DHC WG
Internet-Draft
Intended status: Standards Track
Expires: August 31, 2020

CJ. Bernardos
UC3M
A. Mourad
InterDigital

February 28, 2020

SLAP quadrant selection options for DHCPv6
draft-ietf-dhc-slap-quadrant-03

Abstract

The IEEE originally structured the 48-bit MAC address space in such a way that half of it was reserved for local use. Recently, the IEEE has been working on a new specification (IEEE 802c) 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.

The IEEE is working on mechanisms to allocate addresses in the one of these quadrants (IEEE 802.1CQ). There is work also in the IETF on specifying a new mechanism that extends DHCPv6 operation to handle the local MAC address assignments. In this document, we complement this ongoing IETF work by defining a mechanism to allow choosing the SLAP quadrant to use in the allocation of the MAC address to the requesting device/client.

This document proposes extensions to DHCPv6 protocols to enable a DHCPv6 client or a DHCPv6 relay to indicate a preferred SLAP quadrant to the server, so that the server allocates the MAC address to the given client out of the quadrant requested by relay or client.

Table of Contents

1. Introduction

The IEEE originally structured the 48-bit MAC address space in such a way that half of it was reserved for local use (where the U/L bit is set to 1). Recently, the IEEE has been working on a new specification (IEEE 802c [IEEEStd802c-2017]) 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. These four regions, called SLAP quadrants, are briefly described below (see Figure 1 and Figure 2 for details):

- Quadrant "Extended Local Identifier" (ELI) MAC addresses are assigned based on a Company ID (CID), which takes 24-bits, leaving the remaining 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).

- Quadrant "Standard Assigned Identifier" (SAI) MAC addresses are assigned based on 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.

- Quadrant "Administratively Assigned Identifier" (AAI) MAC addresses are assigned locally by an administrator. Multicast IPv6 packets 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.

- Quadrant "Reserved for future use" where MAC addresses may be assigned using new methods yet to be defined, or by an administrator like in the AAI quadrant.

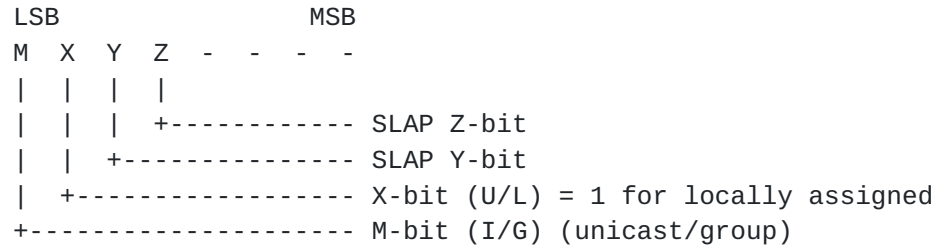

Figure 1: IEEE 48-bit MAC address structure

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

Figure 2: SLAP quadrants

1.1. Problem statement

The IEEE is working on mechanisms to allocate addresses in the SAI quadrant (IEEE 802.1CQ project). There is also ongoing work in the IETF [I-D.ietf-dhc-mac-assign] specifying a new mechanism that extends DHCPv6 operation to handle the local MAC address assignments. In this document, we complement ongoing IETF work with mechanisms to allow choosing the SLAP quadrant to use in the allocation of the MAC address to the requesting device/client. This document proposes extensions to DHCPv6 protocols to enable a DHCPv6 client or a DHCPv6 relay to indicate a preferred SLAP quadrant to the server, so that the server allocates the MAC address to the given client out of the quadrant requested by relay or client.

In the following, we describe two application scenarios where a need arises to assign local MAC addresses according to preferred SLAP quadrants.

1.1.1. WiFi devices

Today, most WiFi devices come with interfaces that have a "burned in" MAC address, allocated from the universal address space using a 24-bit Organizationally Unique Identifier (OUI, assigned to IEEE 802 interface vendors). However, recently, the need to assign local (instead of universal) MAC addresses has emerged in particular in the following two scenarios:

- IoT (Internet of Things): where there are a lot of cheap, sometimes short lived and disposable devices. Examples of this include: sensors and actuators for health or home automation applications. In this scenario, it is common that upon a first boot, the device uses a temporary MAC address, to send initial DHCP packets to available DHCP servers. IoT devices typically request a single MAC address for each available network interface. Once the server assigns a MAC address, the device abandons its temporary MAC address. This type of device is typically not moving. In general, any type of SLAP quadrant would be good for assigning addresses from, but ELI/SAI quadrants might be more suitable in some scenarios, such as if the addresses need to belong to the CID assigned to the IoT communication device vendor.

- Privacy: Today, MAC addresses allow the exposure of users' locations making it relatively easy to track users' movement. One of the mechanisms considered to mitigate this problem is the use of local random MAC addresses, changing them every time the user connects to a different network. In this scenario, devices are typically mobile. Here, AAI is probably the best SLAP quadrant to assign addresses from, as it is the best fit for randomization of

addresses, and it is not required for the addresses to survive when changing networks.

1.1.2. Hypervisor: migratable vs non-migratable functions

In large scale virtualization environments, thousands of virtual machines (VMs) are active. These VMs are typically managed by a hypervisor, in charge of spawning and stopping VMs as needed. The hypervisor is also typically in charge of assigning new MAC addresses to the VMs. If a DHCP solution is in place for that, the hypervisor 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. If we assume very large data center environments, such as the ones that are typically used nowadays, it is expected that the data center is divided in different network regions, each one managing its own local address space. In this scenario, there are two possible situations that need to be tackled:

- Migratable functions. If a VM (providing a given function) needs to be migrated to another region of the data center (e.g., for maintenance, resilience, end-user mobility, etc.), the VM's networking context needs to migrate with the VM. This includes the VM's MAC address(es). Therefore, for this case, it is better to allocate addresses from the ELI/SAI SLAP quadrant, which can be centrally allocated by the DHCP server.

- Non-migratable functions. If a VM will not be migrated to another region of the data center, there are no requirements associated with its MAC address. In this case, it is more efficient to allocate it from the AAI SLAP quadrant, that does not need to be unique across multiple data centers (i.e., each region can manage its own MAC address assignment, without checking for duplicates globally).

2. Terminology

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.

Where relevant, the DHCPv6 terminology from the DHCPv6 Protocol [RFC8415] also applies here. Additionally, the following definitions are updated for this document.

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 [RFC8415] and supports the new options (IA_LL and LLADDR) specified in [I-D.ietf-dhc-mac-assign]. The client may or may not support address assignment and prefix delegation as specified in [RFC8415].

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 [RFC8415] and supports the new options (IA_LL and LLADDR) specified in [I-D.ietf-dhc-mac-assign]. The server may or may not support address assignment and prefix delegation as specified in [RFC8415].

relay — A node that acts as an intermediary to deliver DHCP messages between clients and servers. A relay, based on local knowledge and policies, may include the preferred SLAP quadrant in a QUAD option sent to the server. The relay implements basic DHCPv6 relay agent functionality as described in in [RFC8415] and supports the new options (IA_LL and LLADDR) specified in [I-D.ietf-dhc-mac-assign].

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.

3. Quadrant Selection Mechanisms

The following section describes some examples of how the quadrant preference mechanisms could be used.

Let's take first an IoT scenario as an example. An IoT device might decide on its own the SLAP quadrant it wants to use to obtain a local MAC address, using the following information to take the decision:

- Type of IoT deployment: e.g., industrial, domestic, rural, etc. For small deployments, such as domestic ones, the IoT itself can decide to use the AAI quadrant (this might not even involve the

use of DHCP, by the device just configuring a random address computed by the device itself). For large deployments, such as industrial or rural ones, where thousands of devices might co-exist, the IoT can decide to use the ELI or SAI quadrants.

- Mobility: if the IoT device can move, then it might prefer to select the SAI or AAI quadrants to minimize address collisions when moving to another network. If the device is known to remain fixed, then the ELI is probably the most suitable one to use.

- Managed/unmanaged: depending on whether the IoT device is managed during its lifetime or cannot be re-configured, the selected quadrant might be different. For example, it can be managed, this means that network topology changes might occur during its lifetime (e.g., due to changes on the deployment, such as extensions involving additional devices), and this might have an impact on the preferred quadrant (e.g., to avoid potential collisions in the future).

- Operation/battery lifetime: depending on the expected lifetime of the device a different quadrant might be preferred (as before, to minimize potential address collisions in the future).

The previous are considerations that the device vendor/administrator may wish to use when defining the IoT device's MAC address request policy (i.e., how to select a given SLAP quadrant). IoT devices are typically very resource constrained, so there may only be simple decision making process based on pre-configured preferences.

If we now take the WiFi device scenario, considering for example that a laptop or smartphone connects to a network using its built in MAC address. Due to privacy/security concerns, the device might want to configure a local MAC address. The device might use different parameters and context information to decide, not only which SLAP quadrant to use for the local MAC address configuration, but also when to perform a change of address (e.g., it might be needed to change address several times). This information includes, but it is not limited to:

- Type of network the device is connected: public, work, home.

- Trusted network? Y/N.

- First time visited network? Y/N.

- Network geographical location.

- Mobility? Y/N.

- Operating System (OS) network profile, including security/trust related parameters. Most modern OSs keep metadata associated to the networks they can attach to, as for example the level of trust the user or administrator assigns to the network. This information is used to configure how the device behaves in terms of advertising itself on the network, firewall settings, etc. But this information can also be used to decide whether to configure a local MAC address or not, from which SLAP quadrant and how often.

- Triggers coming from applications regarding location privacy. An app might request to the OS to maximize location privacy (due to the nature of the application) and this might require that the OS forces the use of a local MAC address, or that the local MAC address is changed.

This information can be used by the device to select the SLAP quadrant. For example, if the device is moving around (e.g., while connected to a public network in an airport), it is likely that it might change access point several times, and therefore it is best to minimize the chances of address collision, using the SAI or AAI quadrants. If the device is not moving and attached to a trusted network (e.g. at work), then it is probably best to select the ELI quadrant. These are just some examples of how to use this information to select the quadrant.

Additionally, the information can also be used to trigger subsequent changes of MAC address, to enhance location privacy. Besides, changing the SLAP quadrant used might also be used as an additional enhancement to make it harder to track the user location.

Last, if we consider the data center scenario, a hypervisor might request local MAC addresses to be assigned to virtual machines. As in the previous scenarios, the hypervisor might select the preferred SLAP quadrant using information provided by the cloud management system (CMS) or virtualization infrastructure manager (VIM) running on top of the hypervisor. This information might include, but is not limited to:

- Migratable VM. If the function implemented by the VM is subject to be moved to another physical server or not. This has an impact on the preference for the SLAP quadrant, as some quadrants are better suited (e.g., ELI/SAI) for supporting migration in a large data center.

- VM connectivity characteristics , e.g.,: standalone, part of a pool, part of a service graph/chain. If the connectivity characteristics of the VM are known, this can be used by the hypervisor to select the best SLAP quadrant.

4. DHCPv6 Extensions

4.1. Address Assignment from the Preferred SLAP Quadrant Indicated by the Client

Next, we describe the protocol operations for a client to select a preferred SLAP quadrant using the DHCPv6 signaling procedures described in [I-D.ietf-dhc-mac-assign]. The signaling flow is shown in Figure 3.

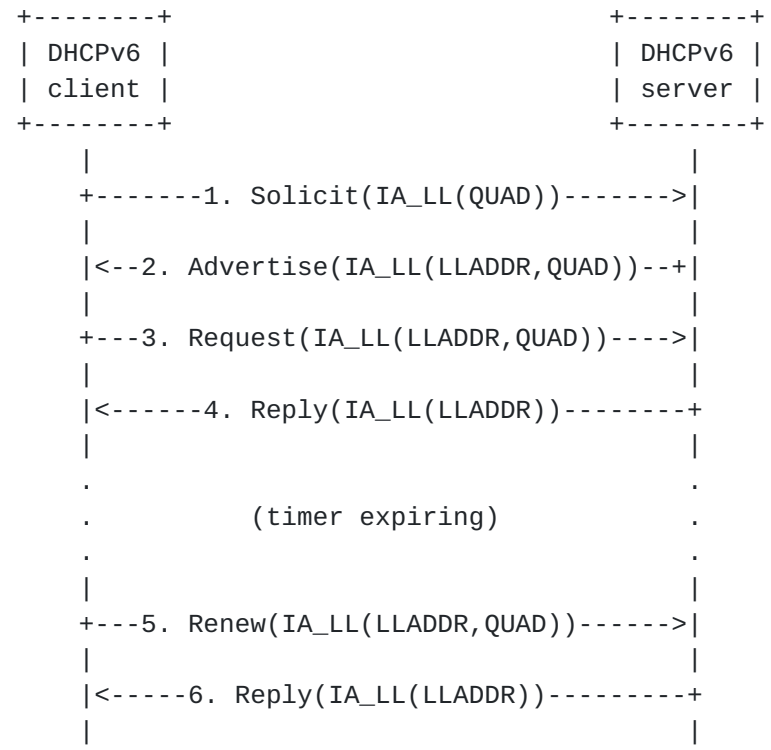

Figure 3: DHCPv6 signaling flow (client-server)

1. Link-layer addresses (i.e., MAC 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. In order to indicate the preferred SLAP quadrant(s), the IA_LL option includes the new OPTION_SLAP_QUAD option(s) in the IA-LL-option field (with the LLAADR option).

2. 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 sends back an Advertise message with an IA_LL option containing an LLADDR option that specifies the addresses being offered. If the server supports the new QUAD IA-LL-option, and manages a block of addresses

belonging to the requested quadrant(s), the addresses being offered MUST belong to the requested quadrant(s). If the server does not have a configured address pool matching the client's request, it MUST return the IA_LL option containing a Status Code option with status set to NoQuadAvail (IANA-2). If the client specified more than one SLAP quadrant, the server MUST only return a NoQuadAvail status code if the address pool configured at the server does not match any of the specified SLAP quadrants. If the server has a configured address pool of the correct quadrant, but no available addresses, it MUST return the IA_LL option containing a Status Code option with status set to NoAddrsAvail.

3. 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. It includes the preferred SLAP quadrant(s) in the new QUAD IA-LL-option.

4. 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. Note that 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 (following standard DHCPv6 procedures).

5. When the assigned addresses are about to expire, the client sends a Renew message. It includes the preferred SLAP quadrant(s) in the new QUAD IA-LL-option, so in case the server is unable to extend the lifetime on the existing address(es), the preferred quadrants are known for the allocation of any "new" addresses.

6. The server responds with a Reply message, including an LLADDR option with extended lifetime.

The client SHOULD check if the received MAC address comes from one of the requested quadrants. If not, SHOULD NOT configure the obtained address. It MAY repeat the process selecting a different DHCP server.

4.2. Address Assignment from the SLAP Quadrant Indicated by the Relay

Next, we describe the protocol operations for a relay to select a preferred SLAP quadrant using the DHCPv6 signaling procedures described in [I-D.ietf-dhc-mac-assign]. This is useful when a DHCPv6 server is operating over a large infrastructure split in different network regions, where each region might have different requirements. The signaling flow is shown in Figure 4.

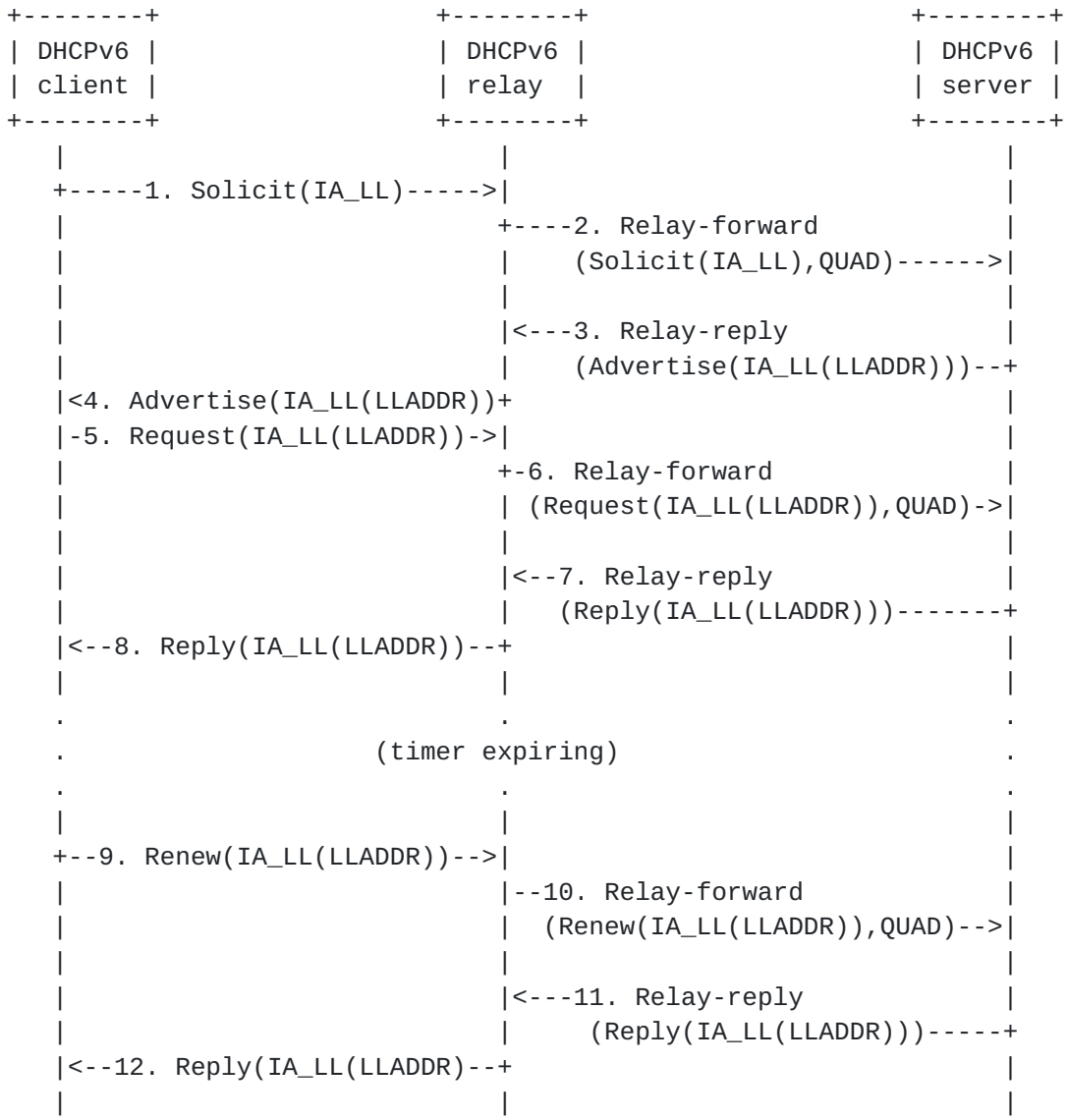

Figure 4: DHCPv6 signaling flow (client-relay-server)

1. Link-layer addresses (i.e., MAC 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.

2. The DHCP relay receives the Solicit message and encapsulates it in a Relay-forward message. The relay, based on local knowledge and policies, includes in the Relay-forward message the QUAD option with the preferred quadrant. The relay might know which quadrant to request based on local configuration (e.g., the served network contains IoT devices only, thus requiring ELI/SAI) or other means. Note that if a client sends multiple instances of the IA_LLoption in the same message, the DHCP relay MUST ONLY add a singleinstance of the QUAD option. The server SHOULD apply thecontents of the relay's supplied QUAD option for all of theclient's IA_LLs, unless configured to do otherwise.

3. The server, upon receiving the forwarded Solicit message including a IA_LL option, inspects its content and decide may offer an address or addresses for each LLADDR option according to its policy. The server sends back an Advertise message with an IA_LL option containing an LLADDR option that specifies the addresses being offered. This message is sent to the Relay in a Relay-reply message. If the server supports the semantics of the preferred quadrant included in the QUAD option, and manages a block of addresses belonging to the requested quadrant(s), then the addresses being offered SHOULD belong to the requested quadrant(s). Note that if the client is sending multiple instances of the IA_LL in the same message, a single QUAD option is supported, applying to all of the IA_LLs in the client's message.

4. The relay sends the received Advertise message to the client.

5. 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.

6. The relay forwards the received Request in a Relay-forward message. It adds in the Relay-forward a QUAD IA-LL-option with the preferred quadrant.

7. Upon reception of the forwarded 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, in a Relay-reply back to the relay.

8. Upon receiving a Reply message, the client parses the IA_LL container option and may start using all provided addresses.

9. When the assigned addresses are about to expire, the client sends a Renew message.

10. This message is forwarded by the Relay in a Relay-forward message, including a QUAD IA-LL-option with the preferred quadrant.

11. The server responds with a Reply message, including an LLADDR option with extended lifetime. This message is sent in a Relay-reply message.

12. The relay sends the Reply message back to the client.

If a client provides a QUAD IA-LL-option and the relay agent is still configured to add its preference value, the server SHOULD follow what is administratively configured in a QUAD_RELAY_PREF internal variable. If the value is set to 1, then the realy provided preference overrides what the client has provided, while if the variable is set to 0, then the client's provided preference is considered. It is RECOMMENDED that QUAD_RELAY_PREF is set to 1 at the server.

The client SHOULD check if the received MAC address belongs to a quadrant it is willing to use/configure. If not, SHOULD NOT configure the obtained address.

5. DHCPv6 Options Definitions

5.1. Quad (IA-LL) option

The QUAD option is used to specify the preferences for the selected quadrants within an IA_LL. The option must either be encapsulated in the IA-LL-options field of an IA_LL option or in a Relay-forward message.

The format of the QUAD option is:

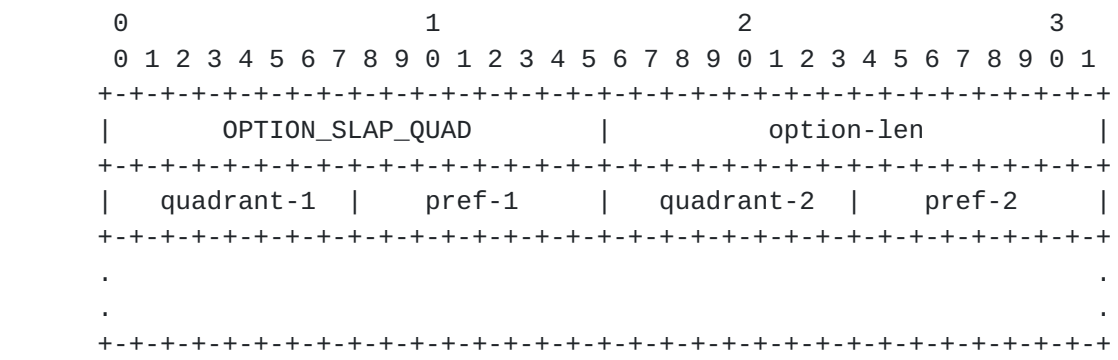

Figure 5: Quad Option Format

option-code OPTION_SLAP_QUAD (IANA-1).

option-len 2 * number of included (quadrant, preference).

quadrant-n Identifier of the quadrant (0: AAI, 1: ELI, 2: Reserved, 3: SAI). Each quadrant MUST only appear once at most in the option.

pref-n Preference associated to quadrant-n. A higher value means a more preferred quadrant.

A quadrant value MUST only appear once at most in the option. If an option includes more than one occurence of the same quadrant value, only the first occurence is processed and the rest MUST be ignored by the server.

If a same preference value is used for more than one quadrant, it is left to the server implementation which quadrant should be preferred (if the server can assign addresses from all of some of the quadrants with the same assigned preference).

If the client or relay agent provide the OPTION_SLAP_QUAD, the server uses the quadrant-n/pref-n values to order the selection of the quadrants. If the server can provide an assignment from one of the specified quadrants, it should proceed with the assignment. If the server cannot provide an assignment from one of the specified quadrant-n fields, it MUST NOT assign any addresses and return a status of NoQuadAvail (IANA-2) in the IA_LL Option.

There is no requirement that the client or relay agent order the quadrant/pref values in any specific order; hence servers MUST NOT assume that quadrant-1/pref-1 have the highest preference (except if there is only 1 set of values).

Client and relay agent MUST NOT place the same quadrant value into the option more than once; however servers should be capable of dealing with this by using the more preferred instance (as it requires no special handling).

6. IANA Considerations

IANA is requested to assign the QUAD (IANA-1) option code from the DHCPv6 "Option Codes" registry maintained at http://www.iana.org/assignments/dhcpv6-parameters and use the following data when adding the option to the registry:

Value: IANA-1
Description: OPTION_SLAP_QUAD
Client ORO: No
Singleton Option: No
Reference: this document

IANA is requested to assign the NoQuadAvail (IANA-2) code from the DHCPv6 "Status Codes" registry maintained athttp://www.iana.org/assignments/dhcpv6-parameters and use the following data when adding the option to the registry:

Value: IANA-2
Description: NoQuadAvail
Reference: this document

7. Security Considerations

See [RFC8415] for the DHCPv6 security considerations. See [RFC8200] for the IPv6 security considerations.

See also [I-D.ietf-dhc-mac-assign] for security considerations regarding link-layer address assignments using DHCP.

8. Acknowledgments

The authors would like to thank Bernie Volz for his very valuable comments on this document. We also want to thank Ian Farrer and Tomek Mrugalski for their very detailed and helpful reviews.

The work in this draft will be further developed and explored under the framework of the H2020 5Growth (Grant 856709) and 5G-DIVE projects (Grant 859881).

9. References

9.1. 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>.

[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>.

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

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

9.2. Informative References

[I-D.ietf-dhc-mac-assign] Volz, B., Mrugalski, T., and C. Bernardos, "Link-Layer Addresses Assignment Mechanism for DHCPv6", draft-ietf-dhc-mac-assign-03 (work in progress), January 2020.

[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".

Authors' Addresses

Carlos J. Bernardos
Universidad Carlos III de Madrid
Av. Universidad, 30
Leganes, Madrid 28911
Spain

Phone: +34 91624 6236
Email: cjbc@it.uc3m.es
URI: http://www.it.uc3m.es/cjbc/

Alain Mourad
InterDigital Europe

Email: Alain.Mourad@InterDigital.com
URI: http://www.InterDigital.com/