Dynamic Host Configuration (DHC)
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018
B. Volz
Cisco
T. Mrugalski
ISC
March 5, 2018

Link-Layer Addresses Assignment Mechanism for DHCPv6
draft-bvtm-dhc-mac-assign-00

Abstract

In certain environments, e.g. large scale virtualization deployments, new devices are created in an automated manner. Such devices typically have their link-layer (MAC) addresses randomized. With sufficient scale, the likelihood of collision is not acceptable. Therefore an allocation mechanism is required. This draft proposes an extension to DHCPv6 that allows a scalable approach to link-layer address assignments.

Table of Contents

1. Introduction

There are several new 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 random link-layer (MAC) address, but that does not scale well due to the birthday paradox. Another use case is IoT devices. Typically there is no need to provide global uniqueness of MAC addresses for such devices. On the other hand, the huge number of such devices would likely exhaust a vendor's OUI (Organizationally Unique Identifier) global address space. For those reasons, it is desired to have some form of local authority that would be able to assign MAC addresses.

The IEEE originally set aside half of the 48-bit MAC Address space for local use (where the U/L bit is set to 1). In 2017, the IEEE specified an optional specification (IEEE 802c) that divides this space into quadrants (Standards Assigned Identifier, Extended Local Identifier, Administratively Assigned Identifier, and a Reserved

quadrant) - more details are in Appendix A. The IEEE is also working to specify protocols and procedures for assignment of locally unique addresses (IEEE 802.1cq). This work may serve as one such protocol for assignment. For additional background, see [IEEE-802-Tutorial].

This document proposes a new mechanism that extends DHCPv6 operation to handle MAC address assignments.

Since DHCPv6 ([I-D.ietf-dhc-rfc3315bis]) is a protocol that can allocate various types of resources (non-temporary addresses, temporary addresses, prefixes, but also many options) and has necessary infrastructure (numerous server and client implementations, large deployed relay infrastructure, supportive solutions, like leasequery and failover) to maintain such assignment, it is a good candidate to address the desired functionality.

2. Requirements

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 DHCPv6 terminology relevant to this specification from the DHCPv6 Protocol [I-D.ietf-dhc-rfc3315bis] applies here.

client
: A device that is interested in obtaining link-layer addresses. It implements the basic DHCPv6 mechanisms needed by a DHCPv6 client as described in [I-D.ietf-dhc-rfc3315bis] and supports the new options (IA_LL and LLADDR) specified in this document. The client may or may not support address assignment and prefix delegation as specified in [I-D.ietf-dhc-rfc3315bis].

server
: Software that manages link-layer address allocation and is able to respond to client queries. It implements basic DHCPv6 server functionality as described in [I-D.ietf-dhc-rfc3315bis] and supports the new options (IA_LL and LLADDR) specified in this document. The server may or may not support address assignment and prefix

delegation as specified in [I-D.ietf-dhc-rfc3315bis].

address — Unless specified otherwise, an address means a link-layer (or MAC) address, as defined in IEEE802. The address is typically 6 bytes 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.

4. Mechanism Overview

This mechanism is designed to be generic and usable in many deployments, but there are two scenarios it attempts to address in particular. The first one pertains to large scale virtualization. In such environments the governing entity is often called a hypervisor and is frequently required to spawn new VMs. The hypervisor needs to assign new MAC addresses to those machines. It acts as a DHCP client and requests available DHCP servers to assign one or more MAC addresses (an address block). The hypervisor does not use those addresses for itself, but rather uses them to create new VMs with appropriate MAC addresses. It is worth pointing out the cumulative nature of a scenario. Over time, the hypervisor is likely to increase its MAC addresses usage. While some obsolete VMs will be deleted and their MAC addresses will become eligible for release or reuse, it is unexpected for all MAC addresses to be released.

Another usage scenario is related to IoT (Internet of Things). With the emergence of IoT, a new class of cheap, sometimes short lived and disposable devices, has emerged. Examples may include various sensors (e.g. medical) and actuators or controllable LED lights. Upon first boot, the device uses a temporary MAC address, as described in [IEEE-802.11-02/109r0], to send initial DHCP packets to available DHCP servers. Such devices will typically request a single MAC address for each available network interface, which typically means one MAC address per device. Once the server assigns a MAC address, the device abandons its temporary MAC address.

In all scenarios the protocol operates in fundamentally the same way. The device requesting an address, acting as a DHCP client, will send a Solicit message with a IA_LL option to all available DHCP servers.

That IA_LL option MUST include a LLADDR option specifying the link-layer-type and link-layer-len and may specify a specific address or address block as a hint for the server. Each available server responds with an Advertise message with offered link-layer address or addresses. The client then picks the best server, as governed by [I-D.ietf-dhc-rfc3315bis], and will send a Request message. The target server will then assign the link-layer addresses and send a Reply message. Upon reception, the client can start using those link-layer addresses.

Normal DHCP mechanisms are in use. The client is expected to periodically renew the link-layer addresses as governed by T1 and T2 timers. This mechanism can be administratively disabled by the server sending "infinity" as the T1 and T2 values (see Section 7.7 of [I-D.ietf-dhc-rfc3315bis]).

The client can release link-layer addresses when they are no longer needed by sending a Release message (see Section 18.2.7 of [I-D.ietf-dhc-rfc3315bis]).

Confirm, Decline, Reconfigure and Information-Request messages are not used in link-layer address assignment.

Clients implementing this mechanism SHOULD use the Rapid Commit option as specified in Section 5.1 and 18.2.1 of [I-D.ietf-dhc-rfc3315bis].

An administrator may make the address assignment permanent by specifying use of infinite lifetimes, as defined in Section 7.7 of [I-D.ietf-dhc-rfc3315bis]. An administrator may also the disable the need for the renewal mechanism by setting the T1 and T2 values to infinity.

DISCUSSION: A device may send its link-layer address in a LLADDR option to ask the server to register that address to the client (if available), making the assignment permanent for the lease duration. The client MUST be prepared to use a different address if the server choses not to honor its hint.

5. 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. This is especially true 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 MAC addresses requested by the hypervisor(s) will likely increase as new 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. Therefore an approach similar to what is used for 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 DHCPv6 mechanisms are reused to 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 DHCPv6. In other words, the device may support only assignment of link-layer addresses, but not IPv6 addresses or prefixes.

6. Information Encoding

A client MUST send a LLADDR option encapsulated in a IA_LL option to specify the link-layer-type and link-layer-len values. A client sets the link-layer-address field to all zeros if it has no hint as to the starting address of the address block; otherwise it sets this field to the starting address. A client sets the extra-addresses field to either 0 for a single address or to the size of the requested address block minus 1. A client SHOULD set the valid-lifetime field to 0 (as it is ignored by the server).

7. Requesting Addresses

The link-layer addresses are assigned in blocks. The smallest block is a single address. To request an assignment, the client sends a Solicit message with a IA_LL option in the message. The IA_LL option MUST contain a LLADDR option as specified in Section 6.

The server, upon receiving a 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 chose to ignore some or all of the requested address block. In particular, the server may send a different starting address than requested, or grant a smaller number of addresses than requested. The server sends back an Advertise message 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 [I-D.ietf-dhc-rfc3315bis]) with status set to NoAddrsAvail.

The client MUST be able to handle a response that contains an address or addresses different than those requested.

The client waits for available servers to send Advertise responses and picks one server as defined in Section 18.2.9 of [I-D.ietf-dhc-rfc3315bis]. 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.

Upon reception of a Request message with IA_LL container option, the server assigns requested addresses. The server MAY alter the allocation at this time. It 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.

A client that has included a Rapid Commit option in the Solicit, may receive a Reply in response to the Solicit and skip the Advertise and Request steps above (see Section 18.2.1 of [I-D.ietf-dhc-rfc3315bis]).

8. Renewing Addresses

Address renewals follow the normal DHCPv6 renewals processing described in Section 18.2.4 of [I-D.ietf-dhc-rfc3315bis]. Once the T1 timer elapses, the client starts sending Renew messages with the IA_LL option containing a LLADDR option for the address block being renewed. The server responds with a Reply message that contains the renewed address block. The server SHOULD NOT alter the address block being renewed, unless its policy has changed. The server MUST NOT shrink the address block. The server MAY expand the address block by adding additional addresses.

TODO: The growth in hypervisor case. The hypervisor client will start with certain number of MAC addresses and then will keep asking for more over time.

TODO: May have to review this renewing behavior? Server should renew or not the entire address block. Not sure about expanding the block? Earlier (Advertise), we required server to use at most the requested block size; so not sure why to allow expanding for Renew.

If the client is unable to Renew before the T2 timer elapses, it starts sending Rebind messages as described in 18.2.5 of [I-D.ietf-dhc-rfc3315bis].

9. Releasing Addresses

The client may decide to release a leased address block. A client MUST release the whole block in its entirety. A client releases an address block by sending a Release message that includes the 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 [I-D.ietf-dhc-rfc3315bis].

10. 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 link-layer address or addresses, represented by an LLADDR option. Each such option represents an address block, which is expressed as a first address with a number that specifies how many additional addresses are included.

10.1. Identity Association for Link-Layer Addresses Option

The Identity Association for Link-Layer Addresses option (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:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          OPTION_IA_LL         |          option-len           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IAID (4 octets)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         T1 (4 octets)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         T2 (4 octets)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                         IA_LL-options                         .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: IA_LL Option Format

option-code   OPTION_IA_LL (tbd1).

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 four octets long field.

T1   The time at 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 four octets long field.

T2   The time at 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 four octets long field.

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).

Each IA_LL carries one block of link-layer addresses.

The status of any operations involving this IA_LL is indicated in a Status Code option (see Section 21.13 of [I-D.ietf-dhc-rfc3315bis]) 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 as having 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 SHOULD 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 [I-D.ietf-dhc-rfc3315bis]), 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 in [I-D.ietf-dhc-rfc3315bis].

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.

10.2. Link-Layer Addresses Option

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

The format of the Link-Layer Addresses option is:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          OPTION_LLADDR        |           option-len          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |       link-layer-type         |         link-layer-len        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                      link-layer-address                       |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                       extra-addresses                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                       valid-lifetime                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                                                               .
    .                       LLaddr-options                          .
    .                                                               .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: LLADDR Option Format

option-code   OPTION_LLADDR (tbd2).

option-len   12 + link-layer-len field (typically 6) + length of LLaddr-options field. Assuming a typical link-layer address of 6 is used and there are no extra options, length should be equal to 18.

link-layer-type The link-layer type MUST be a valid hardware type assigned by the IANA, as described in [RFC5494]. The type is stored in network byte order.

link-layer-len Specifies the length of the link-layer-address field (typically 6, for a link-layer-type of 1 (Ethernet)). A two octets long field.

link-layer-address Specific value of link-layer address that is being requested. A special case of address consisting of only zeroes means any address. This value can be only sent by a client that requests a new block. In responses from a server, this value specifies the first address allocated.

extra-addresses Number of additional addresses that follow address specified in link-layer-address. For requesting a single address, use 0. For example: link-layer-address: 02:04:06:08:0a and extra-addresses 3

designates a block of 4 addresses, starting from 02:04:06:08:0a (inclusive) and ending with 02:04:06:08:0d (inclusive). In responses from a server, this value specifies the number of additional addresses allocated. A four octets long field.

valid-lifetime The valid lifetime for the address(es) in the option, expressed in units of seconds. A four octets long field.

LLaddr-options any encapsulated options that are specific to this particular address block. Currently there are no such options defined, but they may appear in the future.

In a message sent by a client to a server, the valid lifetime field SHOULD 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 [I-D.ietf-dhc-rfc3315bis], 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.

11. Client Behavior

TODO: We need start this section by clearly defining what 'client' means in this context (either hypervisor acting on behalf of the client to be spawned or the IOT device acting on its own behalf).

12. Server Behavior

TODO: Need to describe server operation. Likely also recommend assigning MAC addresses from an appropriate quadrant (see Appendix).

13. IANA Considerations

IANA is kindly requested to assign new value for options OPTION_LL (tbd1) and OPTION_LLADDR (tbd2) and add those values to the DHCPv6 Option Codes registry maintained at http://www.iana.org/assignments/dhcpv6-parameters.

14. Security Considerations

TODO:

15. Privacy Considerations

TODO:

16. References

16.1. Normative References

[I-D.ietf-dhc-rfc3315bis]
Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) bis", draft-ietf-dhc-rfc3315bis-12 (work in progress), March 2018.

[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>.

16.2. Informative References

[IEEE-802-Tutorial]
Thaler, P., "Emerging IEEE 802 Work on MAC Addressing, https://datatracker.ietf.org/meeting/96/materials/slides-96-edu-ieee802work-0/".

[IEEE-802.11-02/109r0]
Edney, J., Haverinen, H., Honkanen, J-P., and P. Orava, "Temporary MAC address for anonymity, https://mentor.ieee.org/802.11/dcn/02/11-02-0109-00-000i-temporary-mac-address-for-anonymity.ppt".

[IEEEStd802c-2017]
IEEE Computer Society, "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture, Amendment 2: Local Medium Access Control (MAC) Address Usage, IEEE Std 802c-2017".

[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>.

[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>.

Appendix A. IEEE 802c Summary

This appendix provides a brief summary of IEEE802c from [IEEEStd802c-2017].

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 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 4 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:

```
LSB                MSB
M  X  Y  Z  -  -  -  -
|  |  |  |
|  |  |  +------------ SLAP Z-bit
|  |  +--------------- SLAP Y-bit
|  +------------------ X-bit (U/L) = 1 for locally assigned
+--------------------- M-bit (I/G) (unicast/group)
```

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

SLAP Quadrants

Extended Local Identifier (ELI) derived MAC addresses 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 RA.

Standard Assigned Identifier (SAI) derived MAC addresses 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.

Administratively Assigned Identifier (AAI) derived MAC addresses are assigned locally. Administrators manage the space as needed. Note that multicast IPv6 packets ([RFC2464]) use a destination address starting in 33-33 and this falls within this space and therefore should not be used to avoid conflict with IPv6 multicast addresses. For 48-bit MAC addresses, 44 bits are available.

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

Authors' Addresses

Bernie Volz
Cisco Systems, Inc.
1414 Massachusetts Ave
Boxborough, MA 01719
USA

Email: volz@cisco.com

Tomek Mrugalski
Internet Systems Consortium, Inc.
950 Charter Street
Redwood City, CA 94063
USA

Email: tomasz.mrugalski@gmail.com