Stream: Internet Engineering Task Force (IETF)
RFC: 9686
Category: Standards Track
Published: December 2024
ISSN: 2070-1721

Authors:
W. Kumari *Google, LLC*
S. Krishnan *Cisco Systems, Inc.*
R. Asati *Independent*
L. Colitti *Google, LLC*
J. Linkova *Google, LLC*
S. Jiang *BUPT*

RFC 9686
Registering Self-Generated IPv6 Addresses Using DHCPv6

Abstract

This document defines a method to inform a DHCPv6 server that a device has one or more self-generated or statically configured addresses.

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/rfc9686.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

It is very common operational practice, especially in enterprise networks, to use IPv4 DHCP logs for troubleshooting or forensics purposes. An example of this includes a help desk dealing with a ticket such as "The CEO's laptop cannot connect to the printer"; if the Media Access Control (MAC) address of the printer is known (for example, from an inventory system), the printer's IPv4 address can be retrieved from the DHCP log or lease table and the printer can be pinged to determine if it is reachable. Another common example is a security operations team discovering suspicious events in outbound firewall logs and then consulting DHCP logs to determine which employee's laptop had that IPv4 address at that time so that they can quarantine it and remove the malware.

This operational practice relies on the DHCP server knowing the IP address assignments. This works quite well for IPv4 addresses, as most addresses are either assigned by DHCP [RFC2131] or statically configured by the network operator. For IPv6, however, this practice is much harder to implement, as devices often self-configure IPv6 addresses via Stateless Address Autoconfiguration (SLAAC) [RFC4862].

This document provides a mechanism for a device to inform the DHCPv6 server that the device has a self-configured IPv6 address (or has a statically configured address), and thus provides parity with IPv4 by making DHCPv6 infrastructure aware of self-assigned IPv6 addresses.

2. Conventions and Definitions

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. Registration Mechanism Overview

The DHCPv6 protocol is used as the address registration protocol and a DHCPv6 server performs the role of an address registration server. This document introduces a new Address Registration (OPTION_ADDR_REG_ENABLE) option, which indicates that the server supports the registration mechanism. Before registering any addresses, the client **MUST** determine whether the network supports address registration. It can do this by including the Address Registration option code in the Option Request option (see Section 21.7 of [RFC8415]) of the Information-Request, Solicit, Request, Renew, or Rebind messages it sends to the server as part of the regular stateless or stateful DHCPv6 configuration process. If the server supports address registration, it includes an Address Registration option in its Advertise or Reply messages. To avoid undesired multicast traffic, if the DHCPv6 infrastructure does not support (or is not willing to receive) any address registration information, the client **MUST NOT** register any addresses using the mechanism in this specification. Otherwise, the client registers addresses as described below.

After successfully assigning a self-generated or statically configured valid IPv6 address [RFC4862] on one of its interfaces, a client implementing this specification multicasts an ADDR-REG-INFORM message (see Section 4.2) in order to inform the DHCPv6 server that this self-generated address is in use. Each ADDR-REG-INFORM message contains a DHCPv6 Identity Association (IA) Address option [RFC8415] to specify the address being registered.

The address registration mechanism overview is shown in Figure 1.

```
  +--------+        +------------------+       +---------------+
  | CLIENT |        | FIRST-HOP ROUTER |       | DHCPv6 SERVER |
  +--------+        +---------+--------+       +-------+-------+
      |      SLAAC            |                        |
      |<--------------------> |                        |
      |                       |                        |
      |                                                |
      |  src: link-local address                       |
      | -------------------------------------------->  |
      |    INFORMATION-REQUEST or SOLICIT/...          |
      |       - OPTION REQUEST OPTION                  |
      |          -- OPTION_ADDR_REG_ENABLE             |
      |                                                |
      |    ...                                         |
      |                                                |
      |                                                |
      |<---------------------------------------------  |
      |     REPLY or ADVERTISE MESSAGE                 |
      |       - OPTION_ADDR_REG_ENABLE                 |
      |                                                |
      |                                                |
      |  src: address being registered                 |
      | -------------------------------------------->  |
      |    ADDR-REG-INFORM MESSAGE                     |Register/
      |                                                |log addresses
      |                                                |
      |                                                |
      | <--------------------------------------------  |
      |        ADDR-REG-REPLY MESSAGE                  |
      |                                                |
```

Figure 1: Address Registration Procedure Overview

4. DHCPv6 Address Registration Procedure

4.1. DHCPv6 Address Registration Option

The Address Registration option (OPTION_ADDR_REG_ENABLE) indicates that the server supports the mechanism described in this document. The format of the Address Registration option is described as follows:

```
 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-code          |           option-len          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: DHCPv6 Address Registration Option

option-code: OPTION_ADDR_REG_ENABLE (148)

option-len: 0

If a client has the address registration mechanism enabled, it **MUST** include this option in all Option Request options that it sends.

A server that is configured to support the address registration mechanism **MUST** include this option in Advertise and Reply messages if the client message it is replying to contained this option in the Option Request option.

4.2. DHCPv6 Address Registration Request Message

The DHCPv6 client sends an ADDR-REG-INFORM message to inform that an IPv6 address is assigned to the client's interface. The format of the ADDR-REG-INFORM message is described as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    msg-type   |               transaction-id                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                            options                            .
.                           (variable)                          .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: DHCPv6 ADDR-REG-INFORM Message

msg-type: Identifies the DHCPv6 message type; set to ADDR-REG-INFORM (36).

transaction-id: The transaction ID for this message exchange.

options: The options carried in this message.

The client **MUST** generate a transaction ID as described in [RFC8415] and insert this value in the transaction-id field.

The client **MUST** include the Client Identifier option [RFC8415] in the ADDR-REG-INFORM message.

The ADDR-REG-INFORM message **MUST NOT** contain the Server Identifier option and **MUST** contain exactly one IA Address option containing the address being registered. The valid-lifetime and preferred-lifetime fields in the option **MUST** match the current Valid Lifetime and Preferred Lifetime of the address being registered.

The ADDR-REG-INFORM message is dedicated for clients to initiate an address registration request toward an address registration server. Consequently, clients **MUST NOT** put any Option Request option(s) in the ADDR-REG-INFORM message. Clients **MAY** include other options, such as the Client FQDN option [RFC4704].

The client sends the DHCPv6 ADDR-REG-INFORM message to the All_DHCP_Relay_Agents_and_Servers multicast address (ff02::1:2). The client **MUST** send separate messages for each address being registered.

Unlike other types of messages, which are sent from the link-local address of the client, the ADDR-REG-INFORM message **MUST** be sent from the address being registered. This is primarily for "fate sharing" purposes; for example, if the network implements some form of Layer 2 security to prevent a client from spoofing other clients' MAC addresses, this prevents an attacker from spoofing ADDR-REG-INFORM messages.

On clients with multiple interfaces, the client **MUST** only send the packet on the network interface that has the address being registered, even if it has multiple interfaces with different addresses. If the same address is configured on multiple interfaces, then the client **MUST** send the ADDR-REG-INFORM message each time the address is configured on an interface that did not previously have it and refresh each registration independently from the others.

The client **MUST** only send the ADDR-REG-INFORM message for valid addresses [RFC4862] of global scope [RFC4007]. This includes Unique Local Addresses (ULAs), which are defined in [RFC4193] to have global scope. This also includes statically assigned addresses of global scope (such addresses are considered to be valid indefinitely). The client **MUST NOT** send the ADDR-REG-INFORM message for addresses configured by DHCPv6.

The client **SHOULD NOT** send the ADDR-REG-INFORM message unless it has received a Router Advertisement (RA) message with either the M or O flags set to 1.

Clients **MUST** discard any received ADDR-REG-INFORM messages.

4.2.1. Server Message Processing

Servers **MUST** discard any ADDR-REG-INFORM messages that meet any of the following conditions:

- the message does not include a Client Identifier option;
- the message includes a Server Identifier option;

- the message does not include the IA Address option, or the IP address in the IA Address option does not match the source address of the original ADDR-REG-INFORM message sent by the client. The source address of the original message is the source IP address of the packet if it is not relayed or is the peer-address field of the innermost Relay-forward message if it is relayed; or
- the message includes an Option Request option.

If the message is not discarded, the address registration server **SHOULD** verify that the address being registered is "appropriate to the link" as defined by [RFC8415] or within a prefix delegated to the client via DHCPv6 for Prefix Delegation (DHCPv6-PD) (see Section 6.3 of [RFC8415]). If the address being registered fails this verification, the server **MUST** drop the message and **SHOULD** log this fact. If the message passes the verification, the server:

- **MUST** log the address registration information (as is done normally for clients to which it has assigned an address), unless it is configured not to do so. The server **SHOULD** log the client DHCP Unique Identifier (DUID) and the link-layer address, if available. The server **MAY** log any other information.
- **SHOULD** register a binding between the provided Client Identifier and IPv6 address in its database, if no binding exists. The lifetime of the binding is equal to the Valid Lifetime of the address reported by the client. If there is already a binding between the registered address and the same client, the server **MUST** update its lifetime. If there is already a binding between the registered address and another client, the server **SHOULD** log the fact and update the binding.
- **SHOULD** mark the address as unavailable for use and not include it in future Advertise messages.
- **MUST** send back an ADDR-REG-REPLY message to ensure the client does not retransmit.

If a client is multihomed (i.e., connected to multiple administrative domains, each operating its own DHCPv6 infrastructure), the requirement to verify that the registered address is appropriate for the link or belongs to a delegated prefix ensures that each DHCPv6 server only registers bindings for addresses from the given administrative domain.

As mentioned in Section 4.2, although a client "**MUST NOT** send the ADDR-REG-INFORM message for addresses configured by DHCPv6", if a server does receive such a message, it **SHOULD** log and discard it.

DHCPv6 relay agents and switches that relay address registration messages directly from clients **MUST** include the client's link-layer address in the relayed message using the Client Link-Layer Address option [RFC6939] if they would do so for other DHCPv6 client messages such as Solicit, Request, and Rebind.

4.3. DHCPv6 Address Registration Acknowledgement

The server **MUST** acknowledge receipt of a valid ADDR-REG-INFORM message by sending back an ADDR-REG-REPLY message. The format of the ADDR-REG-REPLY message is described as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    msg-type   |               transaction-id                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                            options                            .
.                           (variable)                          .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: DHCPv6 ADDR-REG-REPLY Message

msg-type: Identifies the DHCPv6 message type; set to ADDR-REG-REPLY (37).

transaction-id: The transaction ID for this message exchange.

options: The options carried in this message.

If the ADDR-REG-INFORM message that the server is replying to was not relayed, then the IPv6 destination address of the message **MUST** be the address being registered. If the ADDR-REG-INFORM message was relayed, then the server **MUST** construct the Relay-reply message as specified in Section 19.3 of [RFC8415].

The server **MUST** copy the transaction-id from the ADDR-REG-INFORM message to the transaction-id field of the ADDR-REG-REPLY.

The ADDR-REG-REPLY message **MUST** contain an IA Address option for the address being registered. The option **MUST** be identical to the one in the ADDR-REG-INFORM message that the server is replying to.

Servers **MUST** ignore any received ADDR-REG-REPLY messages.

Clients **MUST** discard any ADDR-REG-REPLY messages that meet any of the following conditions:

- the IPv6 destination address does not match the address being registered;
- the IA Address option does not match the address being registered;
- the address being registered is not assigned to the interface receiving the message; or
- the transaction-id does not match the transaction-id the client used in the corresponding ADDR-REG-INFORM message.

The ADDR-REG-REPLY message only indicates that the ADDR-REG-INFORM message has been received and that the client should not retransmit it. The ADDR-REG-REPLY message **MUST NOT** be considered to be any indication of the address validity and **MUST NOT** be required for the address to be usable. DHCPv6 relays, or other devices that snoop ADDR-REG-REPLY messages, **MUST NOT** add or alter any forwarding or security state based on the ADDR-REG-REPLY message.

4.4. Signaling Address Registration Support

To avoid undesired multicast traffic, the client **MUST NOT** register addresses using this mechanism unless the DHCPv6 infrastructure supports address registration. The client can discover this by including the OPTION_ADDR_REG_ENABLE option in the Option Request options that it sends. If the client receives and processes an Advertise or Reply message with the OPTION_ADDR_REG_ENABLE option, it concludes that the DHCPv6 infrastructure supports address registration. When the client detects address registration support, it **MUST** start the registration process (unless configured not to do so) and **MUST** immediately register any addresses that are already in use. Once the client starts the registration process, it **MUST NOT** stop registering addresses until it disconnects from the link, even if subsequent Advertise or Reply messages do not contain the OPTION_ADDR_REG_ENABLE option.

The client **MUST** discover whether the DHCPv6 infrastructure supports address registration every time it connects to a network or when it detects it has moved to a new link, without utilizing any prior knowledge about address registration support on that network or link. This client behavior allows networks to progressively roll out support for the Address Registration option across the DHCPv6 infrastructure without causing clients to frequently stop and restart address registration if some of the network's DHCPv6 servers support it and some do not.

A client with multiple interfaces **MUST** discover address registration support for each interface independently. The client **MUST NOT** send address registration messages on a given interface unless the client has discovered that the interface is connected to a network that supports address registration.

4.5. Retransmission

To reduce the effects of packet loss on registration, the client **MUST** retransmit the registration message. Retransmissions **SHOULD** follow the standard retransmission logic specified by Section 15 of [RFC8415] with the following default parameters for the initial retransmission time (IRT) and maximum retransmission count (MRC):

- IRT 1 sec
- MRC 3

The client **SHOULD** allow these parameters to be configured by the administrator.

To comply with Section 16.1 of [RFC8415], the client **MUST** leave the transaction ID unchanged in retransmissions of an ADDR-REG-INFORM message. When the client retransmits the registration message, the lifetimes in the packet **MUST** be updated so that they match the current lifetimes of the address.

If an ADDR-REG-REPLY message is received for the address being registered, the client **MUST** stop retransmission.

4.6. Registration Expiry and Refresh

The client **MUST** refresh registrations to ensure that the server is always aware of which addresses are still valid. The client **SHOULD** perform refreshes as described below.

4.6.1. SLAAC Addresses

For an address configured using SLAAC, a function AddrRegRefreshInterval(address) is defined as 80% of the address's current Valid Lifetime. When calculating this value, the client applies a multiplier of AddrRegDesyncMultiplier to avoid synchronization with other clients, which could cause a large number of registration messages to reach the server at the same time. AddrRegDesyncMultiplier is a random value uniformly distributed between 0.9 and 1.1 (inclusive) and is chosen by the client when it starts the registration process, to ensure that refreshes for addresses with the same lifetime are coalesced (see below).

Whenever the client registers or refreshes an address, it calculates a NextAddrRegRefreshTime for that address as AddrRegRefreshInterval seconds in the future but does not schedule any refreshes.

Whenever the network changes the Valid Lifetime of an existing address by more than 1%, for example, by sending a Prefix Information Option (PIO) [RFC4861] with a new Valid Lifetime, the client calculates a new AddrRegRefreshInterval. The client schedules a refresh for min(now + AddrRegRefreshInterval, NextAddrRegRefreshTime). If the refresh would be scheduled in the past, then the refresh occurs immediately.

Justification: This algorithm ensures that refreshes are not sent too frequently while ensuring that the server never believes that the address has expired when it has not. Specifically, after every registration:

- If the network never changes the lifetime of an address (e.g., if no further PIOs are received, or if all PIO lifetimes decrease in step with the passage of time), then no refreshes occur. Refreshes are not necessary, because the address expires at the time the server expects it to expire.
- Any time the network changes the lifetime of an address (i.e., changes the time at which the address will expire), the client ensures that a refresh is scheduled, so that server will be informed of the new expiry.
- Because AddrRegDesyncMultiplier is at most 1.1, the refresh never occurs later than a point 88% between the time when the address was registered and the time when the address will expire. This allows the client to retransmit the registration for up to 12% of the original interval before it expires. This may not be possible if the network sends a Router Advertisement (RA) [RFC4861] very close to the time when the address would have expired. In this case, the client refreshes immediately, which is the best it can do.
- The 1% tolerance ensures that the client will not refresh or reschedule refreshes if the Valid Lifetime experiences minor changes due to transmission delays or clock skew between the client and the router(s) sending the RA.

- AddrRegRefreshCoalesce (Section 4.6.3) allows battery-powered clients to wake up less often. In particular, it allows the client to coalesce refreshes for multiple addresses formed from the same prefix, such as the stable and privacy addresses. Higher values will result in fewer wakeups but may result in more network traffic, because if a refresh is sent early, then the next RA received will cause the client to immediately send a refresh message.
- In typical networks, the lifetimes in periodic RAs either contain constant values or values that decrease over time to match another lifetime, such as the lifetime of a prefix delegated to the network. In both these cases, this algorithm will refresh on the order of once per address lifetime, which is similar to the number of refreshes that are necessary using stateful DHCPv6.
- Because refreshes occur at least once per address lifetime, the network administrator can control the address refresh frequency by appropriately setting the Valid Lifetime in the PIO.

4.6.2. Statically Assigned Addresses

A statically assigned address has an infinite Valid Lifetime that is not affected by RAs. Therefore, whenever the client registers or refreshes a statically assigned address, the next refresh is scheduled for StaticAddrRegRefreshInterval seconds in the future. The default value of StaticAddrRegRefreshInterval is 4 hours. This ensures static addresses are still refreshed periodically, but refreshes for static addresses do not cause excessive multicast traffic. The StaticAddrRegRefreshInterval interval **SHOULD** be configurable.

4.6.3. Transmitting Refreshes

When a refresh is performed, the client **MAY** refresh all addresses assigned to the interface that are scheduled to be refreshed within the next AddrRegRefreshCoalesce seconds. The value of AddrRegRefreshCoalesce is implementation dependent, and a suggested default is 60 seconds.

Registration refresh packets **MUST** be retransmitted using the same logic as used for initial registrations (see Section 4.5).

The client **MUST** generate a new transaction ID when refreshing the registration.

When a Client-Identifier-to-IPv6-address binding expires, the server **MUST** remove it and consider the address as available for use.

The client **MAY** choose to notify the server when an address is no longer being used (e.g., if the client is disconnecting from the network, the address lifetime expired, or the address is being removed from the interface). To indicate that the address is not being used anymore, the client **MUST** set the preferred-lifetime and valid-lifetime fields of the IA Address option in the ADDR-REG-INFORM message to zero. If the server receives a message with a valid-lifetime of zero, it **MUST** act as if the address has expired.

5. Client Configuration

DHCP clients **SHOULD** allow the administrator to disable sending ADDR-REG-INFORM messages. Sending the messages **SHOULD** be enabled by default.

6. Security Considerations

An attacker may attempt to register a large number of addresses in quick succession in order to overwhelm the address registration server and/or fill up log files. Similar attack vectors exist today, e.g., an attacker can DoS the server with messages containing spoofed DHCP Unique Identifiers (DUIDs) [RFC8415].

If a network is using First-Come, First-Served Source Address Validation Improvement (FCFS SAVI) [RFC6620], then the DHCPv6 server can trust that the ADDR-REG-INFORM message was sent by the legitimate holder of the address. This prevents a client from registering an address configured on another client.

One of the use cases for the mechanism described in this document is to identify sources of malicious traffic after the fact. Note, however, that as the device itself is responsible for informing the DHCPv6 server that it is using an address, a malicious or compromised device can simply choose to not send the ADDR-REG-INFORM message. This is an informational, optional mechanism and is designed to aid in troubleshooting and forensics. On its own, it is not intended to be a strong security access mechanism. In particular, the ADDR-REG-INFORM message **MUST NOT** be used for authentication and authorization purposes, because in addition to the reasons above, the packets containing the message may be dropped.

7. Privacy Considerations

If the network doesn't have Multicast Listener Discovery (MLD) snooping enabled, then IPv6 link-local multicast traffic is effectively transmitted as broadcast. In such networks, an on-link attacker listening to DHCPv6 messages might obtain information about IPv6 addresses assigned to the client. As ADDR-REG-INFORM messages contain unique identifiers such as the client's DUID, the attacker may be able to track addresses being registered and map them to the same client, even if the client uses randomized MAC addresses. This privacy consideration is not specific to the proposed mechanism. Section 4.3 of [RFC7844] discusses using the DUID for device tracking in DHCPv6 environments and provides mitigation recommendations.

In general, hiding information about the specific IPv6 address from on-link observers should not be considered a security measure, as such information is usually disclosed via Duplicate Address Detection [RFC4862] to all nodes anyway, if MLD snooping is not enabled.

If MLD snooping is enabled, an attacker might be able to join the All_DHCP_Relay_Agents_and_Servers multicast address (ff02::1:2) group to listen for address registration messages. However, the same result can be achieved by joining the All Routers Address (ff02::2) group and listen to gratuitous neighbor advertisement messages [RFC9131]. It should be noted that this particular scenario shares the fate with DHCPv6 address assignment: if an attacker can join the All_DHCP_Relay_Agents_and_Servers multicast group, they would be able to monitor all DHCPv6 messages sent from the client to DHCPv6 servers and relays and therefore obtain the information about addresses being assigned via DHCPv6. Layer 2 isolation allows mitigating this threat by blocking on-link peer-to-peer communication between nodes.

8. IANA Considerations

This document introduces the following entities, which have been allocated in the "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)" registry group defined at <http://www.iana.org/assignments/dhcpv6-parameters>. These include:

- One new DHCPv6 option, described in Section 4.1, which has been allocated in the "Option Codes" registry:

  Value: 148
  Description: OPTION_ADDR_REG_ENABLE
  Client ORO: Yes
  Singleton Option: Yes
  Reference: RFC 9686

- Two new DHCPv6 messages, which have been allocated in the "Message Types" registry (for more information, see Sections 4.2 and 4.3, respectively, for each DHCPv6 message):

  Value: 36
  Description: ADDR-REG-INFORM
  Reference: RFC 9686
  Value: 37
  Description: ADDR-REG-REPLY
  Reference: RFC 9686

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>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <https://www.rfc-editor.org/info/rfc2131>.

[RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", RFC 4007, DOI 10.17487/RFC4007, March 2005, <https://www.rfc-editor.org/info/rfc4007>.

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

[RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", RFC 4704, DOI 10.17487/RFC4704, October 2006, <https://www.rfc-editor.org/info/rfc4704>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC6939] Halwasia, G., Bhandari, S., and W. Dec, "Client Link-Layer Address Option in DHCPv6", RFC 6939, DOI 10.17487/RFC6939, May 2013, <https://www.rfc-editor.org/info/rfc6939>.

[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>.

[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>.

[RFC9131] Linkova, J., "Gratuitous Neighbor Discovery: Creating Neighbor Cache Entries on First-Hop Routers", RFC 9131, DOI 10.17487/RFC9131, October 2021, <https://www.rfc-editor.org/info/rfc9131>.

9.2. Informative References

[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>.

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

Acknowledgements

Many thanks to Bernie Volz for the significant review and feedback, as well as Hermin Anggawijaya, Carlos Jesus Bernardos, Brian Carpenter, Stuart Cheshire, Roman Danyliw, Alan DeKok, James Guichard, James Guichard, Erik Kline, Mallory Knodel, Murray Kucherawy, David Lamparter, Ted Lemon, Eric Levy-Abegnoli, Aditi Patange, Jim Reid, Michael Richardson, Patrick Rohr, John Scudder, Mark Smith, Gunter Van de Velde, Eric Vyncke, Timothy Winters, and Peter Yee for their feedback, comments, and guidance. We apologize if we inadvertently forgot to acknowledge anyone's contributions.

Contributors

Gang Chen
China Mobile
53A, Xibianmennei Ave.
Xuanwu District
Beijing
China
Email: phdgang@gmail.com

Authors' Addresses

Warren Kumari
Google, LLC
Email: warren@kumari.net

Suresh Krishnan
Cisco Systems, Inc.
Email: suresh.krishnan@gmail.com

Rajiv Asati
Independent
Email: rajiv.asati@gmail.com

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

Jen Linkova
Google, LLC
1 Darling Island Rd
Pyrmont 2009
Australia
Email: furry13@gmail.com

Sheng Jiang
Beijing University of Posts and Telecommunications
No. 10 Xitucheng Road
Beijing
Haidian District, 100083
China
Email: shengjiang@bupt.edu.cn