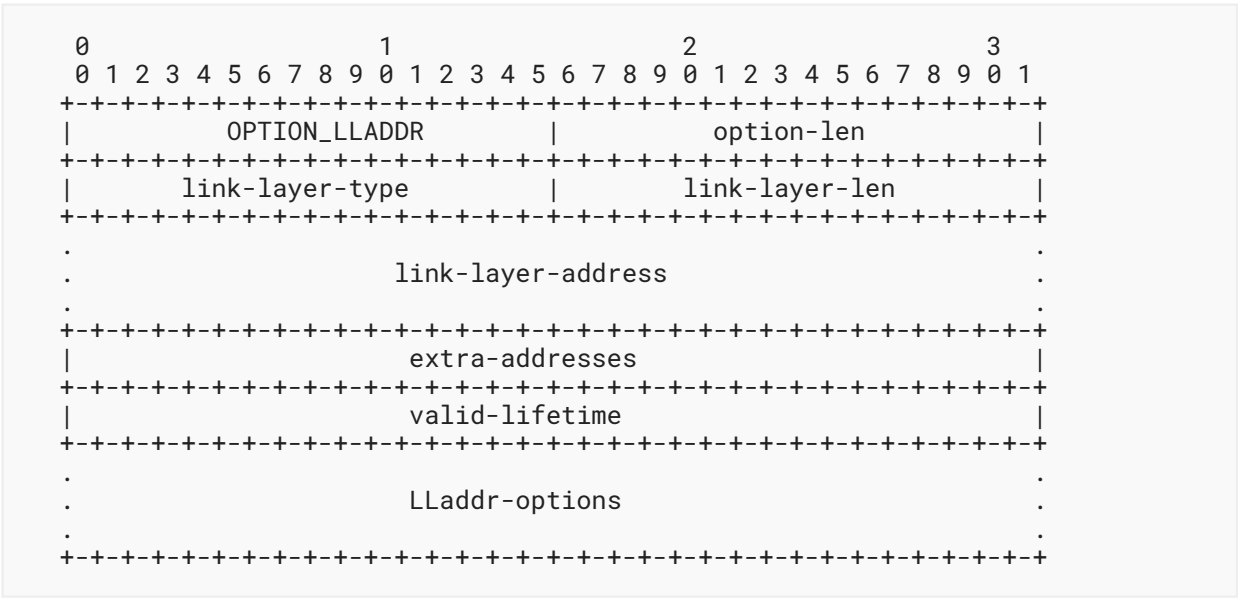

Figure 2: LLADDR Option Format

option-code
: OPTION_LLADDR (139).

option-len
: 12 + link-layer-len field value + length of LLaddr-options field. Assuming a link-layer-address length of 6 and no extra options, the option-len would be 18.

link-layer-type
: The link-layer type **MUST** be a valid hardware type assigned by IANA, as described in [RFC5494], and registered in the "Hardware Types" registry at <https://www.iana.org/assignments/arp-parameters>. A 2-octet field containing an unsigned integer.

link-layer-len
: Specifies the length, in octets, of the link-layer-address field (typically 6 for a link-layer-type of 1 (Ethernet) and 6 (IEEE 802 Networks)). This is to accommodate link layers that may have variable-length addresses. A 2-octet field containing an unsigned integer.

link-layer-address	Specifies the address of the first link-layer address that is being requested or assigned depending on the message. A client **MAY** send a special value to request any address. For link-layer types 1 and 6, see Section 7 for details on this field. A link-layer-len length octet field containing an address.
extra-addresses	Specifies the number of additional addresses that follow the address specified in link-layer-address. For a single address, 0 is used. For example, link-layer-address 02:04:06:08:0a and extra-addresses 3 designate a block of four addresses, starting from 02:04:06:08:0a and ending with 02:04:06:08:0d (inclusive). A 4-octet field containing an unsigned integer.
valid-lifetime	The valid lifetime for the address(es) in the option, expressed in units of seconds. A 4-octet field containing an unsigned integer.
LLaddr-options	Any encapsulated options that are specific to this particular address block. Currently, there are no such options defined, but there may be in the future.

In a message sent by a client to a server, the valid lifetime field **MUST** be set to 0. The server **MUST** ignore any received value.

In a message sent by a server to a client, the client **MUST** use the value in the valid lifetime field for the valid lifetime for the address block. The value in the valid lifetime field is the number of seconds remaining in the lifetime.

As per Section 7.7 of [RFC8415], the valid lifetime of 0xffffffff is taken to mean "infinity" and should be used carefully.

More than one LLADDR option can appear in an IA_LL option.

12. Selecting Link-Layer Addresses for Assignment to an IA_LL

A server selects link-layer addresses to be assigned to an IA_LL according to the assignment policies determined by the server administrator and the requirements of that address space.

Link-layer addresses are typically specific to a link and the server **SHOULD** follow the steps in Section 13.1 of [RFC8415] to determine the client's link.

For IEEE 802 MAC addresses (see [IEEEStd802] as amended by [IEEEStd802c]):

1. Server administrators **SHOULD** follow the IEEE 802 Specifications with regard to the unicast address pools made available for assignment (see Appendix A and [IEEEStd802c]) -- only address space reserved for local use or with the authorization of the assignee may be used.
2. Servers **MUST NOT** allow administrators to configure address pools that would cross the boundary of 2^{42} bits (for 48-bit MAC addresses) to avoid issues with changes in the first octet of the address and the special bits therein (see Appendix A). Clients **MUST** reject assignments where the assigned block would cross this boundary (they **MUST** decline the allocation -- see Section 18.2.8 of [RFC8415]).

3. A server **MAY** use options supplied by a relay agent or client to select the quadrant (see Appendix A) from which addresses are to be assigned. This **MAY** include options like those specified in [RFC8948].

13. IANA Considerations

IANA has assigned the OPTION_IA_LL (138) option code from the "Option Codes" subregistry of the "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)" registry maintained at <http://www.iana.org/assignments/dhcpv6-parameters>:

Value: 138
Description: OPTION_IA_LL
Client ORO: No
Singleton Option: No
Reference: RFC 8947

IANA has assigned the OPTION_LLADDR (139) option code from the "Option Codes" subregistry of the "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)" registry maintained at <http://www.iana.org/assignments/dhcpv6-parameters>:

Value: 139
Description: OPTION_LLADDR
Client ORO: No
Singleton Option: No
Reference: RFC 8947

14. Security Considerations

See Section 22 of [RFC8415] and Section 23 of [RFC7227] for the DHCP security considerations. See [RFC8200] for the IPv6 security considerations.

As discussed in Section 22 of [RFC8415]:

> DHCP lacks end-to-end encryption between clients and servers; thus, hijacking, tampering, and eavesdropping attacks are all possible as a result.

In some network environments, it is possible to secure them, as discussed later in Section 22 of [RFC8415].

If not all parties on a link use this mechanism to obtain an address from the space assigned to the DHCP server, there is the possibility of the same link-layer address being used by more than one device. Note that this issue would exist on these networks even if DHCP were not used to obtain the address.

Server implementations **SHOULD** consider configuration options to limit the maximum number of addresses to allocate (both in a single request and in total) to a client. However, note that this does not prevent a bad client actor from pretending to be many different clients and consuming all available addresses.

15. Privacy Considerations

See Section 23 of [RFC8415] for the DHCP privacy considerations.

For a client requesting a link-layer address directly from a server, as the address assigned to a client will likely be used by the client to communicate on the link, the address will be exposed to those able to listen in on this communication. For those peers on the link that are able to listen in on the DHCP exchange, they would also be able to correlate the client's identity (based on the DUID used) with the assigned address. Additional mechanisms, such as the ones described in [RFC7844], can also be used to improve anonymity by minimizing what is exposed.

As discussed in Section 23 of [RFC8415], DHCP servers and hypervisors may need to consider the implications of assigning addresses sequentially. Though in general, this is only of link-local concern unlike for IPv6 address assignment and prefix delegation, as these may be used for communication over the Internet.

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>.

[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>.

[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>.

[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>.

16.2. Informative References

[IEEE-P802.1CQ-Project] IEEE, "P802.1CQ - Standard for Local and Metropolitan Area Networks: Multicast and Local Address Assignment", <https://standards.ieee.org/project/802_1CQ.html>.

[IEEEStd802] IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture, IEEE Std 802", IEEE STD 802-2014, DOI 10.1109/IEEESTD.2014.6847097, <https://doi.org/10.1109/IEEESTD.2014.6847097>.

[IEEEStd802.11] IEEE, "IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Std 802.11, DOI 10.1109/IEEESTD.2016.7786995, <https://doi.org/10.1109/IEEESTD.2016.7786995>.

[IEEEStd802c] IEEE, "IEEE Standard for Local and Metropolitan Area Networks:Overview and Architecture--Amendment 2: Local Medium Access Control (MAC) Address Usage", IEEE Std 802c-2017, DOI 10.1109/IEEESTD.2017.8016709, <https://doi.org/10.1109/IEEESTD.2017.8016709>.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <https://www.rfc-editor.org/info/rfc2464>.

[RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, DOI 10.17487/RFC4429, April 2006, <https://www.rfc-editor.org/info/rfc4429>.

[RFC5494] Arkko, J. and C. Pignataro, "IANA Allocation Guidelines for the Address Resolution Protocol (ARP)", RFC 5494, DOI 10.17487/RFC5494, April 2009, <https://www.rfc-editor.org/info/rfc5494>.

[RFC7227] Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and S. Krishnan, "Guidelines for Creating New DHCPv6 Options", BCP 187, RFC 7227, DOI 10.17487/RFC7227, May 2014, <https://www.rfc-editor.org/info/rfc7227>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-editor.org/info/rfc7228>.

[RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity Profiles for DHCP Clients", RFC 7844, DOI 10.17487/RFC7844, May 2016, <https://www.rfc-editor.org/info/rfc7844>.

[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>.

[RFC8948] Bernardos, CJ. and A. Mourad, "Structured Local Address Plan (SLAP) Quadrant Selection Option for DHCPv6", RFC 8948, DOI 10.17487/RFC8948, December 2020, <https://www.rfc-editor.org/info/rfc8948>.

Appendix A. IEEE 802c Summary

This appendix provides a brief summary of IEEE 802c [IEEEStd802c].

The original IEEE 802 specifications assigned half of the 48-bit MAC address space to local use -- these addresses have the U/L bit set to 1 and are locally administered with no imposed structure.

In 2017, the IEEE issued the IEEE Std 802c specification, which defines a new optional "Structured Local Address Plan (SLAP) that specifies different assignment approaches in four specified regions of the local MAC address space". Under this plan, there are four SLAP quadrants that use different assignment policies.

The first octet of the MAC address Z and Y bits define the quadrant for locally assigned addresses (X-bit is 1). In IEEE representation, these bits are as follows:

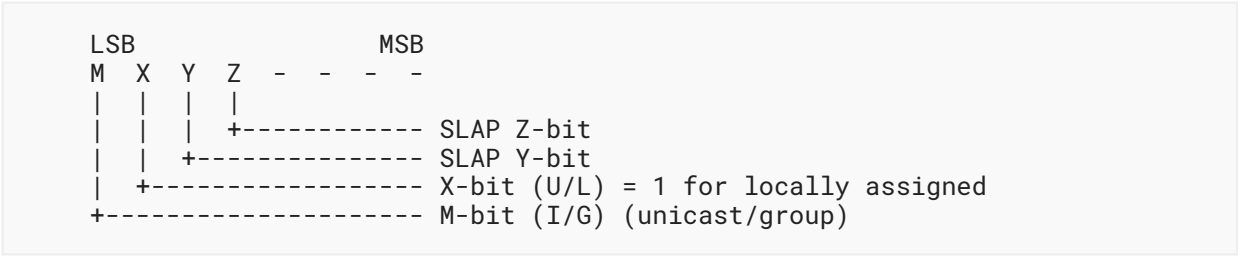

Figure 3: SLAP Bits

The SLAP quadrants are:

Quadrant	Y-bit	Z-bit	Local Identifier Type	Local Identifier
01	0	1	Extended Local	ELI
11	1	1	Standard Assigned	SAI
00	0	0	Administratively Assigned	AAI
10	1	0	Reserved	Reserved

Table 1: SLAP Quadrants

MAC addresses derived from an Extended Local Identifier (ELI) are based on an assigned Company ID (CID), which is 24 bits (including the M, X, Y, and Z bits) for 48-bit MAC addresses. This leaves 24 bits for the locally assigned address for each CID for unicast (M-bit = 0) and also for multicast (M-bit = 1). The CID is assigned by the IEEE Registration Authority (RA).

MAC addresses derived from a Standard Assigned Identifier (SAI) are assigned by a protocol specified in an IEEE 802 standard. For 48-bit MAC addresses, 44 bits are available. Multiple protocols for assigning SAIs may be specified in IEEE standards. Coexistence of multiple protocols may be supported by limiting the subspace available for assignment by each protocol.

MAC addresses derived from an Administratively Assigned Identifier (AAI) are assigned locally. Administrators manage the space as needed. Note that multicast IPv6 packets [RFC2464] use a destination address starting in 33-33, so AAI addresses in that range should not be assigned. For 48-bit MAC addresses, 44 bits are available.

The last quadrant is reserved for future use. While this quadrant may also be used similar to AAI space, administrators should be aware that future specifications may define alternate uses that could be incompatible.

Acknowledgments

Thanks to the DHC Working Group participants that reviewed this document and provided comments and support. With special thanks to Ian Farrer for his thorough reviews and shepherding of this document through the IETF process. Thanks also to directorate reviewers Samita Chakrabarti, Roni Even, and Tianran Zhou and IESG members Martin Duke, Benjamin Kaduk, Murray Kucherawy, Warren Kumari, Barry Leiba, Alvaro Retana, Éric Vyncke, and Robert Wilton for their suggestions. And thanks to Roger Marks, Robert Grow, and Antonio de la Oliva for comments related to IEEE work and references.

Authors' Addresses

Bernie Volz
Cisco Systems, Inc.
300 Beaver Brook Rd
Boxborough, MA 01719
United States of America
Email: volz@cisco.com

Tomek Mrugalski
Internet Systems Consortium, Inc.
PO Box 360
Newmarket, NH 03857
United States of America
Email: tomasz.mrugalski@gmail.com

Carlos J. Bernardos
Universidad Carlos III de Madrid
Av. Universidad, 30
28911 Leganes, Madrid
Spain
Phone: +34 91624 6236
Email: cjbc@it.uc3m.es
URI: http://www.it.uc3m.es/cjbc/