Stream:	Internet Engineering Task Force (IETF)				
RFC:	9915				
STD:	102				
Obsoletes:	8415				
Category:	Standards Track				
Published:	January 2026				
ISSN:	2070-1721				
Authors:	T. Mrugalski	B. Volz	M. Richardson	S. Jiang	T. Winters
	ISC	*Individual Contributor*	*SSW*	*BUPT*	*QA Cafe*

RFC 9915
Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

Abstract

This document specifies the Dynamic Host Configuration Protocol for IPv6 (DHCPv6), an extensible mechanism for configuring nodes with network configuration parameters, IP addresses, and prefixes. Parameters can be provided statelessly or in combination with stateful assignment of one or more IPv6 addresses and/or IPv6 prefixes. DHCPv6 can operate either in place of or in addition to stateless address autoconfiguration (SLAAC).

This document obsoletes RFC 8415. It incorporates verified errata and obsoletes the assignment of temporary addresses (the IA_TA option) and the server unicast capability (the Server Unicast option and UseMulticast status code).

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/rfc9915.

Copyright Notice

Table of Contents

1. Introduction

This document specifies DHCP for IPv6 (DHCPv6), a client/server protocol that provides managed configuration of devices. The basic operation of DHCPv6 provides configuration for clients connected to the same link as the server. Relay agent functionality is also defined for enabling communication between clients and servers that are not on the same link.

DHCPv6 can provide a device with addresses assigned by a DHCPv6 server and other configuration information; this data is carried in options. DHCPv6 can be extended through the definition of new options to carry configuration information not specified in this document.

DHCPv6 also supports a mechanism for automated delegation of IPv6 prefixes. Through this mechanism, a server can delegate prefixes to clients. Use of this mechanism is specified as part of [RFC7084] and by [TR-187]. Note that those documents use "requesting router" and "delegating router" where this document uses "client" and "server", respectively.

DHCP can also be used just to provide other configuration options (i.e., no addresses or prefixes). That implies that the server does not have to track any state; thus, this mode is called "stateless DHCPv6". Mechanisms necessary to support stateless DHCPv6 are much simpler than mechanisms needed to support stateful DHCPv6.

1.1. Relationship to Previous DHCPv6 Standards

[RFC8415] provided a unified, corrected, and cleaned-up definition of DHCPv6 that also covered all applicable errata filed against older RFCs at the time of its writing. It also obsoleted a small number of mechanisms: delayed authentication, lifetime, and timer hints sent by a client.

This document obsoletes [RFC8415]. It applies verified errata reports and obsoletes two features that have not been widely implemented - the assignment of temporary addresses using the IA_TA option and allowing clients to unicast some messages directly to the server if the server sent the Server Unicast option to a client in an early exchange. It also clarifies the UDP ports used by clients, servers, and relay agents (Section 7.2). See Appendix A for a list of differences from [RFC8415].

1.2. Topics Out of Scope

This document specifies DHCPv6 behavior. The server policy, such as what options to assign to which clients, which subnets or pools of resources to use, which clients' requests should be denied, etc. are out of scope for this document.

Server configuration, operation, and management are also out of scope. An approach to manage DHCPv6 relays and servers is specified in [RFC9243].

Merging DHCPv4 [RFC2131] and DHCPv6 configuration is out of scope for this document. [RFC4477] discusses some issues and possible strategies for running DHCPv4 and DHCPv6 services together. While [RFC4477] is a bit dated, it provides a good overview of the issues at hand. The consensus of the IETF at the time of writing is that DHCPv4 should be used rather than DHCPv6 when conveying IPv4 configuration information to nodes. For IPv6-only networks, [RFC7341] describes a transport mechanism to carry DHCPv4 messages using DHCPv6 for the dynamic provisioning of IPv4 address and configuration information.

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.

This document also makes use of internal conceptual variables to describe protocol behavior and external variables that an implementation must allow system administrators to change. The specific variable names, how their values change, and how their settings influence protocol behavior are provided to demonstrate protocol behavior. An implementation is not required to have them in the exact form described here, as long as its external behavior is consistent with that described in this document.

3. Background

In [RFC8415], the "Background" section contained background on IPv6 specifications of relevance to DHCPv6. That text has been removed from the current document; however, this section has been retained to keep the major section numbering consistent with [RFC8415]. Those interested can refer to [RFC8415] itself for more information on the topic.

4. Terminology

This section defines terminology specific to IPv6 and DHCP used in this document.

4.1. IPv6 Terminology

IPv6 terminology from [RFC8200], [RFC4291], and [RFC4862] relevant to this specification is included below.

address
: An IP-layer identifier for an interface or a set of interfaces.

GUA
: Global unicast address (see [RFC4291]).

host
: Any node that is not a router.

IP
: Internet Protocol Version 6 (IPv6). The terms "IPv4" and "IPv6" are used only in contexts where it is necessary to avoid ambiguity.

interface
: A node's attachment to a link.

link
: A communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IP. Examples are Ethernet (simple or bridged); Point-to-Point Protocol (PPP) and PPP over Ethernet (PPPoE) links; and Internet-layer (or higher) "tunnels", such as tunnels over IPv4 or IPv6 itself.

link-layer identifier
: A link-layer identifier for an interface -- for example, IEEE 802 addresses for Ethernet or Token Ring network interfaces.

link-local address
: An IPv6 address having a link-only scope, indicated by having the prefix (fe80::/10), that can be used to reach neighboring nodes attached to the same link. Every IPv6 interface on which DHCPv6 can reasonably be useful has a link-local address.

multicast address
: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.

neighbor
: A node attached to the same link.

node
: A device that implements IP.

packet
: An IP header plus payload.

prefix
: The initial bits of an address, or a set of IP addresses that share the same initial bits.

prefix length
: The number of bits in a prefix.

router
: A node that forwards IP packets not explicitly addressed to itself.

ULA
: Unique local address (see [RFC4193]).

unicast address
: An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.

4.2. DHCP Terminology

Terminology specific to DHCP can be found below.

appropriate to the link
: An address is "appropriate to the link" when the address is consistent with the DHCP server's knowledge of the network topology, prefix assignment, and address assignment policies.

binding
: A binding (or client binding) is a group of server data records containing the information the server has about the addresses or delegated prefixes in an Identity Association (IA) or configuration information explicitly assigned to the client. Configuration information that

has been returned to a client through a policy, such as the information returned to all clients on the same link, does not require a binding. A binding containing information about an IA is indexed by the tuple <DUID, IA-type, IAID> (where IA-type is the type of lease in the IA -- for example, address or delegated prefix). A binding containing configuration information for a client is indexed by <DUID>. See below for definitions of DUID, IA, and IAID.

configuration parameter
: An element of the configuration information set on the server and delivered to the client using DHCP. Such parameters may be used to carry information to be used by a node to configure its network subsystem and enable communication on a link or internetwork, for example.

container option
: An option that encapsulates other options (for example, the IA_NA option (see Section 21.4) may contain IA Address options (see Section 21.6)).

DHCP
: Dynamic Host Configuration Protocol for IPv6. The terms "DHCPv4" and "DHCPv6" are used only in contexts where it is necessary to avoid ambiguity.

DHCP client
: Also referred to as "client". A node that initiates requests on a link to obtain configuration parameters from one or more DHCP servers.

DHCP domain
: A set of links managed by DHCP and operated by a single administrative entity.

DHCP relay agent
: Also referred to as "relay agent". A node that acts as an intermediary to deliver DHCP messages between clients and servers. In certain configurations, there may be more than one relay agent between clients and servers, so a relay agent may send DHCP messages to another relay agent.

DHCP server
: This document condenses this term to "server". A node that responds to requests from clients. It may or may not be on the same link as the client(s).

DUID
: A DHCP Unique Identifier for a DHCP participant. Each DHCP client and server has exactly one DUID. See Section 11 for details of the ways in which a DUID may be constructed.

encapsulated option
: A DHCP option that is usually only contained in another option. For example, the IA Address option is contained in IA_NA options (see Section 21.6 and Section 21.4, respectively). See Section 9 of [RFC7227] for a more complete definition.

IA
: Identity Association: a collection of leases assigned to a client. Each IA has an associated IAID (see below). A client may have more than one IA assigned to it -- for example, one for each of its interfaces. Each IA holds one type of lease; for example, an identity association for non-

temporary addresses (IA_NA) holds addresses, and an identity association for prefix delegation (IA_PD) holds delegated prefixes. Throughout this document, "IA" is used to refer to an identity association without identifying the type of a lease in the IA. This document defines three IA types: IA_NA, IA_TA (obsoleted), and IA_PD. Another IA type (IA_LL) was defined in [RFC8947] and more may be defined.

IA option(s)
: In this document, one or more IA_NA, IA_TA (obsoleted), and/or IA_PD options. Another IA type (IA_LL) was defined in [RFC8947] and more may be defined.

IAID
: Identity Association Identifier: an identifier for an IA, chosen by the client. Each IA has an IAID, which is chosen to be unique among IAIDs for IAs of a specific type that belong to that client.

IA_NA
: Identity Association for Non-temporary Addresses: an IA that carries assigned addresses. See Section 21.4 for details on the IA_NA option.

IA_PD
: Identity Association for Prefix Delegation: an IA that carries delegated prefixes. See Section 21.21 for details on the IA_PD option.

IA_TA
: Identity Association for Temporary Addresses: an IA that carries temporary addresses (see [RFC8981]). This option is obsoleted by this document. See [RFC8415] for details.

lease
: A contract by which the server grants the use of an address or delegated prefix to the client for a specified period of time.

message
: A unit of data carried as the payload of a UDP datagram, exchanged among DHCP servers, relay agents, and clients.

Reconfigure key
: A key supplied to a client by a server. Used to provide security for Reconfigure messages (see Section 20.4 for use cases for the reconfigure key).

relaying
: A DHCP relay agent relays DHCP messages between DHCP participants.

retransmission
: Another attempt to send the same DHCP message by a client or server, as a result of not receiving a valid response to the previously sent messages. The retransmitted message is typically modified prior to sending, as required by the DHCP specifications. In particular, the client updates the value of the Elapsed Time option in the retransmitted message.

RKAP
: The Reconfiguration Key Authentication Protocol (see Section 20.4).

singleton option
: An option that is allowed to appear only once as a top-level option or at any encapsulation level. Most options are singletons.

T1
: The time interval after which the client is expected to contact the server that did the assignment to extend (renew) the lifetimes of the addresses assigned (via IA_NA option(s)) and/or prefixes delegated (via IA_PD option(s)) to the client. T1 is expressed as an absolute value in messages (in seconds), is conveyed within IA containers (currently the IA_NA and IA_PD options), and is interpreted as a time interval since the message's reception. The value stored in the T1 field in IA options is referred to as the T1 value. The actual time when the timer expires is referred to as the T1 time.

T2
: The time interval after which the client is expected to contact any available server to extend (rebind) the lifetimes of the addresses assigned (via IA_NA option(s)) and/or prefixes delegated (via IA_PD option(s)) to the client. T2 is expressed as an absolute value in messages (in seconds), is conveyed within IA containers (currently the IA_NA and IA_PD options), and is interpreted as a time interval since the message's reception. The value stored in the T2 field in IA options is referred to as the T2 value. The actual time when the timer expires is referred to as the T2 time.

top-level option
: An option conveyed in a DHCP message directly, i.e., not encapsulated in any other option, as described in Section 9 of [RFC7227].

transaction ID
: An opaque value used to match responses with replies initiated by either a client or a server.

5. Client/Server Exchanges

Clients and servers exchange DHCP messages using UDP (see [RFC0768] and [BCP145]). The client uses a link-local source address or addresses determined through other mechanisms for transmitting and receiving DHCP messages.

A DHCP client sends all messages using a reserved, link-scoped multicast destination address (All_DHCP_Relay_Agents_and_Servers - ff02::1:2) so that the client need not be configured with the address or addresses of DHCP servers.

To allow a DHCP client to send a message to a DHCP server that is not attached to the same link, a DHCP relay agent on the client's link will relay messages between the client and server. The operation of the relay agent is transparent to the client. The discussion of message exchanges in the remainder of this section will omit the description of the relaying of messages by relay agents.

5.1. Client/Server Exchanges Involving Two Messages

When a DHCP client does not need to have a DHCP server assign IP addresses or delegated prefixes to it, the client can obtain other configuration information such as a list of available DNS servers [RFC3646] or NTP servers [RFC5908] through a single message and reply exchange with a DHCP server. To obtain other configuration information, the client first sends an Information-request message to the All_DHCP_Relay_Agents_and_Servers multicast address. Servers respond with a Reply message containing the other configuration information for the client.

A client may also request the server to expedite address assignment and/or prefix delegation by using a two-message exchange instead of the normal four-message exchange as discussed in the next section. Expedited assignment can be requested by the client, and servers may or may not honor the request (see Sections 18.3.1 and 21.14 for more details and why servers may not honor this request). Clients may request this expedited service in environments where it is likely that there is only one server available on a link and no expectation that a second server would become available, or when completing the configuration process as quickly as possible is a priority.

To request the expedited two-message exchange, the client sends a Solicit message to the All_DHCP_Relay_Agents_and_Servers multicast address requesting the assignment of addresses and/or delegated prefixes and other configuration information. This message includes an indication (the Rapid Commit option; see Section 21.14) that the client is willing to accept an immediate Reply message from the server. The server that is willing to commit the assignment of addresses and/or delegated prefixes to the client immediately responds with a Reply message. The configuration information and the addresses and/or delegated prefixes in the Reply message are then immediately available for use by the client.

Each address or delegated prefix assigned to the client has associated preferred and valid lifetimes specified by the server. To request an extension of the lifetimes assigned to an address or delegated prefix, the client sends a Renew message to the server. The server sends a Reply message to the client with the new lifetimes, allowing the client to continue to use the address or delegated prefix without interruption. If the server is unable to extend the lifetime of an address or delegated prefix, it indicates this by returning the address or delegated prefix with lifetimes of 0. At the same time, the server may assign other addresses or delegated prefixes.

See Section 18 for descriptions of additional two-message exchanges between the client and server.

5.2. Client/Server Exchanges Involving Four Messages

To request the assignment of one or more addresses and/or delegated prefixes, a client first locates a DHCP server and then requests the assignment of addresses and/or delegated prefixes and other configuration information from the server. The client sends a Solicit message to the All_DHCP_Relay_Agents_and_Servers multicast address to find available DHCP servers. Any server that can meet the client's requirements responds with an Advertise message. The client

then chooses one of the servers and sends a Request message to the server asking for the lease of addresses and/or delegated prefixes and other configuration information. The server responds with a Reply message that contains the leased addresses, delegated prefixes, and configuration.

As described in the previous section, the client can request an extension of the lifetimes assigned to addresses or delegated prefixes (this is a two-message exchange).

5.3. Server/Client Exchanges

A server that has previously communicated with a client and negotiated for the client to listen for Reconfigure messages may send the client a Reconfigure message to initiate the client to update its configuration by sending an Information-request, Renew, or Rebind message. Reconfigure messages are authenticated as per Section 20.4. The client then performs the two-message exchange as described earlier. This can be used to expedite configuration changes to a client, such as the need to renumber a network (see [RFC6879] and [RFC9096]).

6. Operational Models

This section describes some of the current most common DHCP operational models. The described models are not mutually exclusive and are sometimes used together. For example, a device may start in stateful mode to obtain an address and, at a later time when an application is started, request additional parameters using stateless mode.

This document assumes that the DHCP servers and the client, communicating with the servers via a specific interface, belong to a single provisioning domain.

DHCP may be extended to support additional stateful services that may interact with one or more of the models described below. Such interaction should be considered and documented as part of any future protocol extension.

6.1. Stateless DHCP

Stateless DHCP can be used at any time, typically when a node requires some missing or expired configuration information that is available via DHCP.

This is the simplest and most basic operation for DHCP and requires a client (and a server) to support only two messages -- Information-request and Reply. Note that DHCP servers and relay agents typically also need to support the Relay-forward and Relay-reply messages to accommodate operation when clients and servers are not on the same link.

6.2. DHCP for Non-Temporary Address Assignment

This model of operation was the original motivation for DHCP. It is appropriate for situations where stateless address autoconfiguration alone is insufficient or impractical, e.g., because of network policy, additional requirements such as dynamic updates to the DNS, or client-specific requirements.

The model of operation for non-temporary address assignment is as follows:

- The server is provided with prefixes from which it may assign addresses to clients, as well as any related network topology information as to which prefixes are present on which links.
- A client requests a non-temporary address to be assigned by the server. The server allocates an address or addresses appropriate for the link on which the client is connected.
- The server returns the allocated address or addresses to the client.

Each address has associated preferred and valid lifetimes (see Section 12.1), which constitute an agreement about the length of time over which the client is allowed to use the address. A client can request an extension of the lifetimes on an address and is required to terminate the use of an address if the valid lifetime of the address expires.

Typically, clients request other configuration parameters, such as the DNS name server addresses and domain search lists, when requesting addresses.

Clients can also request more than one address or set of addresses (see Sections 6.5 and 12).

6.3. DHCP for Prefix Delegation

The prefix delegation mechanism is another stateful mode of operation and was originally intended for simple delegation of prefixes from a DHCP server to DHCP clients (typically routers). It is appropriate for situations in which the client (1) does not have knowledge about the topology of the networks to which it is attached and (2) does not require other information to choose a prefix for delegation. This mechanism is appropriate for use by an ISP to delegate a prefix to a subscriber, where the delegated prefix would possibly be subnetted and assigned to the links within the subscriber's network. [RFC7084] and [RFC7368] describe such use in detail.

The design of this prefix delegation mechanism meets the requirements for prefix delegation in [RFC3769].

DHCP prefix delegation itself does not require that the client forward IP packets not addressed to itself and thus does not require that the client (or server) be a router as defined in [RFC8200]. Also, in many cases (such as tethering or hosting virtual machines), hosts are already forwarding IP packets and thus are operating as routers as defined in [RFC8200].

The model of operation for prefix delegation is as follows:

- The server is provisioned with prefixes to be delegated to clients.
- A client requests prefix(es) from the server, as described in Section 18.
- The server chooses prefix(es) for delegation and responds with prefix(es) to the client.
- The client is then responsible for the delegated prefix(es). For example, the client might assign a subnet from a delegated prefix to one of its interfaces and begin sending Router Advertisements for the prefix on that link.

Each prefix has an associated preferred lifetime and valid lifetime (see Section 12.2), which constitute an agreement about the length of time over which the client is allowed to use the prefix. A client can request an extension of the lifetimes on a delegated prefix and is required to terminate the use of a delegated prefix if the valid lifetime of the prefix expires.

Figure 1 illustrates a network architecture in which prefix delegation could be used.

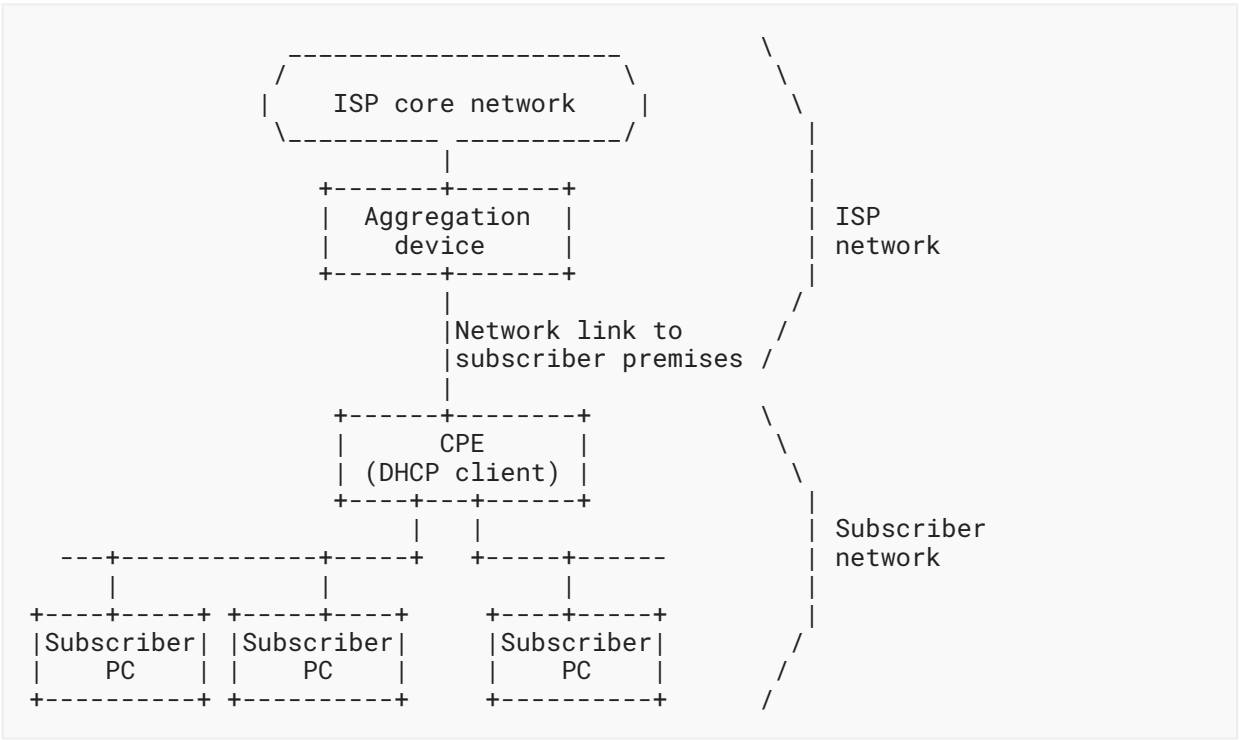


Figure 1: Prefix Delegation Network

In this example, the server (in the ISP core network or integrated in the aggregation device) is configured with a set of prefixes to be used for assignment to customers at the time of each customer's first connection to the ISP service. The prefix delegation process begins when the client (or Customer Premises Equipment (CPE)) requests configuration information through DHCP. The DHCP messages from the client are received by the server via the aggregation device. When the server receives the request, it selects an available prefix or prefixes for delegation to the client. The server then returns the prefix or prefixes to the client.

The client subnets the delegated prefix and assigns the longer prefixes to links in the subscriber's network. In a typical scenario based on the network shown in Figure 1, the client subnets a single delegated /48 prefix into /64 prefixes and assigns one /64 prefix to each of the links in the subscriber network.

The prefix delegation options can be used in conjunction with other DHCP options carrying other configuration information to the client. The client may, in turn, provide DHCP service to nodes attached to the internal network. For example, the client may obtain the addresses of DNS and NTP servers from the ISP server and then pass that configuration information on to the subscriber hosts through a DHCP server in the client.

If the client uses a delegated prefix to configure addresses on interfaces on itself or other nodes behind it, the preferred and valid lifetimes of those addresses **MUST** be no longer than the remaining preferred and valid lifetimes, respectively, for the delegated prefix at any time. In particular, if the delegated prefix or a prefix derived from it is advertised for stateless address autoconfiguration [RFC4862], the advertised preferred and valid lifetimes **MUST NOT** exceed the corresponding remaining lifetimes of the delegated prefix.

A client that has delegated any of the address space received through DHCP Prefix Delegation **MUST NOT** issue a DHCP Release on the relevant delegated prefix while any of the address space is outstanding. That includes addresses leased out by DHCPv6 (IA_NA), prefixes delegated via DHCPv6-PD (IA_PD), and addresses autoconfigured by IPv6 Router Advertisements. Requirement WPD-9 in [RFC9096] makes this the best current practice.

[RFC9096], Section 3.3 provides further guidance on coordination of lifetimes between WAN (DHCPv6-PD client) and LAN (DHCPv6-PD server) sides.

Several problems related to Prefix Delegation and Relay Agents and a set of requirements to address them are defined in [RFC8987].

6.4. DHCP for Customer Edge Routers

The DHCP requirements and network architecture for Customer Edge Routers are described in [RFC7084], with improvements for renumbering described in [RFC9096]. This model of operation combines address assignment (see Section 6.2) and prefix delegation (see Section 6.3). In general, this model assumes that a single set of transactions between the client and server will assign or extend the client's non-temporary addresses and delegated prefixes.

6.5. Multiple Addresses and Prefixes

DHCP allows a client to receive multiple addresses. During typical operation, a client sends one instance of an IA_NA option and the server assigns at most one address from each prefix assigned to the link to which the client is attached. In particular, the server can be configured to serve addresses out of multiple prefixes for a given link. This is useful in cases such as when a network renumbering event is in progress. In a typical deployment, the server will grant one address for each IA_NA option (see Section 21.4).

To meet the recommendations of [RFC7934], a client can explicitly request multiple addresses by sending multiple IA_NA options. A client can send multiple IA_NA options in its initial transmissions. Alternatively, it can send an extra Request message with additional new IA_NA options (or include them in a Renew message).

The same principle also applies to prefix delegation. In principle, DHCP allows a client to request new prefixes to be delegated by sending additional IA_PD options (see Section 21.21). However, a typical operator usually prefers to delegate a single, larger prefix. In most deployments, it is recommended that the client request a larger prefix in its initial transmissions rather than request additional prefixes later on.

The exact behavior of the server (whether to grant additional addresses and prefixes or not) is up to the server policy and is out of scope for this document.

For more information on how the server distinguishes between IA option instances, see Section 12.

6.6. Registering Self-Generated Addresses

[RFC9686] introduces a method for devices to register their self-generated or statically configured addresses in the DHCPv6 servers. The general idea is that devices would notify the server about addresses that they are using, so that the server can log or record these addresses as required by local policy.

The major specificity of this mechanism is that the address selection is not done by the DHCP server, but by the device itself. The majority of the lifecycle remains the same in principle: a lease is created by the server, the device performs periodic actions to get the lease renewed, and, eventually, the lease can expire. However, this mechanism uses different message types (ADDR-REG-INFORM and ADDR-REG-REPLY) and has different source address requirements, as defined in [RFC9686].

7. DHCP Constants

This section describes various program and networking constants used by DHCP.

7.1. Multicast Addresses

The following multicast addresses are used by DHCPv6:

All_DHCP_Relay_Agents_and_Servers (ff02::1:2)
: A link-scoped multicast address used by a client to communicate with neighboring (i.e., on-link) relay agents and servers. All servers and relay agents are members of this multicast group.

All_DHCP_Servers (ff05::1:3)
: A site-scoped multicast address used by a relay agent to communicate with servers, either because the relay agent wants to send messages to all servers or because it does not know the unicast addresses of the servers. Note that in order for a relay agent to use this address, it must have an address of sufficient scope to be reachable by the servers. All servers within the site are members of this multicast group on the interfaces that are within the site.

7.2. UDP Ports

Clients **MUST** listen for DHCP messages on UDP port 546. Servers and relay agents **MUST** listen for DHCP messages on UDP port 547.

Therefore, clients **MUST** send DHCP messages to UDP destination port 547. Servers **MUST** send Relay-reply messages to UDP destination port 547 and client messages to UDP destination port 546. Relay agents **MUST** send Relay-forward and Relay-reply messages to UDP destination port 547 and client messages to UDP destination port 546.

It is **RECOMMENDED** for clients to send messages from UDP source port 546 and for servers and relay agents from UDP source port 547. However, clients, servers, and relay agents **MAY** send DHCP messages from any UDP source port they are allowed to use.

Please note that the Relay Source Port Option [RFC8357] changes some of these rules for servers and relays agents.

7.3. DHCP Message Types

DHCP defines the following message types. The formats of these messages are provided in Sections 8 and 9. Additional message types have been defined and may be defined in the future; see <https://www.iana.org/assignments/dhcpv6-parameters>. The numeric encoding for each message type is shown in parentheses.

SOLICIT (1)
: A client sends a Solicit message to locate servers.

ADVERTISE (2)
: A server sends an Advertise message to indicate that it is available for DHCP service, in response to a Solicit message received from a client.

REQUEST (3)
: A client sends a Request message to request configuration parameters, including addresses and/or delegated prefixes, from a specific server.

CONFIRM (4)
: A client sends a Confirm message to any available server to determine whether the addresses it was assigned are still appropriate to the link to which the client is connected.

RENEW (5)
: A client sends a Renew message to the server that originally provided the client's leases and configuration parameters to extend the lifetimes on the leases assigned to the client and to update other configuration parameters.

REBIND (6)
: A client sends a Rebind message to any available server to extend the lifetimes on the leases assigned to the client and to update other configuration parameters; this message is sent after a client receives no response to a Renew message.

REPLY (7)
: A server sends a Reply message containing assigned leases and configuration parameters in response to a Solicit, Request, Renew, or Rebind message received from a client. A server sends a Reply message containing configuration parameters in response to an Information-request message. A server sends a Reply message in response to a Confirm message confirming or denying that the addresses assigned to the client are appropriate to the link to which the client is connected. A server sends a Reply message to acknowledge receipt of a Release or Decline message.

RELEASE (8)
: A client sends a Release message to the server that assigned leases to the client to indicate that the client will no longer use one or more of the assigned leases.

DECLINE (9)
: A client sends a Decline message to a server to indicate that the client has determined that one or more addresses assigned by the server are already in use on the link to which the client is connected.

RECONFIGURE (10)
: A server sends a Reconfigure message to a client to inform the client that the server has new or updated configuration parameters and that the client is to initiate a Renew/Reply, Rebind/Reply, or Information-request/Reply transaction with the server in order to receive the updated information.

INFORMATION-REQUEST (11)
: A client sends an Information-request message to a server to request configuration parameters without the assignment of any leases to the client.

RELAY-FORW (12)
: A relay agent sends a Relay-forward message to relay messages to servers, either directly or through another relay agent. The received message -- either a client message or a Relay-forward message from another relay agent -- is encapsulated in an option in the Relay-forward message.

RELAY-REPL (13)
: A server sends a Relay-reply message to a relay agent containing a message that the relay agent delivers to a client. The Relay-reply message may be relayed by other relay agents for delivery to the destination relay agent.

  The server encapsulates the client message as an option in the Relay-reply message, which the relay agent extracts and relays to the client.

7.4. DHCP Option Codes

DHCP makes extensive use of options in messages; some of these are defined later, in Section 21. Additional options are defined in other documents or may be defined in the future (see [RFC7227] for guidance on new option definitions).

7.5. Status Codes

DHCP uses status codes to communicate the success or failure of operations requested in messages from clients and servers and to provide additional information about the specific cause of the failure of a message. The specific status codes are defined in Section 21.13.

If the Status Code option (see Section 21.13) does not appear in a message in which the option could appear, the status of the message is assumed to be Success.

7.6. Transmission and Retransmission Parameters

The table of values (Table 1) is used to describe the message transmission behavior of clients and servers. Some of the values are adjusted by a randomization factor and backoffs (see Section 15). Transmissions may also be influenced by rate limiting (see Section 14.1).

Parameter	Default	Description
SOL_MAX_DELAY	1 sec	Max delay of first Solicit
SOL_TIMEOUT	1 sec	Initial Solicit timeout
SOL_MAX_RT	3600 secs	Max Solicit timeout value
REQ_TIMEOUT	1 sec	Initial Request timeout
REQ_MAX_RT	30 secs	Max Request timeout value
REQ_MAX_RC	10	Max Request retry attempts
CNF_MAX_DELAY	1 sec	Max delay of first Confirm
CNF_TIMEOUT	1 sec	Initial Confirm timeout
CNF_MAX_RT	4 secs	Max Confirm timeout
CNF_MAX_RD	10 secs	Max Confirm duration
REN_TIMEOUT	10 secs	Initial Renew timeout
REN_MAX_RT	600 secs	Max Renew timeout value
REB_TIMEOUT	10 secs	Initial Rebind timeout

Parameter	Default	Description
REB_MAX_RT	600 secs	Max Rebind timeout value
INF_MAX_DELAY	1 sec	Max delay of first Information-request
INF_TIMEOUT	1 sec	Initial Information-request timeout
INF_MAX_RT	3600 secs	Max Information-request timeout value
REL_TIMEOUT	1 sec	Initial Release timeout
REL_MAX_RC	4	Max Release retry attempts
DEC_TIMEOUT	1 sec	Initial Decline timeout
DEC_MAX_RC	4	Max Decline retry attempts
REC_TIMEOUT	2 secs	Initial Reconfigure timeout
REC_MAX_RC	8	Max Reconfigure attempts
HOP_COUNT_LIMIT	8	Max hop count in a Relay-forward message
IRT_DEFAULT	86400 secs (24 hours)	Default information refresh time
IRT_MINIMUM	600 secs	Min information refresh time
MAX_WAIT_TIME	60 secs	Max required time to wait for a response

Table 1: Transmission and Retransmission Parameters

7.7. Representation of Time Values and "Infinity" as a Time Value

All time values for lifetimes, T1, and T2 are unsigned 32-bit integers and are expressed in units of seconds. The value 0xffffffff is taken to mean "infinity" when used as a lifetime (as in [RFC4861]) or a value for T1 or T2.

Setting the valid lifetime of an address or a delegated prefix to 0xffffffff ("infinity") amounts to a permanent assignment of an address or delegation to a client and should only be used in cases where permanent assignments are desired.

Care should be taken in setting T1 or T2 to 0xffffffff ("infinity"). A client will never attempt to extend the lifetimes of any addresses in an IA with T1 set to 0xffffffff. A client will never attempt to use a Rebind message to locate a different server to extend the lifetimes of any addresses in an IA with T2 set to 0xffffffff.

8. Client/Server Message Formats

All DHCP messages sent between clients and servers share an identical fixed-format header and a variable-format area for options.

All values in the message header and in options are in network byte order.

Options are stored serially in the "options" field, with no padding between the options. Options are byte-aligned but are not aligned in any other way (such as on 2-byte or 4-byte boundaries).

The following diagram illustrates the format of DHCP messages sent between clients and servers:

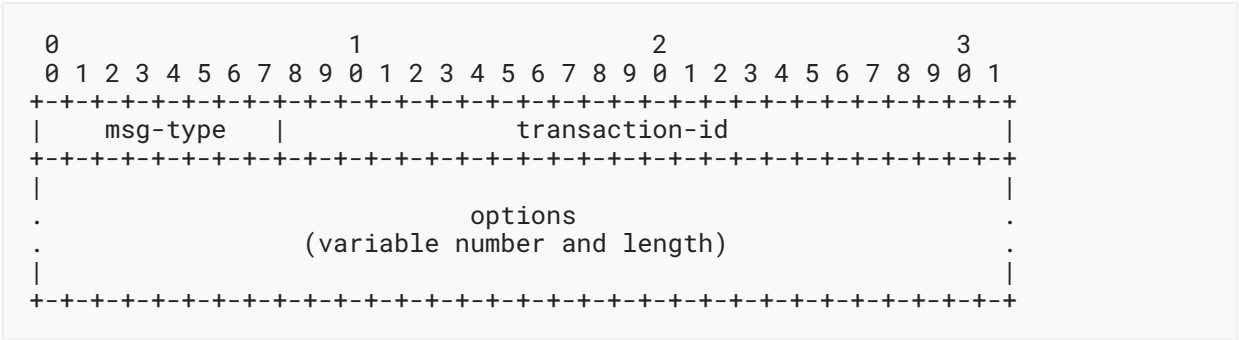


Figure 2: Client/Server Message Format

msg-type: Identifies the DHCP message type; the available message types are listed in Section 7.3. A 1-octet field.

transaction-id: The transaction ID for this message exchange. A 3-octet field.

options: Options carried in this message; options are described in Section 21. A variable-length field (4 octets less than the size of the message).

9. Relay Agent/Server Message Formats

Relay agents exchange messages with other relay agents and servers to relay messages between clients and servers that are not connected to the same link.

All values in the message header and in options are in network byte order.

Options are stored serially in the "options" field, with no padding between the options. Options are byte-aligned but are not aligned in any other way (such as on 2-byte or 4-byte boundaries).

There are two relay agent messages (Relay-forward and Relay-reply), which share the following format:

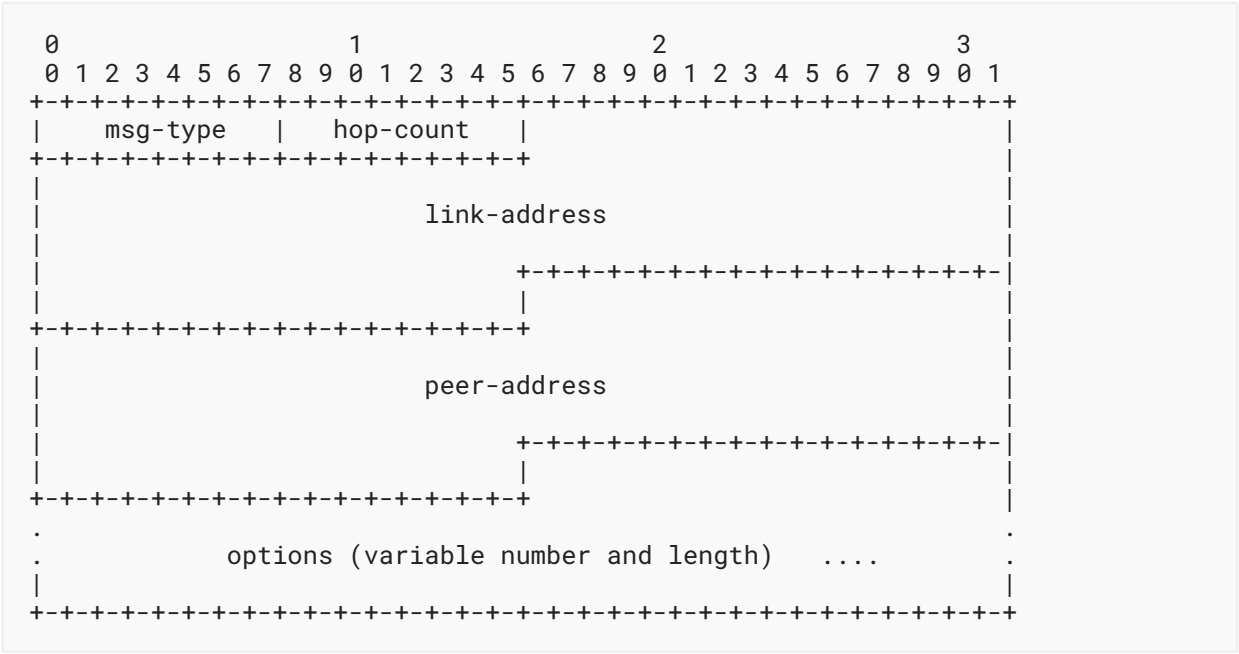


Figure 3: Relay Agent/Server Message Format

The following sections describe the use of the relay agent message header.

9.1. Relay-forward Message

The following list defines the use of message fields in a Relay-forward message.

msg-type: RELAY-FORW (12). A 1-octet field.

hop-count: Number of relay agents that have already relayed this message. A 1-octet field.

link-address: An address that may be used by the server to identify the link on which the client is located. This is typically a globally scoped unicast address (i.e., GUA or ULA), but see the discussion in Section 19.1.1. A 16-octet field.

peer-address: The address of the client or relay agent from which the message to be relayed was received. A 16-octet field.

options: **MUST** include a Relay Message option (see Section 21.10); **MAY** include other options, such as the Interface-Id option (see Section 21.18), added by the relay agent. A variable-length field (34 octets less than the size of the message).

See Section 13.1 for an explanation of how the link-address field is used.

9.2. Relay-reply Message

The following list defines the use of message fields in a Relay-reply message.

msg-type: RELAY-REPL (13). A 1-octet field.

hop-count: Copied from the Relay-forward message. A 1-octet field.

link-address: Copied from the Relay-forward message. A 16-octet field.

peer-address: Copied from the Relay-forward message. A 16-octet field.

options: **MUST** include a Relay Message option (see Section 21.10); **MAY** include other options, such as the Interface-Id option (see Section 21.18). A variable-length field (34 octets less than the size of the message).

10. Representation and Use of Domain Names

So that domain names may be encoded uniformly, a domain name or a list of domain names is encoded using the technique described in Section 3.1 of [RFC1035]. The message compression scheme in Section 4.1.4 of [RFC1035] **MUST NOT** be used.

11. DHCP Unique Identifier (DUID)

Each DHCP client and server has a DUID. DHCP servers use DUIDs to identify clients for the selection of configuration parameters and in the association of IAs with clients. DHCP clients use DUIDs to identify a server in messages where a server needs to be identified. See Sections 21.2 and 21.3 for details regarding the representation of a DUID in a DHCP message.

Clients and servers **MUST** treat DUIDs as opaque values and **MUST** only compare DUIDs for equality. Clients and servers **SHOULD NOT** in any other way interpret DUIDs. Clients and servers **MUST NOT** restrict DUIDs to the types defined in this document, as additional DUID types may be defined in the future. It should be noted that an attempt to parse a DUID to obtain a client's link-layer address is unreliable, as there is no guarantee that the client is still using the same link-layer address as when it generated its DUID. Also, such an attempt will be more and more unreliable as more clients adopt privacy measures such as those defined in [RFC7844]. If this capability is required, it is recommended to rely on the Client Link-Layer Address option instead [RFC6939].

The DUID is carried in an option because it may be variable in length and because it is not required in all DHCP messages. The DUID is designed to be unique across all DHCP clients and servers, and stable for any specific client or server. That is, the DUID used by a client or server **SHOULD NOT** change over time if at all possible; for example, a device's DUID should not change as a result of a change in the device's network hardware or changes to virtual interfaces (e.g., logical PPP (over Ethernet) interfaces that may come and go in Customer Premises Equipment routers). The client may change its DUID as specified in [RFC7844].

The motivation for having more than one type of DUID is that the DUID must be globally unique and must also be easy to generate. The sort of globally unique identifier that is easy to generate for any given device can differ quite widely. Also, some devices may not contain any persistent storage. Retaining a generated DUID in such a device is not possible, so the DUID scheme must accommodate such devices.

11.1. DUID Contents

A DUID consists of a 2-octet type code represented in network byte order, followed by a variable number of octets that make up the actual identifier. The length of the DUID (not including the type code) is at least 1 octet and at most 128 octets. The following types are currently defined:

Type	Description
1	Link-layer address plus time
2	Vendor-assigned unique ID based on Enterprise Number
3	Link-layer address
4	Universally Unique Identifier (UUID) [RFC6355]

Table 2: DUID Types

Formats for the variable field of the DUID for the first three of the above types are shown below. The fourth type, DUID-UUID [RFC6355], can be used in situations where there is a UUID stored in a device's firmware settings.

11.2. DUID Based on Link-Layer Address Plus Time (DUID-LLT)

This type of DUID consists of a 2-octet type field containing the value 1, a 2-octet hardware type code, and 4 octets containing a time value, followed by the link-layer address of any one network interface that is connected to the DHCP device at the time that the DUID is generated. The time value is the time that the DUID is generated, represented in seconds since midnight (UTC), January 1, 2000, modulo 2^32. The hardware type **MUST** be a valid hardware type assigned by IANA; see [IANA-HARDWARE-TYPES]. Both the time and the hardware type are stored in network byte order. For Ethernet hardware types, the link-layer address is stored in canonical form, as described in [RFC2464].

The following diagram illustrates the format of a DUID-LLT:

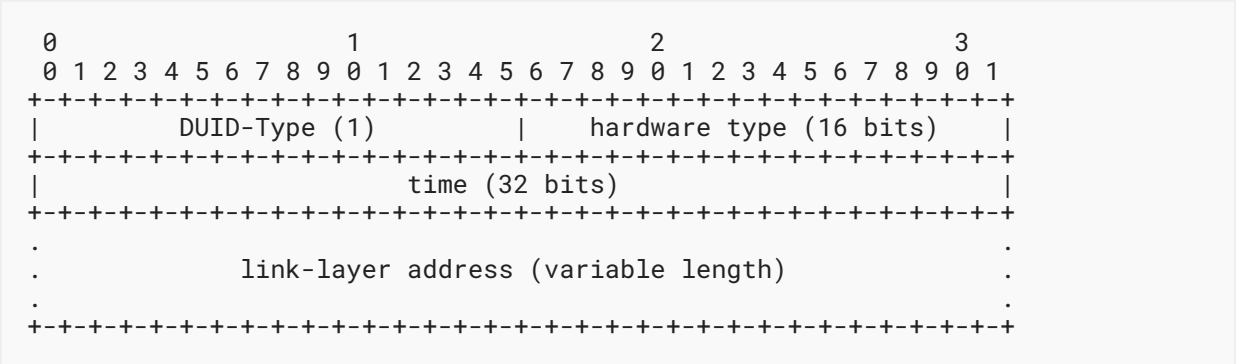


Figure 4: DUID-LLT Format

The choice of network interface can be completely arbitrary, as long as that interface provides a globally unique link-layer address for the link type; the same DUID-LLT **SHOULD** be used in configuring all network interfaces connected to the device, regardless of which interface's link-layer address was used to generate the DUID-LLT.

Clients and servers using this type of DUID **MUST** store the DUID-LLT in stable storage and **MUST** continue to use this DUID-LLT even if the network interface used to generate the DUID-LLT is removed. Clients and servers that do not have any stable storage **MUST NOT** use this type of DUID.

Clients and servers that use this DUID **SHOULD** attempt to configure the time prior to generating the DUID, if that is possible, and **MUST** use some sort of time source (for example, a real-time clock) in generating the DUID, even if that time source could not be configured prior to generating the DUID. The use of a time source makes it unlikely that two identical DUID-LLTs will be generated if the network interface is removed from the client and another client then uses the same network interface to generate a DUID-LLT. A collision between two DUID-LLTs is very unlikely even if the clocks have not been configured prior to generating the DUID.

This method of DUID generation is recommended for all general-purpose computing devices such as desktop computers and laptop computers, and also for devices such as printers, routers, and so on, that contain some form of writable non-volatile storage.

It is possible that this algorithm for generating a DUID could result in a client identifier collision. A DHCP client that generates a DUID-LLT using this mechanism **MUST** provide an administrative interface that replaces the existing DUID with a newly generated DUID-LLT.

11.3. DUID Assigned by Vendor Based on Enterprise Number (DUID-EN)

The vendor assigns this form of DUID to the device. This DUID consists of the 4-octet vendor's registered Private Enterprise Number as maintained by IANA [IANA-PEN] followed by a unique identifier assigned by the vendor. The following diagram summarizes the structure of a DUID-EN:

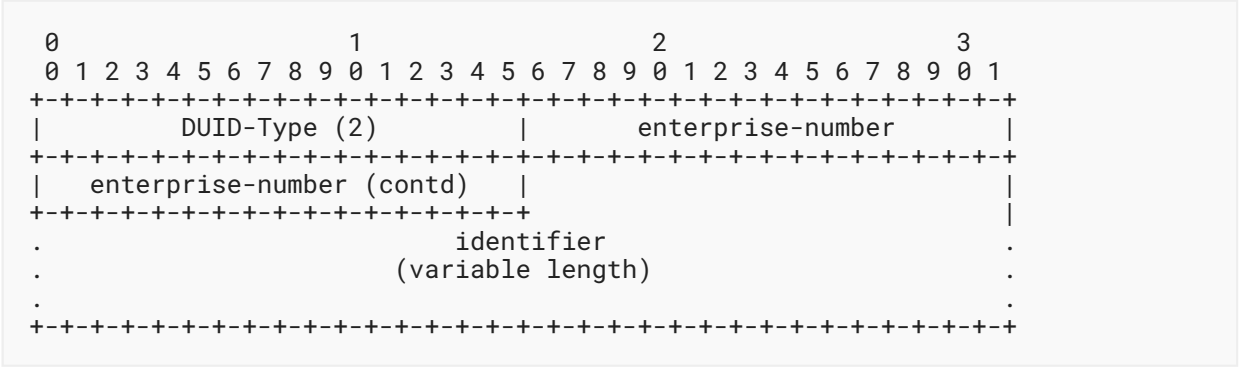


Figure 5: DUID-EN Format

The source of the identifier is left up to the vendor defining it, but each identifier part of each DUID-EN **MUST** be unique to the device that is using it, and **MUST** be assigned to the device no later than at the first usage and stored in some form of non-volatile storage. This typically means being assigned during the manufacturing process in the case of physical devices or, in the case of virtual machines, when the image is created or booted for the first time. The generated DUID **SHOULD** be recorded in non-erasable storage. The enterprise-number is the vendor's registered Private Enterprise Number as maintained by IANA [IANA-PEN]. The enterprise-number is stored as an unsigned 32-bit number.

An example DUID of this type might look like this:

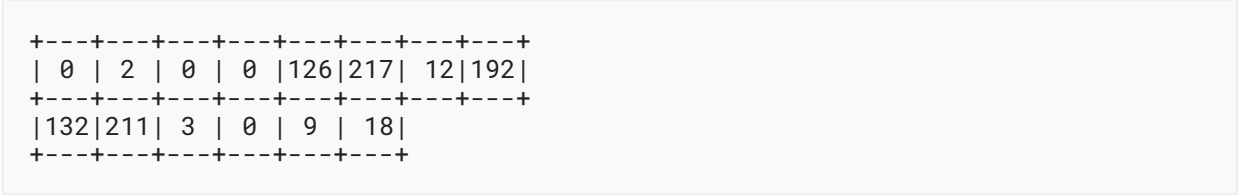


Figure 6: DUID-EN Example

This example includes the 2-octet type of 2 and the Enterprise Number (32473) (from [RFC5612]), followed by 8 octets of identifier data (0x0CC084D303000912).

11.4. DUID Based on Link-Layer Address (DUID-LL)

This type of DUID consists of 2 octets containing a DUID type of 3 and a 2-octet network hardware type code, followed by the link-layer address of any one network interface that is permanently connected to the client or server device. For example, a node that has a network interface implemented in a chip that is unlikely to be removed and used elsewhere could use a DUID-LL. The hardware type **MUST** be a valid hardware type assigned by IANA; see [IANA-HARDWARE-TYPES]. The hardware type is stored in network byte order. The link-layer address is stored in canonical form, as described in [RFC2464]. The following diagram illustrates the format of a DUID-LL:

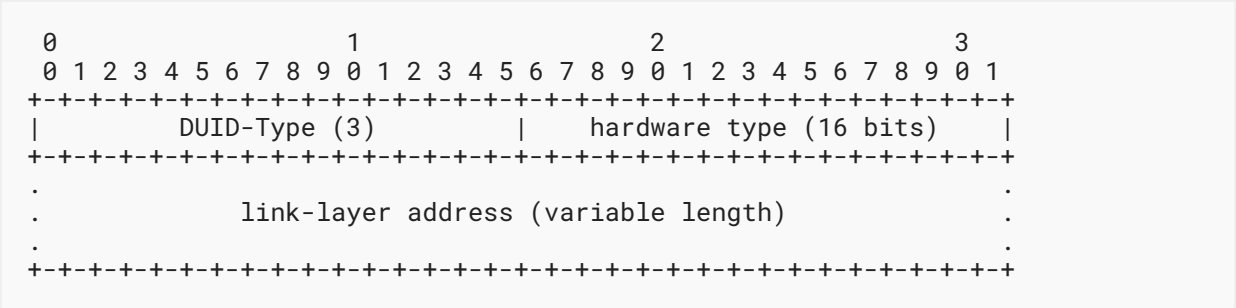


Figure 7: DUID-LL Format

The choice of network interface can be completely arbitrary, as long as that interface provides a unique link-layer address and is permanently attached to the device on which the DUID-LL is being generated. The same DUID-LL **SHOULD** be used in configuring all network interfaces connected to the device, regardless of which interface's link-layer address was used to generate the DUID.

A DUID-LL is recommended for devices that have a permanently connected network interface with a link-layer address and do not have nonvolatile, writable stable storage. A DUID-LL **SHOULD NOT** be used by DHCP clients or servers that cannot tell whether or not a network interface is permanently attached to the device on which the DHCP client is running.

11.5. DUID Based on Universally Unique Identifier (DUID-UUID)

This type of DUID consists of 16 octets containing a 128-bit UUID. [RFC6355] details when to use this type and how to pick an appropriate source of the UUID.

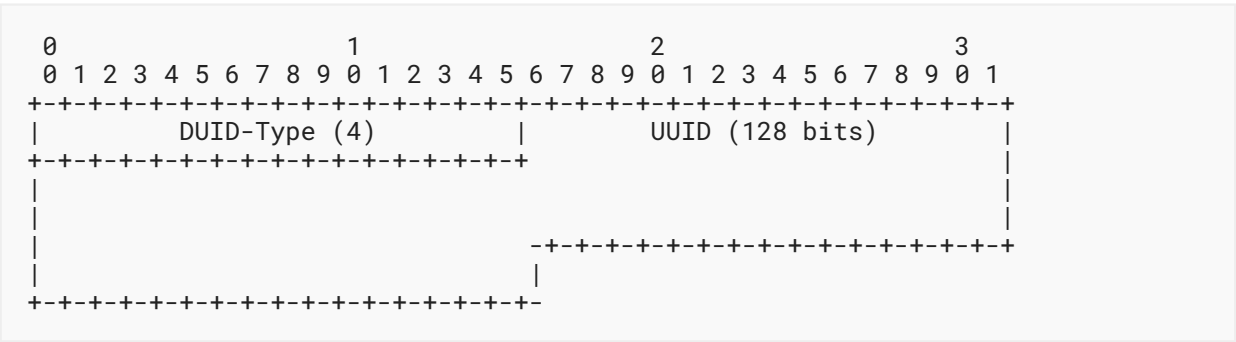


Figure 8: DUID-UUID Format

12. Identity Association

An Identity Association (IA) is a construct through which a server and a client can identify, group, and manage a set of related IPv6 addresses or delegated prefixes. Each IA consists of an IAID and associated configuration information.

The IAID uniquely identifies the IA and **MUST** be chosen to be unique among the IAIDs for that IA type on the client (e.g., an IA_NA with an IAID of 0 and an IA_PD with an IAID of 0 are each considered unique). The IAID is chosen by the client. For any given use of an IA by the client, the IAID for that IA **MUST** be consistent across restarts of the DHCP client. The client may maintain consistency by either storing the IAID in non-volatile storage or using an algorithm that will consistently produce the same IAID as long as the configuration of the client has not changed. There may be no way for a client to maintain consistency of the IAIDs if it does not have non-volatile storage and the client's hardware configuration changes. If the client uses only one IAID, it can use a well-known value, e.g., zero.

If the client wishes to obtain a distinctly new address or prefix and deprecate the existing one, the client sends a Release message to the server for the IAs using the original IAID. The client then creates a new IAID, to be used in future messages to obtain leases for the new IA.

12.1. Identity Associations for Address Assignment

A client must associate at least one distinct IA with each of its network interfaces for which it is to request the assignment of IPv6 addresses from a DHCP server. The client uses the IAs assigned to an interface to obtain configuration information from a server for that interface. Each such IA must be associated with exactly one interface.

The configuration information in an IA_NA option consists of one or more IPv6 addresses along with the T1 and T2 values for the IA. See Section 21.4 for details regarding the representation of an IA_NA in a DHCP message.

Each address in an IA has a preferred lifetime and a valid lifetime, as defined in [RFC4862]. The lifetimes are transmitted from the DHCP server to the client in the IA Address option (see Section 21.6). The lifetimes apply to the use of addresses; see Section 5.5.4 of [RFC4862].

12.2. Identity Associations for Prefix Delegation

An IA_PD is different from an IA for address assignment in that it does not need to be associated with exactly one interface. One IA_PD can be associated with the client, with a set of interfaces, or with exactly one interface. A client configured to request delegated prefixes must create at least one distinct IA_PD. It may associate a distinct IA_PD with each of its downstream network interfaces and use that IA_PD to obtain a prefix for that interface from the server.

The configuration information in an IA_PD option consists of one or more prefixes along with the T1 and T2 values for the IA_PD. See Section 21.21 for details regarding the representation of an IA_PD in a DHCP message.

Each delegated prefix in an IA has a preferred lifetime and a valid lifetime, as defined in [RFC4862]. The lifetimes are transmitted from the DHCP server to the client in the IA Prefix option (see Section 21.22). The lifetimes apply to the use of delegated prefixes; see Section 5.5.4 of [RFC4862].

13. Assignment to an IA

13.1. Selecting Addresses for Assignment to an IA_NA

A server selects addresses to be assigned to an IA_NA according to the address assignment policies determined by the server administrator and the specific information the server determines about the client from some combination of the following sources:

- The link to which the client is attached. The server determines the link as follows:
  - If the server receives the message directly from the client and the source address in the IP datagram in which the message was received is a link-local address, then the client is on the same link to which the interface over which the message was received is attached.
  - If the server receives the message from a forwarding relay agent, then the client is on the same link as the one to which the interface, identified by the link-address field in the message from the relay agent, is attached. According to [RFC6221], the server **MUST** ignore any link-address field whose value is zero. The link-address in this case may come from any of the Relay-forward messages encapsulated in the received Relay-forward, and in general the most encapsulated (closest Relay-forward to the client) has the most useful value.
- The DUID supplied by the client.
- Other information in options supplied by the client, e.g., IA Address options (see Section 21.6) that include the client's requests for specific addresses.
- Other information in options supplied by the relay agent.

By default, DHCP server implementations **SHOULD NOT** generate predictable addresses (see Section 4.7 of [RFC7721]). Server implementers are encouraged to review [RFC8981], [RFC7824], [RFC7707], and [RFC7943] as to possible considerations for how to generate addresses.

A server **MUST NOT** assign an address that is otherwise reserved for some other purpose. For example, a server **MUST NOT** assign addresses that use a reserved IPv6 Interface Identifier [RFC5453] [RFC7136] [IANA-RESERVED-IID].

See [RFC7969] for a more detailed discussion on how servers determine a client's location on the network.

13.2. Assignment of Prefixes for IA_PD

The mechanism through which the server selects prefix(es) for delegation is not specified in this document. Examples of ways in which the server might select prefix(es) for a client include static assignment based on subscription to an ISP, dynamic assignment from a pool of available prefixes, and selection based on an external authority such as a RADIUS server using the Framed-IPv6-Prefix option as described in [RFC3162].

14. Transmission of Messages by a Client

Unless otherwise specified in this document or in a document that describes how IPv6 is carried over a specific type of link (for link types that do not support multicast), a client sends DHCP messages to the All_DHCP_Relay_Agents_and_Servers multicast address.

DHCP servers **SHOULD NOT** check to see whether the Layer 2 address used was multicast or not, as long as the Layer 3 address was correct.

A client uses multicast to reach all servers or an individual server. An individual server is indicated by specifying that server's DUID in a Server Identifier option (see Section 21.3) in the client's message. (All servers will receive this message, but only the indicated server will respond.) All servers are indicated when this option is not supplied.

14.1. Rate Limiting

A DHCPv6 client **MUST** limit the rate of DHCP messages it transmits or retransmits. This will minimize the impact of prolonged message bursts or loops, for example when a client rejects a server's response, repeats the request and gets the same server response, which, again, gets rejected by the client.

This loop can repeat infinitely if there is not a quit/stop mechanism. Therefore, a client must not initiate transmissions too frequently.

A recommended method for implementing the rate-limiting function is a token bucket (see Appendix A of [RFC3290]), limiting the average rate of transmission to a certain number in a certain time interval. This method of bounding burstiness also guarantees that the long-term transmission rate will not be exceeded.

A transmission rate limit **SHOULD** be configurable. A possible default could be 20 messages in 20 seconds.

For a device that has multiple interfaces, the limit **MUST** be enforced on a per-interface basis.

Rate limiting of forwarded DHCP messages and server-side messages is out of scope for this specification.

14.2. Client Behavior when T1 and/or T2 Are 0

In certain cases, T1 and/or T2 values may be set to 0. Currently, there are two such cases:

1. a client received an IA_NA option (see Section 21.4) with a zero value
2. a client received an IA_PD option (see Section 21.21) with a zero value

This is an indication that the renew and rebind times are left to the discretion of the client. However, they are not completely discretionary.

When T1 and/or T2 values are set to 0, the client **MUST** choose a time to avoid message storms. In particular, it **MUST NOT** transmit immediately. If the client received multiple IA options, it **SHOULD** pick renew and/or rebind transmission times so all IA options are handled in one exchange, if possible. The client **MUST** choose renew and rebind times to not violate rate-limiting restrictions as defined in Section 14.1.

15. Reliability of Client-Initiated Message Exchanges

DHCP clients are responsible for reliable delivery of messages in the client-initiated message exchanges described in Section 18. If a DHCP client fails to receive an expected response from a server, the client must retransmit its message according to the retransmission strategy described below.

Note that the procedure described in this section is slightly modified when used with the Solicit message. The modified procedure is described in Section 18.2.1.

The client begins the message exchange by transmitting a message to the server. The message exchange terminates when either (1) the client successfully receives the appropriate response or responses from a server or servers or (2) the message exchange is considered to have failed according to the retransmission mechanism described below.

The client **MUST** update an "elapsed-time" value within an Elapsed Time option (see Section 21.9) in the retransmitted message. In some cases, the client may also need to modify values in IA Address options (see Section 21.6) or IA Prefix options (see Section 21.22) if a valid lifetime for any of the client's leases expires before retransmission. Thus, whenever this document refers to a "retransmission" of a client's message, it refers to both modifying the original message and sending this new message instance to the server.

The client retransmission behavior is controlled and described by the following variables:

RT: Retransmission timeout

IRT: Initial retransmission time

MRC: Maximum retransmission count

MRT: Maximum retransmission time

MRD: Maximum retransmission duration

RAND: Randomization factor

Specific values for each of these parameters relevant to the various messages are given in the subsections of Section 18.2, using values defined in Table 1 in Section 7.6. The algorithm for RAND is common across all message transmissions.

With each message transmission or retransmission, the client sets RT according to the rules given below. If RT expires before the message exchange terminates, the client recomputes RT and retransmits the message.

Each of the computations of a new RT includes a randomization factor (RAND), which is a random number chosen with a uniform distribution between -0.1 and +0.1. The randomization factor is included to minimize synchronization of messages transmitted by DHCP clients.

The algorithm for choosing a random number does not need to be cryptographically sound. The algorithm **SHOULD** produce a different sequence of random numbers from each invocation of the DHCP client.

RT for the first message transmission is based on IRT:

```
RT = IRT + RAND*IRT
```

RT for each subsequent message transmission is based on the previous value of RT:

```
RT = 2*RTprev + RAND*RTprev
```

MRT specifies an upper bound on the value of RT (disregarding the randomization added by the use of RAND). If MRT has a value of 0, there is no upper limit on the value of RT. Otherwise:

```
if (RT > MRT)
   RT = MRT + RAND*MRT
```

MRC specifies an upper bound on the number of times a client may retransmit a message. Unless MRC is zero, the message exchange fails once the client has transmitted the message MRC times.

MRD specifies an upper bound on the length of time a client may retransmit a message. Unless MRD is zero, the message exchange fails once MRD seconds have elapsed since the client first transmitted the message.

If both MRC and MRD are non-zero, the message exchange fails whenever either of the conditions specified in the previous two paragraphs is met.

If both MRC and MRD are zero, the client continues to transmit the message until it receives a response.

A client is not expected to listen for a response during the entire RT period and may turn off listening capabilities after waiting at least the shorter of RT and MAX_WAIT_TIME due to power consumption saving or other reasons. Of course, a client **MUST** listen for a Reconfigure if it has negotiated for its use with the server.

16. Message Validation

This section describes which options are valid in which kinds of message types and explains what to do when a client or server receives a message that contains known options that are invalid for that message. For example, an IA option is not allowed to appear in an Information-request message.

Clients and servers **MAY** choose to either (1) extract information from such a message if the information is of use to the recipient or (2) ignore such a message completely and just discard it.

If a server receives a message that it considers invalid, it **MAY** send a Reply message (or Advertise message, as appropriate) with a Server Identifier option (see Section 21.3), a Client Identifier option (see Section 21.2) (if one was included in the message), and a Status Code option (see Section 21.13) with status UnspecFail.

Clients, relay agents, and servers **MUST NOT** discard messages that contain unknown options (or instances of vendor options with unknown enterprise-number values). These options should be ignored as if they were not present. This is critical to provide for future extensions of DHCP.

A client or server **MUST** discard any received DHCP messages with an unknown message type.

Clients **SHOULD NOT** accept multicast messages.

Servers **SHOULD NOT** accept unicast traffic from clients. The Server Unicast option (see Section 21.12) and UseMulticast status code (see Section 21.13) have been obsoleted; hence, clients should no longer send messages to a server's unicast address nor receive the UseMulticast status code. However, a server that previously supported the Server Unicast option and is upgraded to not support it **MAY** continue to receive unicast messages if it previously sent the client the Server Unicast option. However, this causes no harm and the client will eventually switch back to sending multicast messages (such as after the lease's rebinding time is reached or the client is rebooted).

Relay agents **SHOULD NOT** accept unicast messages from clients.

Note: The multicast/unicast rules mentioned above apply to the DHCP messages within this document. Messages defined in other and future documents may have different rules.

16.1. Use of Transaction IDs

The "transaction-id" field holds a value used by clients and servers to correlate server responses to client messages. A client **SHOULD** generate a random number that cannot easily be guessed or predicted to use as the transaction ID for each new message it sends. Note that if a client generates easily predictable transaction identifiers, it may become more vulnerable to certain kinds of attacks from off-path intruders. A client **MUST** leave the transaction ID unchanged in retransmissions of a message.

16.2. Solicit Message

Clients **MUST** discard any received Solicit messages.

Servers **MUST** discard any Solicit messages that do not include a Client Identifier option or that do include a Server Identifier option.

16.3. Advertise Message

Clients **MUST** discard any received Advertise message that meets any of the following conditions:

- the message does not include a Server Identifier option (see Section 21.3).
- the message does not include a Client Identifier option (see Section 21.2).
- the contents of the Client Identifier option do not match the client's DUID.
- the "transaction-id" field value does not match the value the client used in its Solicit message.

Servers and relay agents **MUST** discard any received Advertise messages.

16.4. Request Message

Clients **MUST** discard any received Request messages.

Servers **MUST** discard any received Request message that meets any of the following conditions:

- the message does not include a Server Identifier option (see Section 21.3).
- the contents of the Server Identifier option do not match the server's DUID.
- the message does not include a Client Identifier option (see Section 21.2).

16.5. Confirm Message

Clients **MUST** discard any received Confirm messages.

Servers **MUST** discard any received Confirm messages that do not include a Client Identifier option (see Section 21.2) or that do include a Server Identifier option (see Section 21.3).

16.6. Renew Message

Clients **MUST** discard any received Renew messages.

Servers **MUST** discard any received Renew message that meets any of the following conditions:

- the message does not include a Server Identifier option (see Section 21.3).
- the contents of the Server Identifier option do not match the server's identifier.
- the message does not include a Client Identifier option (see Section 21.2).

16.7. Rebind Message

Clients **MUST** discard any received Rebind messages.

Servers **MUST** discard any received Rebind messages that do not include a Client Identifier option (see Section 21.2) or that do include a Server Identifier option (see Section 21.3).

16.8. Decline Message

Clients **MUST** discard any received Decline messages.

Servers **MUST** discard any received Decline message that meets any of the following conditions:

- the message does not include a Server Identifier option (see Section 21.3).
- the contents of the Server Identifier option do not match the server's identifier.
- the message does not include a Client Identifier option (see Section 21.2).

16.9. Release Message

Clients **MUST** discard any received Release messages.

Servers **MUST** discard any received Release message that meets any of the following conditions:

- the message does not include a Server Identifier option (see Section 21.3).
- the contents of the Server Identifier option do not match the server's identifier.
- the message does not include a Client Identifier option (see Section 21.2).

16.10. Reply Message

Clients **MUST** discard any received Reply message that meets any of the following conditions:

- the message does not include a Server Identifier option (see Section 21.3).
- the "transaction-id" field in the message does not match the value used in the original message.

If the client included a Client Identifier option (see Section 21.2) in the original message, the Reply message **MUST** include a Client Identifier option, and the contents of the Client Identifier option **MUST** match the DUID of the client. If the client did not include a Client Identifier option in the original message, the Reply message **MUST NOT** include a Client Identifier option.

Servers and relay agents **MUST** discard any received Reply messages.

16.11. Reconfigure Message

Servers and relay agents **MUST** discard any received Reconfigure messages.

Clients **MUST** discard any Reconfigure message that meets any of the following conditions:

- the message was not unicast to the client.
- the message does not include a Server Identifier option (see Section 21.3).
- the message does not include a Client Identifier option (see Section 21.2) that contains the client's DUID.
- the message does not include a Reconfigure Message option (see Section 21.19).
- the Reconfigure Message option msg-type is not a valid value.
- the message does not include authentication (such as RKAP; see Section 20.4) or fails authentication validation.

16.12. Information-request Message

Clients **MUST** discard any received Information-request messages.

Servers **MUST** discard any received Information-request message that meets any of the following conditions:

- the message includes a Server Identifier option (see Section 21.3), and the DUID in the option does not match the server's DUID.
- the message includes an IA option.

16.13. Relay-forward Message

Clients **MUST** discard any received Relay-forward messages.

16.14. Relay-reply Message

Clients and servers **MUST** discard any received Relay-reply messages.

17. Client Source Address and Interface Selection

The client's behavior regarding interface selection is different, depending on the purpose of the configuration.

17.1. Source Address and Interface Selection for Address Assignment

When a client sends a DHCP message to the All_DHCP_Relay_Agents_and_Servers multicast address, it **SHOULD** send the message through the interface for which configuration information (including the addresses) is being requested. However, the client **MAY** send the message through another interface if the interface for which configuration is being requested is a logical interface without direct link attachment or the client is certain that two interfaces are attached to the same link.

17.2. Source Address and Interface Selection for Prefix Delegation

Delegated prefixes are not associated with a particular interface in the same way as addresses are for address assignment as mentioned in Section 17.1 above.

When a client sends a DHCP message for the purpose of prefix delegation, it **SHOULD** be sent on the interface associated with the upstream router (typically, connected to an ISP network); see [RFC7084]. The upstream interface is typically determined by configuration. This rule applies even in the case where a separate IA_PD is used for each downstream interface.

18. DHCP Configuration Exchanges

A client initiates a message exchange with a server or servers to acquire or update configuration information of interest. A client has many reasons to initiate the configuration exchange. Some of the more common ones are:

1. as part of the operating system configuration/bootstrap process,
2. when requested to do so by the application layer (through an operating-system-specific API),
3. when a Router Advertisement indicates that DHCPv6 is available for address configuration (see Section 4.2 of [RFC4861]),
4. as required to extend the lifetime of address(es) and/or delegated prefix(es), using Renew and Rebind messages, or
5. upon the receipt of a Reconfigure message, when requested to do so by a server.

The client is responsible for creating IAs and requesting that a server assign addresses and/or delegated prefixes to the IAs. The client first creates the IAs and assigns IAIDs to them. The client then transmits a Solicit message containing the IA options describing the IAs. The client **MUST NOT** be using any of the addresses or delegated prefixes for which it tries to obtain the bindings by sending the Solicit message. In particular, if the client had some valid bindings and has chosen to start the server discovery process to obtain the same bindings from a different server, the client **MUST** stop using the addresses and delegated prefixes for the bindings that it had obtained from the previous server (see Section 18.2.7 for more details on what "stop using" means in this context) and that it is now trying to obtain from a new server.

A DHCP client that does not need to have a DHCP server assign IP addresses or delegated prefixes to it can obtain configuration information such as a list of available DNS servers [RFC3646] or NTP servers [RFC5908] through a single message and reply exchange with a DHCP server. To obtain configuration information, the client first sends an Information-request message (see Section 18.2.6) to the All_DHCP_Relay_Agents_and_Servers multicast address. Servers respond with a Reply message containing the configuration information for the client (see Section 18.3.6).

To request the assignment of one or more addresses or delegated prefixes, a client first locates a DHCP server and then requests the assignment of addresses/prefixes and other configuration information from the server. The client does this by sending the Solicit message (see Section

18.2.1) to the All_DHCP_Relay_Agents_and_Servers multicast address and collecting Advertise messages from the servers that respond to the client's message; the client then selects a server from which it wants to obtain configuration information. This process is referred to as server discovery. When the client has selected the server, it sends a Request message to that server as described in Section 18.2.2.

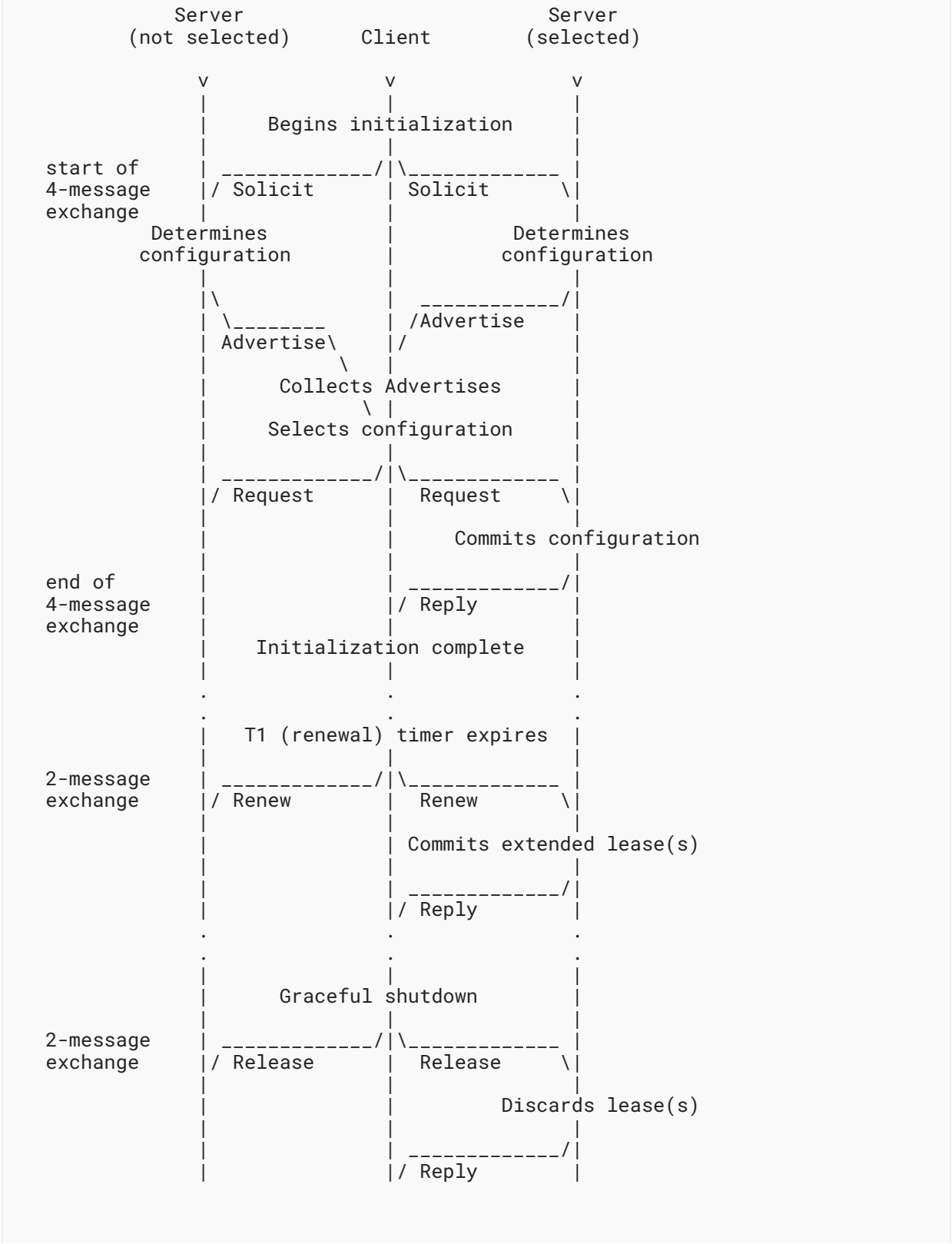
A client willing to perform the Solicit/Reply message exchange described in Section 18.2.1 includes a Rapid Commit option (see Section 21.14) in its Solicit message.

Servers that can assign addresses or delegated prefixes to the IAs respond to the client with an Advertise message or Reply message if the client included a Rapid Commit option and the server is configured to accept it.

If the server responds with an Advertise message, the client initiates a configuration exchange as described in Section 18.2.2.

A server may initiate a message exchange with a client by sending a Reconfigure message to cause the client to send a Renew, Rebind, or Information-request message to refresh its configuration information as soon as the Reconfigure message is received by the client.

Figure 9 shows a timeline diagram of the messages exchanged between a client and two servers for the typical lifecycle of one or more leases. This starts with the four-message Solicit/Advertise/Request/Reply exchange to obtain the lease(s), followed by a two-message Renew/Reply exchange to extend the lifetime on the lease(s), and then ends with a two-message Release/Reply exchange to end the client's use of the lease(s).


Server
(not selected)
Client
Server
(selected)
Begins initialization
start of
4-message
exchange
Solicit
Solicit
Determines
configuration
Determines
configuration
Advertise
/Advertise
Collects Advertises
Selects configuration
Request
Request
Commits configuration
end of
4-message
exchange
Reply
Initialization complete
T1 (renewal) timer expires
2-message
exchange
Renew
Renew
Commits extended lease(s)
Reply
Graceful shutdown
2-message
exchange
Release
Release
Discards lease(s)
Reply

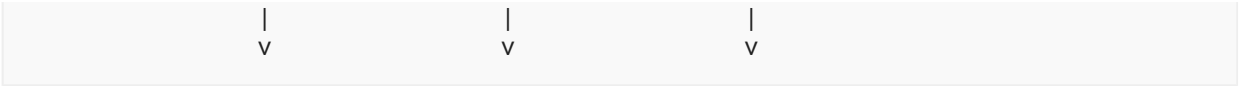


Figure 9: Timeline Diagram of the Messages Exchanged Between a Client and Two Servers for the Typical Lifecycle of One or More Leases

18.1. A Single Exchange for Multiple IA Options

This document assumes that a client **SHOULD** use a single transaction for all of the IA options required on an interface; this simplifies the client implementation and reduces the potential number of transactions required. To facilitate a client's use of a single transaction for all IA options, servers **MUST** return the same T1/T2 values for all IA options in a Reply (see Sections 18.3.2, 18.3.4, and 18.3.5) so that the client will generate a single transaction when renewing or rebinding its leases. However, because some servers may not yet conform to this requirement, a client **MUST** be prepared to select appropriate T1/T2 times as described in Section 18.2.4.

18.2. Client Behavior

A client uses the Solicit message to discover DHCP servers configured to assign leases or return other configuration parameters on the link to which the client is attached.

A client uses Request, Renew, Rebind, Release, and Decline messages during the normal lifecycle of addresses and delegated prefixes. When a client detects that it may have moved to a new link, it uses Confirm if it only has addresses and Rebind if it has delegated prefixes (and addresses). It uses Information-request messages when it needs configuration information but no addresses and no prefixes.

When a client requests multiple IA option types or multiple instances of the same IA types in a Solicit, Request, Renew, or Rebind, it is possible that the available server(s) may only be configured to offer a subset of them. When possible, the client **SHOULD** use the best configuration available and continue to request the additional IAs in subsequent messages. This allows the client to maintain a single session and state machine. In practice, especially in the case of handling IA_NA and IA_PD requests [RFC7084], this situation should be rare or a result of a temporary operational error. Thus, it is more likely that the client will get all configuration if it continues, in each subsequent configuration exchange, to request all the configuration information it is programmed to try to obtain, including any stateful configuration options for which no results were returned in previous message exchanges.

Upon receipt of a Reconfigure message from the server, a client responds with a Renew, Rebind, or Information-request message as indicated by the Reconfigure Message option (see Section 21.19). The client **SHOULD** be suspicious of the Reconfigure message (they may be faked), and it **MUST NOT** abandon any resources it might have already obtained. The client **SHOULD** treat the Reconfigure message as if the T1 timer had expired. The client will expect the server to send IAs and/or other configuration information to the client in a Reply message.

18.2.1. Creation and Transmission of Solicit Messages

The client sets the "msg-type" field to SOLICIT. The client generates a transaction ID and inserts this value in the "transaction-id" field.

The client **MUST** include a Client Identifier option (see Section 21.2) to identify itself to the server. The client includes IA options for any IAs to which it wants the server to assign leases.

The client **MUST** include an Elapsed Time option (see Section 21.9) to indicate how long the client has been trying to complete the current DHCP message exchange.

The client uses IA_NA options (see Section 21.4) to request the assignment of non-temporary addresses and IA_PD options (see Section 21.21) to request prefix delegation. IA_NA or IA_PD options, or a combination, can be included in DHCP messages. In addition, multiple instances of any IA option type can be included.

The client **MAY** include addresses in IA Address options (see Section 21.6) encapsulated within IA_NA option as hints to the server about the addresses for which the client has a preference.

The client **MAY** include values in IA Prefix options (see Section 21.22) encapsulated within IA_PD options as hints for the delegated prefix and/or prefix length for which the client has a preference. See Section 18.2.4 for more on prefix-length hints.

The client **MUST** include an Option Request option (ORO) (see Section 21.7) to request the SOL_MAX_RT option (see Section 21.24) and any other options the client is interested in receiving. The client **MAY** additionally include instances of those options that are identified in the Option Request option, with data values as hints to the server about parameter values the client would like to have returned.

The client includes a Reconfigure Accept option (see Section 21.20) if the client is willing to accept Reconfigure messages from the server.

The client **MUST NOT** include any other options in the Solicit message, except as specifically allowed in the definition of individual options.

The first Solicit message from the client on the interface **SHOULD** be delayed by a random amount of time between 0 and SOL_MAX_DELAY. This random delay helps desynchronize clients that start a DHCP session at the same time, such as after recovery from a power failure or after a router outage after seeing that DHCP is available in Router Advertisement messages (see Section 4.2 of [RFC4861]).

The client transmits the message according to Section 15, using the following parameters:

- IRT: SOL_TIMEOUT
- MRT: SOL_MAX_RT
- MRC: 0
- MRD: 0

A client that wishes to use the Rapid Commit two-message exchange includes a Rapid Commit option (see Section 21.14) in its Solicit message. The client may receive a number of different replies from different servers. The client will make note of any valid Advertise messages that it receives. The client will discard any Reply messages that do not contain the Rapid Commit option.

Upon receipt of a valid Reply with the Rapid Commit option, the client processes the message as described in Section 18.2.10.

At the end of the first RT period, if no suitable Reply messages are received but the client has valid Advertise messages, then the client processes the Advertise as described in Section 18.2.9.

If the client subsequently receives a valid Reply message that includes a Rapid Commit option, it does one of the following:

- processes the Reply message as described in Section 18.2.10 and discards any Reply messages received in response to the Request message
- processes any Reply messages received in response to the Request message and discards the Reply message that includes the Rapid Commit option

If the client is waiting for an Advertise message, the mechanism described in Section 15 is modified as follows for use in the transmission of Solicit messages. The message exchange is not terminated by the receipt of an Advertise before the first RT has elapsed. Rather, the client collects valid Advertise messages until the first RT has elapsed. Also, the first RT **MUST** be selected to be strictly greater than IRT by choosing RAND to be strictly greater than 0.

A client **MUST** collect valid Advertise messages for the first RT seconds, unless it receives a valid Advertise message with a preference value of 255. The preference value is carried in the Preference option (see Section 21.8). Any valid Advertise that does not include a Preference option is considered to have a preference value of 0. If the client receives a valid Advertise message that includes a Preference option with a preference value of 255, the client immediately begins a client-initiated message exchange (as described in Section 18.2.2) by sending a Request message to the server from which the Advertise message was received. If the client receives a valid Advertise message that does not include a Preference option with a preference value of 255, the client continues to wait until the first RT elapses. If the first RT elapses and the client has received a valid Advertise message, the client **SHOULD** continue with a client-initiated message exchange by sending a Request message.

If the client does not receive any valid Advertise messages before the first RT has elapsed, it then applies the retransmission mechanism described in Section 15. The client terminates the retransmission process as soon as it receives any valid Advertise message, and the client acts on the received Advertise message without waiting for any additional Advertise messages.

A DHCP client **SHOULD** choose MRC and MRD values of 0. If the DHCP client is configured with either MRC or MRD set to a value other than 0, it **MUST** stop trying to configure the interface if the message exchange fails. After the DHCP client stops trying to configure the interface, it **SHOULD** restart the reconfiguration process after some external event, such as user input, system restart, or when the client is attached to a new link.

18.2.2. Creation and Transmission of Request Messages

The client uses a Request message to populate IAs with leases and obtain other configuration information. The client includes one or more IA options in the Request message. The server then returns leases and other information about the IAs to the client in IA options in a Reply message.

The client sets the "msg-type" field to REQUEST. The client generates a transaction ID and inserts this value in the "transaction-id" field.

The client **MUST** include the identifier of the destination server in a Server Identifier option (see Section 21.3).

The client **MUST** include a Client Identifier option (see Section 21.2) to identify itself to the server. The client adds any other appropriate options, including one or more IA options.

The client **MUST** include an Elapsed Time option (see Section 21.9) to indicate how long the client has been trying to complete the current DHCP message exchange.

The client **MUST** include an Option Request option (see Section 21.7) to request the SOL_MAX_RT option (see Section 21.24) and any other options the client is interested in receiving. The client **MAY** additionally include instances of those options that are identified in the Option Request option, with data values as hints to the server about parameter values the client would like to have returned.

The client includes a Reconfigure Accept option (see Section 21.20) if the client is willing to accept Reconfigure messages from the server.

The client transmits the message according to Section 15, using the following parameters:

- IRT: REQ_TIMEOUT
- MRT: REQ_MAX_RT
- MRC: REQ_MAX_RC
- MRD: 0

If the message exchange fails, the client takes an action based on the client's local policy. Examples of actions the client might take include the following:

- Select another server from a list of servers known to the client -- for example, servers that responded with an Advertise message.
- Initiate the server discovery process described in Section 18.
- Terminate the configuration process and report failure.

18.2.3. Creation and Transmission of Confirm Messages

The client uses a Confirm message when it has only addresses (no delegated prefixes) assigned by a DHCP server to determine if it is still connected to the same link when the client detects a change in network information as described in Section 18.2.12.

The client sets the "msg-type" field to CONFIRM. The client generates a transaction ID and inserts this value in the "transaction-id" field.

The client **MUST** include a Client Identifier option (see Section 21.2) to identify itself to the server.

The client **MUST** include an Elapsed Time option (see Section 21.9) to indicate how long the client has been trying to complete the current DHCP message exchange.

The client includes IA options for all of the IAs assigned to the interface for which the Confirm message is being sent. The IA options include all of the addresses the client currently has associated with those IAs. The client **SHOULD** set the T1 and T2 fields in any IA_NA options (see Section 21.4) and the preferred-lifetime and valid-lifetime fields in the IA Address options (see Section 21.6) to 0, as the server will ignore these fields.

The first Confirm message from the client on the interface **MUST** be delayed by a random amount of time between 0 and CNF_MAX_DELAY. The client transmits the message according to Section 15, using the following parameters:

- IRT: CNF_TIMEOUT
- MRT: CNF_MAX_RT
- MRC: 0
- MRD: CNF_MAX_RD

If the client receives no responses before the message transmission process terminates, as described in Section 15, the client **SHOULD** continue to use any leases, using the last known lifetimes for those leases, and **SHOULD** continue to use any other previously obtained configuration parameters.

18.2.4. Creation and Transmission of Renew Messages

To extend the preferred and valid lifetimes for the leases assigned to the IAs and obtain new addresses or delegated prefixes for IAs, the client sends a Renew message to the server from which the leases were obtained; the Renew message includes IA options for the IAs whose lease lifetimes are to be extended. The client includes IA Address options (see Section 21.6) within IA_NA (see Section 21.4) options for the addresses assigned to the IAs. The client includes IA Prefix options (see Section 21.22) within IA_PD options (see Section 21.21) for the delegated prefixes assigned to the IAs.

The server controls the time at which the client should contact the server to extend the lifetimes on assigned leases through the T1 and T2 values assigned to an IA. However, as the client **SHOULD** renew/rebind all IAs from the server at the same time, the client **MUST** select T1 and T2

times from all IA options that will guarantee that the client initiates transmissions of Renew/Rebind messages not later than at the T1/T2 times associated with any of the client's bindings (earliest T1/T2).

At time T1, the client initiates a Renew/Reply message exchange to extend the lifetimes on any leases in the IA.

A client **MUST** also initiate a Renew/Reply message exchange before time T1 if the client's link-local address used in previous interactions with the server is no longer valid and it is willing to receive Reconfigure messages. This updates the server's information so it is able to continue to communicate with the client (either directly or via Relay-reply messages).

If T1 or T2 had been set to 0 by the server (for an IA_NA or IA_PD) in a previous Reply, the client may, at its discretion, send a Renew or Rebind message, respectively. The client **MUST** follow the rules defined in Section 14.2.

The client sets the "msg-type" field to RENEW. The client generates a transaction ID and inserts this value in the "transaction-id" field.

The client **MUST** include a Server Identifier option (see Section 21.3) in the Renew message, identifying the server with which the client most recently communicated.

The client **MUST** include a Client Identifier option (see Section 21.2) to identify itself to the server. The client adds any appropriate options, including one or more IA options.

The client **MUST** include an Elapsed Time option (see Section 21.9) to indicate how long the client has been trying to complete the current DHCP message exchange.

For IAs to which leases have been assigned, the client includes a corresponding IA option containing an IA Address option for each address assigned to the IA and an IA Prefix option for each prefix assigned to the IA. The client **MUST NOT** include addresses and prefixes in any IA option that the client did not obtain from the server or that are no longer valid (that have a valid lifetime of 0).

The client **MAY** include an IA option for each binding it desires but has been unable to obtain. In this case, if the client includes the IA_PD option to request prefix delegation, the client **MAY** include the IA Prefix option encapsulated within the IA_PD option, with the "IPv6-prefix" field set to 0 and the "prefix-length" field set to the desired length of the prefix to be delegated. The server **MAY** use this value as a hint for the prefix length. The client **SHOULD NOT** include an IA Prefix option with the "IPv6-prefix" field set to 0 unless it is supplying a hint for the prefix length.

The client includes an Option Request option (see Section 21.7) to request the SOL_MAX_RT option (see Section 21.24) and any other options the client is interested in receiving. The client **MAY** include options with data values as hints to the server about parameter values the client would like to have returned.

The client transmits the message according to Section 15, using the following parameters:

- IRT: REN_TIMEOUT

- MRT: REN_MAX_RT
- MRC: 0
- MRD: Remaining time until earliest T2

The message exchange is terminated when the earliest time T2 is reached. While the client is responding to a Reconfigure, the client ignores and discards any additional Reconfigure messages it may receive.

The message exchange is terminated when the earliest time T2 is reached, at which point the client begins the Rebind message exchange (see Section 18.2.5).

18.2.5. Creation and Transmission of Rebind Messages

At time T2 (which will only be reached if the server to which the Renew message was sent starting at time T1 has not responded), the client initiates a Rebind/Reply message exchange with any available server.

A Rebind is also used to verify delegated prefix bindings but with different retransmission parameters as described in Section 18.2.3.

The client constructs the Rebind message as described in Section 18.2.4, with the following differences:

- The client sets the "msg-type" field to REBIND.
- The client does not include the Server Identifier option (see Section 21.3) in the Rebind message.

The client transmits the message according to Section 15, using the following parameters:

- IRT: REB_TIMEOUT
- MRT: REB_MAX_RT
- MRC: 0
- MRD: Remaining time until valid lifetimes of all leases in all IAs have expired

If all leases for an IA have expired, the client may choose to include this IA in subsequent Rebind messages to indicate that the client is interested in assignment of the leases to this IA.

The message exchange is terminated when the valid lifetimes of all leases across all IAs have expired, at which time the client uses the Solicit message to locate a new DHCP server and sends a Request for the expired IAs to the new server. If the terminated Rebind exchange was initiated as a result of receiving a Reconfigure message, the client terminates the reconfigure process and resumes as if the Reconfigure message had not been received.

18.2.6. Creation and Transmission of Information-request Messages

The client uses an Information-request message to obtain configuration information without requesting addresses and/or delegated prefixes to be assigned.

The client sets the "msg-type" field to INFORMATION-REQUEST. The client generates a transaction ID and inserts this value in the "transaction-id" field.

The client **SHOULD** include a Client Identifier option (see Section 21.2) to identify itself to the server (however, see Section 4.3.1 of [RFC7844] for reasons why a client may not want to include this option). If the client does not include a Client Identifier option, the server will not be able to return any client-specific options to the client, or the server may choose not to respond to the message at all.

The client **MUST** include an Elapsed Time option (see Section 21.9) to indicate how long the client has been trying to complete the current DHCP message exchange.

The client **MUST** include an Option Request option (see Section 21.7) to request the INF_MAX_RT option (see Section 21.25), the Information Refresh Time option (see Section 21.23), and any other options the client is interested in receiving. The client **MAY** include options with data values as hints to the server about parameter values the client would like to have returned.

When responding to a Reconfigure, the client **MUST** include a Server Identifier option (see Section 21.3) with the identifier from the Reconfigure message to which the client is responding.

The first Information-request message from the client on the interface **MUST** be delayed by a random amount of time between 0 and INF_MAX_DELAY. The client transmits the message according to Section 15, using the following parameters:

- IRT: INF_TIMEOUT
- MRT: INF_MAX_RT
- MRC: 0
- MRD: 0

18.2.7. Creation and Transmission of Release Messages

To release one or more leases, a client sends a Release message to the server.

The client sets the "msg-type" field to RELEASE. The client generates a transaction ID and places this value in the "transaction-id" field.

The client **MUST** include a Server Identifier option (see Section 21.3) in the Renew message, identifying the server that allocated the lease(s).

The client **MUST** include a Client Identifier option (see Section 21.2) to identify itself to the server.

The client **MUST** include an Elapsed Time option (see Section 21.9) to indicate how long the client has been trying to complete the current DHCP message exchange.

The client includes options containing the IAs for the leases it is releasing in the "options" field. The leases to be released **MUST** be included in the IAs. Any leases for the IAs the client wishes to continue to use **MUST NOT** be added to the IAs.

The client **MUST** stop using all of the leases being released before the client begins the Release message exchange process. For an address, this means the address **MUST** have been removed from the interface. For a delegated prefix, this means the prefix **MUST** have been advertised with a Preferred Lifetime and a Valid Lifetime of 0 in a Router Advertisement message as described in part (e) of Section 5.5.3 of [RFC4862]; also see requirement L-13 in Section 4.3 of [RFC7084].

The client **MUST NOT** use any of the addresses it is releasing as the source address in the Release message or in any subsequently transmitted message.

Because Release messages may be lost, the client should retransmit the Release if no Reply is received. However, there are scenarios where the client may not wish to wait for the normal retransmission timeout before giving up (e.g., on power down). Implementations **SHOULD** retransmit one or more times but **MAY** choose to terminate the retransmission procedure early.

The client transmits the message according to Section 15, using the following parameters:

- IRT: REL_TIMEOUT
- MRT: 0
- MRC: REL_MAX_RC
- MRD: 0

If leases are released but the Reply from a DHCP server is lost, the client will retransmit the Release message, and the server may respond with a Reply indicating a status of NoBinding. Therefore, the client does not treat a Reply message with a status of NoBinding in a Release message exchange as if it indicates an error.

Note that if the client fails to release the lease, each lease assigned to the IA will be reclaimed by the server when the valid lifetime of that lease expires.

18.2.8. Creation and Transmission of Decline Messages

If a client detects that one or more addresses assigned to it by a server are already in use by another node, the client sends a Decline message to the server to inform it that the address is suspect.

The Decline message is not used in prefix delegation; thus, the client **MUST NOT** include IA_PD options (see Section 21.21) in the Decline message.

The client sets the "msg-type" field to DECLINE. The client generates a transaction ID and places this value in the "transaction-id" field.

The client **MUST** include a Server Identifier option (see Section 21.3) in the Decline message, identifying the server that allocated the lease(s).

The client **MUST** include a Client Identifier option (see Section 21.2) to identify itself to the server.

The client **MUST** include an Elapsed Time option (see Section 21.9) to indicate how long the client has been trying to complete the current DHCP message exchange.

The client includes options containing the IAs for the addresses it is declining in the "options" field. The addresses to be declined **MUST** be included in the IAs. Any addresses for the IAs the client wishes to continue to use should not be added to the IAs.

The client **MUST NOT** use any of the addresses it is declining as the source address in the Decline message or in any subsequently transmitted message.

The client transmits the message according to Section 15, using the following parameters:

- IRT: DEC_TIMEOUT
- MRT: 0
- MRC: DEC_MAX_RC
- MRD: 0

If addresses are declined but the Reply from a DHCP server is lost, the client will retransmit the Decline message, and the server may respond with a Reply indicating a status of NoBinding. Therefore, the client does not treat a Reply message with a status of NoBinding in a Decline message exchange as if it indicates an error.

The client **SHOULD NOT** send a Release message for other bindings it may have received just because it sent a Decline message. The client **SHOULD** retain the non-conflicting bindings. The client **SHOULD** treat the failure to acquire a binding (due to the conflict) as equivalent to not having received the binding, insofar as how it behaves when sending Renew and Rebind messages.

18.2.9. Receipt of Advertise Messages

Upon receipt of one or more valid Advertise messages, the client selects one or more Advertise messages based upon the following criteria.

- Those Advertise messages with the highest server preference value **SHOULD** be preferred over all other Advertise messages. The client **MAY** choose a less preferred server if that server has a better set of advertised parameters, such as the available set of IAs, as well as the set of other configuration options advertised.
- Within a group of Advertise messages with the same server preference value, a client **MAY** select those servers whose Advertise messages advertise information of interest to the client.

Once a client has selected Advertise message(s), the client will typically store information about each server, such as the server preference value, addresses advertised, when the advertisement was received, and so on.

In practice, this means that the client will maintain independent per-IA state machines for each selected server.

If the client needs to select an alternate server in the case that a chosen server does not respond, the client chooses the next server according to the criteria given above.

The client **MUST** process any SOL_MAX_RT option (see Section 21.24) and INF_MAX_RT option (see Section 21.25) present in an Advertise message, even if the message contains a Status Code option (see Section 21.13) indicating a failure, and the Advertise message will be discarded by the client. A client **SHOULD** only update its SOL_MAX_RT and INF_MAX_RT values if all received Advertise messages that contained the corresponding option specified the same value; otherwise, it should use the default value (see Section 7.6).

The client **MUST** ignore any Advertise message that contains no addresses (IA Address options (see Section 21.6) encapsulated in IA_NA options (see Section 21.4)) and no delegated prefixes (IA Prefix options (see Section 21.22) encapsulated in IA_PD options (see Section 21.21)), with the exception that the client:

- **MUST** process an included SOL_MAX_RT option and
- **MUST** process an included INF_MAX_RT option.

A client can record in an activity log or display to the user any associated status message(s).

The client ignoring an Advertise message **MUST NOT** restart the Solicit retransmission timer.

18.2.10. Receipt of Reply Messages

Upon the receipt of a valid Reply message in response to a Solicit with a Rapid Commit option (see Section 21.14), Request, Confirm, Renew, Rebind, or Information-request message, the client extracts the top-level Status Code option (see Section 21.13) if present.

The client **MUST** process any SOL_MAX_RT option (see Section 21.24) and INF_MAX_RT option (see Section 21.25) present in a Reply message, even if the message contains a Status Code option indicating a failure.

If the client receives a Reply message with a status code of UnspecFail, the server is indicating that it was unable to process the client's message due to an unspecified failure condition. If the client retransmits the original message to the same server to retry the desired operation, the client **MUST** limit the rate at which it retransmits the message and limit the duration of the time during which it retransmits the message (see Section 14.1).

Otherwise (no status code or another status code), the client processes the Reply as described below based on the original message for which the Reply was received.

The client **MAY** choose to report any status code or message from the Status Code option in the Reply message.

The topic of revoking previously assigned options is discussed in Section 18.2.10.5.

When a client receives a requested option that has an updated value from what was previously received, the client **SHOULD** make use of that updated value as soon as possible for its configuration information.

18.2.10.1. Reply for Solicit (with Rapid Commit), Request, Renew, or Rebind

If the client receives a NotOnLink status from the server in response to a Solicit (with a Rapid Commit option; see Section 21.14) or a Request, the client can either reissue the message without specifying any addresses or restart the DHCP server discovery process (see Section 18 or Section 18.2.13).

If the Reply was received in response to a Solicit (with a Rapid Commit option), Request, Renew, or Rebind message, the client updates the information it has recorded about IAs from the IA options contained in the Reply message:

- Calculate T1 and T2 times (based on T1 and T2 values sent in the message and the message reception time), if appropriate for the IA type.
- Add any new leases in the IA option to the IA as recorded by the client.
- Update lifetimes for any leases in the IA option that the client already has recorded in the IA.
- Discard any leases from the IA, as recorded by the client, that have a valid lifetime of 0 in the IA Address or IA Prefix option.
- Leave unchanged any information about leases the client has recorded in the IA but that were not included in the IA from the server.

If the client can operate with the addresses and/or prefixes obtained from the server:

- The client uses the addresses, delegated prefixes, and other information from any IAs that do not contain a Status Code option with the NoAddrsAvail or NoPrefixAvail status code. The client **MAY** include the IAs for which it received the NoAddrsAvail or NoPrefixAvail status code, with no addresses or prefixes, in subsequent Renew and Rebind messages sent to the server, to retry obtaining the addresses or prefixes for these IAs.
- The client **MUST** perform duplicate address detection as per Section 5.4 of [RFC4862], which does list some exceptions, on each of the received addresses in any IAs on which it has not performed duplicate address detection during processing of any of the previous Reply messages from the server. The client performs the duplicate address detection before using the received addresses for any traffic. If any of the addresses are found to be in use on the link, the client sends a Decline message to the server for those addresses as described in Section 18.2.8.
- For each assigned address that does not have any associated reachability information (see the definition of "on-link" in Section 2.1 of [RFC4861]), in order to avoid the problems described in [RFC4943], the client **MUST NOT** assume that any addresses are reachable on-link as a result of receiving an IA Address option (see Section 21.6). Addresses obtained from an IA Address option **MUST NOT** be used to form an implicit prefix with a length other than 128.
- For each delegated prefix, the client assigns a subnet to each of the links to which the associated interfaces are attached.

  When a client subnets a delegated prefix, it must assign additional bits to the prefix to generate unique, longer prefixes. For example, if the client in Figure 1 were delegated 2001:db8:0::/48, it might generate 2001:db8:0:1::/64 and 2001:db8:0:2::/64 for assignment to

the two links in the subscriber network. If the client were delegated 2001:db8:0::/48 and 2001:db8:5::/48, it might assign 2001:db8:0:1::/64 and 2001:db8:5:1::/64 to one of the links, and 2001:db8:0:2::/64 and 2001:db8:5:2::/64 for assignment to the other link.

If the client uses a delegated prefix to configure addresses on interfaces on itself or other nodes behind it, the preferred and valid lifetimes of those addresses **MUST** be no longer than the remaining preferred and valid lifetimes, respectively, for the delegated prefix at any time. In particular, if the delegated prefix or a prefix derived from it is advertised for stateless address autoconfiguration [RFC4862], the advertised preferred and valid lifetimes **MUST NOT** exceed the corresponding remaining lifetimes of the delegated prefix.

Management of the specific configuration information is detailed in the definition of each option in Section 21.

If the Reply message contains any IAs but the client finds no usable addresses and/or delegated prefixes in any of these IAs, the client may either try another server (perhaps restarting the DHCP server discovery process) or use the Information-request message to obtain other configuration information only.

When the client receives a Reply message in response to a Renew or Rebind message, the client:

- Sends a Request message to the server that responded if any of the IAs in the Reply message contain the NoBinding status code. The client places IA options in this message for all IAs. The client continues to use other bindings for which the server did not return an error.
- Sends a Renew/Rebind if any of the IAs are not in the Reply message, but as this likely indicates that the server that responded does not support that IA type, sending immediately is unlikely to produce a different result. Therefore, the client **MUST** rate-limit its transmissions (see Section 14.1) and **MAY** just wait for the normal retransmission time (as if the Reply message had not been received). The client continues to use other bindings for which the server did return information.
- Otherwise accepts the information in the IA.

18.2.10.2. Reply for Release and Decline

When the client receives a valid Reply message in response to a Release message, the client considers the Release event completed, regardless of the Status Code option (see Section 21.13) returned by the server.

When the client receives a valid Reply message in response to a Decline message, the client considers the Decline event completed, regardless of the Status Code option(s) returned by the server.

18.2.10.3. Reply for Confirm

If the client receives any Reply messages that indicate a status of Success (explicit or implicit), the client can use the addresses in the IA and ignore any messages that indicate a NotOnLink status. When the client only receives one or more Reply messages with the NotOnLink status in response to a Confirm message, the client performs DHCP server discovery as described in Section 18.

18.2.10.4. Reply for Information-request

Refer to Section 21.23 for details on how the Information Refresh Time option (whether or not present in the Reply) should be handled by the client.

18.2.10.5. Revoking Previously Provided Options

When a client receives a Reply for a Renew, Rebind, or Information-Request that does not include a requested configuration option that it previously received in a Reply, the client **SHOULD** stop using the previously received configuration information. In other words, the client should behave as if it never received this configuration option and return to the relevant default state. If there is no viable way to stop using the received configuration information, the values received/configured from the option **MAY** persist if there are no other sources for that data and they have no external impact. For example, a client that previously received a Client FQDN option (see [RFC4704]) and used it to set up its hostname is allowed to continue using it if there is no reasonable way for a node to unset its hostname and it has no external impact. As a counterexample, a client that previously received an NTP server address from the DHCP server and does not receive it anymore **MUST** stop using the configured NTP server address. The client **SHOULD** be open to other sources of the same configuration information. This behavior does not apply to any IA options; their processing is described in Section 18.2.10.1.

18.2.11. Receipt of Reconfigure Messages

A client receives Reconfigure messages sent to UDP port 546 on interfaces for which it has acquired configuration information through DHCP. These messages may be sent at any time. Since the results of a reconfiguration event may affect application-layer programs, the client **SHOULD** log these events and **MAY** notify these programs of the change through an implementation-specific interface.

The message **MUST** be dropped if it doesn't pass the validation, as explained in Section 16.11, particularly in cases where the authentication is missing or fails.

Upon receipt of a valid Reconfigure message, the client responds with a Renew message, a Rebind message, or an Information-request message as indicated by the Reconfigure Message option (see Section 21.19). The client ignores the "transaction-id" field in the received Reconfigure message. While the transaction is in progress, the client discards any Reconfigure messages it receives.

The Reconfigure message acts as a trigger that signals the client to complete a successful message exchange. Once the client has received a Reconfigure, the client proceeds with the message exchange (retransmitting the Renew, Rebind, or Information-request message if necessary); the client **MUST** ignore any additional Reconfigure messages until the exchange is complete.

Duplicate messages will be ignored because the client will begin the exchange after the receipt of the first Reconfigure. Retransmitted messages will either (1) trigger the exchange (if the first Reconfigure was not received by the client) or (2) be ignored. The server **MAY** discontinue retransmission of Reconfigure messages to the client once the server receives the Renew, Rebind, or Information-request message from the client.

It might be possible for a duplicate or retransmitted Reconfigure to be sufficiently delayed (and delivered out of order) that it arrives at the client after the exchange (initiated by the original Reconfigure) has been completed. In this case, the client would initiate a redundant exchange. The likelihood of delayed and out-of-order delivery is small enough to be ignored. The consequence of the redundant exchange is inefficiency rather than incorrect operation.

18.2.12. Refreshing Configuration Information

Whenever a client may have moved to a new link, the prefixes/addresses assigned to the interfaces on that link may no longer be appropriate for the link to which the client is attached. Examples of times when a client may have moved to a new link include the following:

- The client reboots (and has stable storage and persistent DHCP state).
- The client is reconnected to a link on which it has obtained leases.
- The client returns from sleep mode.
- The client changes access points (e.g., if using Wi-Fi technology).
- The client's network interface indicates a disconnection event followed by a connection event.

Specific algorithms for detecting network attachment changes are out of scope for this document. Two possible mechanisms for detecting situations where refreshing configuration information may be needed are defined in [RFC6059] and [RFC4957].

When the client detects that it may have moved to a new link and it has obtained addresses and no delegated prefixes from a server, the client **SHOULD** initiate a Confirm/Reply message exchange. The client **MUST** include any IAs assigned to the interface that may have moved to a new link, along with the addresses associated with those IAs, in its Confirm message. Any responding servers will indicate whether those addresses are appropriate for the link to which the client is attached with the status in the Reply message it returns to the client.

If the client has any valid delegated prefixes obtained from the DHCP server, the client **MUST** initiate a Rebind/Reply message exchange as described in Section 18.2.5, with the exception that the retransmission parameters should be set as for the Confirm message (see Section 18.2.3). The client includes IA_NAs and IA_PDs, along with the associated leases, in its Rebind message.

If the client has only obtained network information using Information-request/Reply message exchanges, the client **MUST** initiate an Information-request/Reply message exchange as described in Section 18.2.6.

If the client has not detected having moved to a new link but has detected a significant change regarding the prefixes available on the link, the client **SHOULD** initiate one of the Renew/Reply, Confirm/Reply, or Information-request/Reply exchanges. A change is considered significant when one or more on-link prefixes are added and/or one or more existing on-link prefixes are deprecated. The reason for this is that such a significant change may indicate a configuration change at the server. However, a client **MUST** rate-limit such initiation attempts to avoid flooding a server with requests when there are link issues (for example, only doing one of these at most every 30 seconds).

The above selection of an exchange to initiate depends on the client's current state:

- If the client has any valid delegated prefixes obtained from the server, it sends Renew (as if the T1 time expired) as described in Section 18.2.4.
- Else, if the client obtained an address(es) from the server, it sends Confirm as described in Section 18.2.3.
- Else, if only network information was obtained from the server, it sends an Information-request as described in Section 18.2.6.

18.2.13. Restarting Server Discovery Process

Whenever a client restarts the DHCP server discovery process or selects an alternate server as described in Section 18.2.9, the client **SHOULD** stop using any addresses and delegated prefixes for which it has bindings (see Section 18.2.7) and, if possible, any other configuration information it previously received. The client **SHOULD** also try to obtain new bindings and other configuration information from a new server for the same interface. This facilitates the client using a single state machine for all bindings.

18.3. Server Behavior

For this discussion, the server is assumed to have been configured in an implementation-specific manner with configurations of interest to clients.

A server sends an Advertise message in response to each valid Solicit message it receives to announce the availability of the server to the client.

In most cases, the server will send a Reply in response to Request, Confirm, Renew, Rebind, Decline, Release, and Information-request messages sent by a client. The server will also send a Reply in response to a Solicit with a Rapid Commit option (see Section 21.14) when the server is configured to respond with committed lease assignments.

These Advertise and Reply messages **MUST** always contain the Server Identifier option (see Section 21.3) containing the server's DUID and the Client Identifier option (see Section 21.2) from the client message if one was present.

In most response messages, the server includes options containing configuration information for the client. The server must be aware of the recommendations on message sizes and the use of fragmentation as discussed in Section 5 of [RFC8200]. If the client included an Option Request option (see Section 21.7) in its message, the server includes options in the response message containing configuration parameters for all of the options identified in the Option Request option that the server has been configured to return to the client. The server **MAY** return additional options to the client if it has been configured to do so.

Any message sent from a client may arrive at the server encapsulated in one or more Relay-forward messages. The server **MUST** use the received message to construct the proper Relay-reply message to allow the response to the received message to be relayed through the same relay agents (in reverse order) as the original client message; see Section 19.3 for more details. The server may also need to record this information with each client in case it is needed to send a Reconfigure message at a later time, unless the server has been configured with addresses that can be used to send Reconfigure messages directly to the client (see Section 18.3.11). Note that servers that support leasequery [RFC5007] also need to record this information.

By sending Reconfigure messages, the server **MAY** initiate a configuration exchange to cause DHCP clients to obtain new addresses, prefixes, and other configuration information. For example, an administrator may use a server-initiated configuration exchange when links in the DHCP domain are to be renumbered or when other configuration options are updated, perhaps because servers are moved, added, or removed.

When a client receives a Reconfigure message from the server, the client initiates sending a Renew, Rebind, or Information-request message as indicated by msg-type in the Reconfigure Message option (see Section 21.19). The server sends IAs and/or other configuration information to the client in a Reply message. The server **MAY** include options containing the IAs and new values for other configuration parameters in the Reply message, even if those IAs and parameters were not requested in the client's message.

18.3.1. Receipt of Solicit Messages

The server determines the information about the client and its location as described in Section 13 and checks its administrative policy about responding to the client. If the server is not permitted to respond to the client, the server discards the Solicit message. For example, if the administrative policy for the server is that it may only respond to a client that is willing to accept a Reconfigure message, if the client does not include a Reconfigure Accept option (see Section 21.20) in the Solicit message, the server discards the Solicit message.

If (1) the server is permitted to respond to the client, (2) the client has not included a Rapid Commit option (see Section 21.14) in the Solicit message, or (3) the server has not been configured to respond with committed assignments of leases and other resources, the server sends an Advertise message to the client as described in Section 18.3.9.

If the client has included a Rapid Commit option in the Solicit message and the server has been configured to respond with committed assignments of leases and other resources, the server responds to the Solicit with a Reply message. The server produces the Reply message as though it had received a Request message as described in Section 18.3.2. The server transmits the Reply

message as described in Section 18.3.10. The server **MUST** commit the assignment of any addresses and delegated prefixes or other configuration information before sending a Reply message to a client. In this case, the server includes a Rapid Commit option in the Reply message to indicate that the Reply is in response to a Solicit message.

DISCUSSION:

- When using the Solicit/Reply message exchange, the server commits the assignment of any leases before sending the Reply message. The client can assume that it has been assigned the leases in the Reply message and does not need to send a Request message for those leases.
- Typically, servers that are configured to use the Solicit/Reply message exchange will be deployed so that only one server will respond to a Solicit message. If more than one server responds, the client will only use the leases from one of the servers, while the leases from the other servers will be committed to the client but not used by the client.

18.3.2. Receipt of Request Messages

When the server receives a valid Request message, the server creates the bindings for that client according to the server's policy and configuration information and records the IAs and other information requested by the client.

The server constructs a Reply message by setting the "msg-type" field to REPLY and copying the transaction ID from the Request message into the "transaction-id" field.

The server **MUST** include in the Reply message a Server Identifier option (see Section 21.3) containing the server's DUID and the Client Identifier option (see Section 21.2) from the Request message.

The server examines all IAs in the message from the client.

For each IA_NA option (see Section 21.4) in the Request message, the server checks if the prefixes of included addresses are appropriate for the link to which the client is connected. If any of the prefixes of the included addresses are not appropriate for the link to which the client is connected, the server **MUST** return the IA to the client with a Status Code option (see Section 21.13) with the value NotOnLink. If the server does not send the NotOnLink status code but it cannot assign any IP addresses to an IA, the server **MUST** return the IA option in the Reply message with no addresses in the IA and a Status Code option containing status code NoAddrsAvail in the IA.

For any IA_PD option (see Section 21.21) in the Request message to which the server cannot assign any delegated prefixes, the server **MUST** return the IA_PD option in the Reply message with no prefixes in the IA_PD and with a Status Code option containing status code NoPrefixAvail in the IA_PD.

The server **MAY** assign different addresses and/or delegated prefixes to an IA than those included within the IA of the client's Request message.

For all IAs to which the server can assign addresses or delegated prefixes, the server includes the IAs with addresses (for IA_NAs), prefixes (for IA_PDs), and other configuration parameters and records the IA as a new client binding. The server **MUST NOT** include any addresses or delegated prefixes in the IA that the server does not assign to the client.

The T1/T2 times set in each applicable IA option for a Reply **MUST** be the same values across all IAs. The server **MUST** determine the T1/T2 times across all of the applicable client's bindings in the Reply. This facilitates the client being able to renew all of the bindings at the same time.

The server **SHOULD** include a Reconfigure Accept option (see Section 21.20) if the server policy enables the reconfigure mechanism and the client supports it. Currently, sending this option in a Reply is technically redundant, as the use of the reconfiguration mechanism requires authentication; at present, the only defined mechanism is RKAP (see Section 20.4), and the presence of the reconfigure key signals support for the acceptance of Reconfigure messages. However, there may be better security mechanisms defined in the future that would cause RKAP to not be used anymore.

The server includes other options containing configuration information to be returned to the client as described in Section 18.3.

If the server finds that the client has included an IA in the Request message for which the server already has a binding that associates the IA with the client, the server sends a Reply message with existing bindings, possibly with updated lifetimes. The server may update the bindings according to its local policies, but the server **SHOULD** generate the response again and not simply retransmit previously sent information, even if the "transaction-id" field value matches a previous transmission. The server **MUST NOT** cache its responses.

DISCUSSION:

- Cached replies are bad because lifetimes need to be updated (either decrease the timers by the amount of time elapsed since the original transmission or keep the lifetime values and update the lease information in the server's database). Also, if the message uses any security protection (such as the Replay Detection Method (RDM), as described in Section 20.3), its value must be updated. Additionally, any digests must be updated. Given all of the above, caching replies is far more complex than simply sending the same buffer as before, and it is easy to miss some of those steps.

18.3.3. Receipt of Confirm Messages

When the server receives a Confirm message, the server determines whether the addresses in the Confirm message are appropriate for the link to which the client is attached. If all of the addresses in the Confirm message pass this test, the server returns a status of Success. If any of the addresses do not pass this test, the server returns a status of NotOnLink. If the server is unable to perform this test (for example, the server does not have information about prefixes on the link to which the client is connected) or there were no addresses in any of the IAs sent by the client, the server **MUST NOT** send a Reply to the client.

The server ignores the T1 and T2 fields in the IA options and the preferred-lifetime and valid-lifetime fields in the IA Address options (see Section 21.6).

The server constructs a Reply message by setting the "msg-type" field to REPLY and copying the transaction ID from the Confirm message into the "transaction-id" field.

The server **MUST** include in the Reply message a Server Identifier option (see Section 21.3) containing the server's DUID and the Client Identifier option (see Section 21.2) from the Confirm message. The server includes a Status Code option (see Section 21.13) indicating the status of the Confirm message.

18.3.4. Receipt of Renew Messages

For each IA in the Renew message from a client, the server locates the client's binding and verifies that the information in the IA from the client matches the information stored for that client.

If the server finds the client entry for the IA, the server sends the IA back to the client with new lifetimes and, if applicable, T1/T2 times. If the server is unable to extend the lifetimes of an address or delegated prefix in the IA, the server **MAY** choose not to include the IA Address option (see Section 21.6) for that address or IA Prefix option (see Section 21.22) for that delegated prefix. If the server chooses to include the IA Address or IA Prefix option for such an address or delegated prefix, the server **SHOULD** set T1 and T2 values to the valid lifetime for the IA option unless the server also includes other addresses or delegated prefixes that the server is able to extend for the IA. Setting T1 and T2 to values equal to the valid lifetime informs the client that the leases associated with said IA will not be extended, so there is no point in trying. Also, it avoids generating unnecessary traffic as the remaining lifetime approaches 0.

The server may choose to change the list of addresses or delegated prefixes and the lifetimes in IAs that are returned to the client.

If the server finds that any of the addresses in the IA are not appropriate for the link to which the client is attached, the server returns the address to the client with lifetimes of 0.

If the server finds that any of the delegated prefixes in the IA are not appropriate for the link to which the client is attached, the server returns the delegated prefix to the client with lifetimes of 0.

For each IA for which the server cannot find a client entry, the server has the following choices, depending on the server's policy and configuration information:

- If the server is configured to create new bindings as a result of processing Renew messages, the server **SHOULD** create a binding and return the IA with assigned addresses or delegated prefixes with lifetimes and, if applicable, T1/T2 times and other information requested by the client. If the client included the IA Prefix option within the IA_PD option (see Section 21.21) with a zero value in the "IPv6-prefix" field and a non-zero value in the "prefix-length" field, the server **MAY** use the "prefix-length" value as a hint for the length of the prefixes to be assigned (see [RFC8168] for further details on prefix-length hints).

- If the server is configured to create new bindings as a result of processing Renew messages but the server will not assign any leases to an IA, the server returns the IA option containing a Status Code option (see Section 21.13) with the NoAddrsAvail or NoPrefixAvail status code and a status message for a user.
- If the server does not support creation of new bindings for the client sending a Renew message or if this behavior is disabled according to the server's policy or configuration information, the server returns the IA option containing a Status Code option with the NoBinding status code and a status message for a user.

The server constructs a Reply message by setting the "msg-type" field to REPLY and copying the transaction ID from the Renew message into the "transaction-id" field.

The server **MUST** include in the Reply message a Server Identifier option (see Section 21.3) containing the server's DUID and the Client Identifier option (see Section 21.2) from the Renew message.

The server includes other options containing configuration information to be returned to the client as described in Section 18.3.

The server **MAY** include options containing the IAs and values for other configuration parameters, even if those parameters were not requested in the Renew message.

The T1/T2 values set in each applicable IA option for a Reply **MUST** be the same across all IAs. The server **MUST** determine the T1/T2 values across all of the applicable client's bindings in the Reply. This facilitates the client being able to renew all of the bindings at the same time.

18.3.5. Receipt of Rebind Messages

When the server receives a Rebind message that contains an IA option from a client, it locates the client's binding and verifies that the information in the IA from the client matches the information stored for that client.

If the server finds the client entry for the IA and the server determines that the addresses or delegated prefixes in the IA are appropriate for the link to which the client's interface is attached according to the server's explicit configuration information, the server **SHOULD** send the IA back to the client with new lifetimes and, if applicable, T1/T2 values. If the server is unable to extend the lifetimes of an address in the IA, the server **MAY** choose not to include the IA Address option (see Section 21.6) for this address. If the server is unable to extend the lifetimes of a delegated prefix in the IA, the server **MAY** choose not to include the IA Prefix option (see Section 21.22) for this prefix.

If the server finds that the client entry for the IA and any of the addresses or delegated prefixes are no longer appropriate for the link to which the client's interface is attached according to the server's explicit configuration information, the server returns those addresses or delegated prefixes to the client with lifetimes of 0.

If the server cannot find a client entry for the IA, the server checks if the IA contains addresses (for IA_NAs) or delegated prefixes (for IA_PDs). The server checks if the addresses and delegated prefixes are appropriate for the link to which the client's interface is attached according to the server's explicit configuration information. For any address that is not appropriate for the link to which the client's interface is attached, the server **MAY** include the IA Address option with lifetimes of 0. For any delegated prefix that is not appropriate for the link to which the client's interface is attached, the server **MAY** include the IA Prefix option with lifetimes of 0. The Reply with lifetimes of 0 constitutes an explicit notification to the client that the specific addresses and delegated prefixes are no longer valid and **MUST NOT** be used by the client. If the server chooses to not include any IAs containing IA Address or IA Prefix options with lifetimes of 0 and the server does not include any other IAs with leases and/or status codes, the server does not send a Reply message. In this situation, the server discards the Rebind message.

Otherwise, for each IA for which the server cannot find a client entry, the server has the following choices, depending on the server's policy and configuration information:

- If the server is configured to create new bindings as a result of processing Rebind messages (also see the note below about the Rapid Commit option (Section 21.14)), the server **SHOULD** create a binding and return the IA with allocated leases with lifetimes and, if applicable, T1/T2 values and other information requested by the client. The server **MUST NOT** return any addresses or delegated prefixes in the IA that the server does not assign to the client.
- If the server is configured to create new bindings as a result of processing Rebind messages but the server will not assign any leases to an IA, the server returns the IA option containing a Status Code option (see Section 21.13) with the NoAddrsAvail or NoPrefixAvail status code and a status message for a user.
- If the server does not support creation of new bindings for the client sending a Rebind message or if this behavior is disabled according to the server's policy or configuration information, the server returns the IA option containing a Status Code option with the NoBinding status code and a status message for a user.

When the server creates new bindings for the IA, it is possible that other servers also create bindings as a result of receiving the same Rebind message; see the "DISCUSSION" text in Section 21.14. Therefore, the server **SHOULD** only create new bindings during processing of a Rebind message if the server is configured to respond with a Reply message to a Solicit message containing the Rapid Commit option.

The server constructs a Reply message by setting the "msg-type" field to REPLY and copying the transaction ID from the Rebind message into the "transaction-id" field.

The server **MUST** include in the Reply message a Server Identifier option (see Section 21.3) containing the server's DUID and the Client Identifier option (see Section 21.2) from the Rebind message.

The server includes other options containing configuration information to be returned to the client as described in Section 18.3.

The server **MAY** include options containing the IAs and values for other configuration parameters, even if those IAs and parameters were not requested in the Rebind message.

The T1 or T2 values set in each applicable IA option for a Reply **MUST** be the same values across all IAs. The server **MUST** determine the T1 or T2 values across all of the applicable client's bindings in the Reply. This facilitates the client being able to renew all of the bindings at the same time.

18.3.6. Receipt of Information-request Messages

When the server receives an Information-request message, the client is requesting configuration information that does not include the assignment of any leases. The server determines all configuration parameters appropriate to the client, based on the server configuration policies known to the server.

The server constructs a Reply message by setting the "msg-type" field to REPLY and copying the transaction ID from the Information-request message into the "transaction-id" field.

The server **MUST** include a Server Identifier option (see Section 21.3) containing the server's DUID in the Reply message. If the client included a Client Identifier option (see Section 21.2) in the Information-request message, the server copies that option to the Reply message.

The server includes options containing configuration information to be returned to the client as described in Section 18.3. The server **MAY** include additional options that were not requested by the client in the Information-request message.

If the Information-request message received from the client did not include a Client Identifier option, the server **SHOULD** respond with a Reply message containing any configuration parameters that are not determined by the client's identity. If the server chooses not to respond, the client may continue to retransmit the Information-request message indefinitely.

18.3.7. Receipt of Release Messages

The server constructs a Reply message by setting the "msg-type" field to REPLY and copying the transaction ID from the Release message into the "transaction-id" field.

Upon the receipt of a valid Release message, the server examines the IAs and the leases in the IAs for validity. If the IAs in the message are in a binding for the client and the leases in the IAs have been assigned by the server to those IAs, the server deletes the leases from the IAs and makes the leases available for assignment to other clients. The server ignores leases not assigned to the IAs, although it may choose to log an error.

After all the leases have been processed, the server generates a Reply message and includes a Status Code option (see Section 21.13) with the value Success, a Server Identifier option (see Section 21.3) with the server's DUID, and a Client Identifier option (see Section 21.2) with the client's DUID. For each IA in the Release message for which the server has no binding information, the server adds an IA option using the IAID from the Release message and includes a Status Code option with the value NoBinding in the IA option. No other options are included in the IA option.

A server may choose to retain a record of assigned leases and IAs after the lifetimes on the leases have expired to allow the server to reassign the previously assigned leases to a client.

18.3.8. Receipt of Decline Messages

Upon the receipt of a valid Decline message, the server examines the IAs and the addresses in the IAs for validity. If the IAs in the message are in a binding for the client and the addresses in the IAs have been assigned by the server to those IAs, the server deletes the addresses from the IAs. The server ignores addresses not assigned to the IAs (though it may choose to log an error if it finds such addresses).

The client has found any addresses in the Decline messages to be already in use on its link. Therefore, the server **SHOULD** mark the addresses declined by the client so that those addresses are not assigned to other clients and **MAY** choose to make a notification that addresses were declined. Local policy on the server determines when the addresses identified in a Decline message may be made available for assignment.

After all the addresses have been processed, the server generates a Reply message by setting the "msg-type" field to REPLY and copying the transaction ID from the Decline message into the "transaction-id" field. The server includes a Status Code option (see Section 21.13) with the value Success, a Server Identifier option (see Section 21.3) with the server's DUID, and a Client Identifier option (see Section 21.2) with the client's DUID. For each IA in the Decline message for which the server has no binding information, the server adds an IA option using the IAID from the Decline message and includes a Status Code option with the value NoBinding in the IA option. No other options are included in the IA option.

18.3.9. Creation of Advertise Messages

The server sets the "msg-type" field to ADVERTISE and copies the contents of the "transaction-id" field from the Solicit message received from the client to the Advertise message. The server includes its server identifier in a Server Identifier option (see Section 21.3) and copies the Client Identifier option (see Section 21.2) from the Solicit message into the Advertise message.

The server **MAY** add a Preference option (see Section 21.8) to carry the preference value for the Advertise message. The server implementation **SHOULD** allow the setting of a server preference value by the administrator. The server preference value **MUST** default to 0 unless otherwise configured by the server administrator.

The server includes a Reconfigure Accept option (see Section 21.20) if the server wants to indicate that it supports the Reconfigure mechanism. This information may be used by the client during the server selection process.

The server includes the options the server will return to the client in a subsequent Reply message. The information in these options may be used by the client in the selection of a server if the client receives more than one Advertise message. The server **MUST** include options in the Advertise message containing configuration parameters for all of the options identified in the Option Request option (see Section 21.7) in the Solicit message that the server has been configured to return to the client. If the Option Request option includes a container option, the

server **MUST** include all the options that are eligible to be encapsulated in the container. The Option Request option **MAY** be used to signal support for a feature even when that option is encapsulated, as in the case of the Prefix Exclude option [RFC6603]. In this case, special processing is required by the server. The server **MAY** return additional options to the client if it has been configured to do so.

The server **MUST** include IA options in the Advertise message containing any addresses and/or delegated prefixes that would be assigned to IAs contained in the Solicit message from the client. If the client has included addresses in the IA Address options (see Section 21.6) in the Solicit message, the server **MAY** use those addresses as hints about the addresses that the client would like to receive. If the client has included IA Prefix options (see Section 21.22), the server **MAY** use the prefix contained in the "IPv6-prefix" field and/or the prefix length contained in the "prefix-length" field as hints about the prefixes the client would like to receive. If the server is not going to assign an address or delegated prefix received as a hint in the Solicit message, the server **MUST NOT** include this address or delegated prefix in the Advertise message.

If the server will not assign any addresses to an IA_NA in subsequent Request messages from the client, the server **MUST** include the IA option in the Advertise message with no addresses in that IA and a Status Code option (see Section 21.13) encapsulated in the IA option containing status code NoAddrsAvail.

If the server will not assign any prefixes to an IA_PD in subsequent Request messages from the client, the server **MUST** include the IA_PD option (see Section 21.21) in the Advertise message with no prefixes in the IA_PD option and a Status Code option encapsulated in the IA_PD containing status code NoPrefixAvail.

Transmission of Advertise messages is described in the next section.

18.3.10. Transmission of Advertise and Reply Messages

If the original message was received directly by the server, the server unicasts the Advertise or Reply message directly to the client using the address in the source address field from the IP datagram in which the original message was received. The Advertise or Reply message **MUST** be unicast through the interface on which the original message was received.

If the original message was received in a Relay-forward message, the server constructs a Relay-reply message with the Reply message in the payload of a Relay Message option (see Section 21.10). If the Relay-forward messages included an Interface-Id option (see Section 21.18), the server copies that option to the Relay-reply message. The server unicasts the Relay-reply message directly to the relay agent using the address in the source address field from the IP datagram in which the Relay-forward message was received. See Section 19.3 for more details on the construction of Relay-reply messages.

18.3.11. Creation and Transmission of Reconfigure Messages

The server sets the "msg-type" field to RECONFIGURE and sets the "transaction-id" field to 0. The server includes a Server Identifier option (see Section 21.3) containing its DUID and a Client Identifier option (see Section 21.2) containing the client's DUID in the Reconfigure message.

Because of the risk of denial-of-service (DoS) attacks against DHCP clients, the use of a security mechanism is mandated in Reconfigure messages. The server **MUST** use DHCP authentication in the Reconfigure message (see Section 20.4).

The server **MUST** include a Reconfigure Message option (see Section 21.19) to select whether the client responds with a Renew message, a Rebind message, or an Information-request message.

The server **MUST NOT** include any other options in the Reconfigure message, except as specifically allowed in the definition of individual options.

A server sends each Reconfigure message to a single DHCP client, using an IPv6 unicast address of sufficient scope belonging to the DHCP client. If the server does not have an address to which it can send the Reconfigure message directly to the client, the server uses a Relay-reply message (as described in Section 19.3) to send the Reconfigure message to a relay agent that will relay the message to the client. The server may obtain the address of the client (and the appropriate relay agent, if required) through the information the server has about clients that have been in contact with the server (see Section 18.3) or through some external agent.

To reconfigure more than one client, the server unicasts a separate message to each client. The server may initiate the reconfiguration of multiple clients concurrently; for example, a server may send a Reconfigure message to additional clients while previous reconfiguration message exchanges are still in progress.

The Reconfigure message causes the client to initiate a Renew/Reply, Rebind/Reply, or Information-request/Reply message exchange with the server. The server interprets the receipt of a Renew, Rebind, or Information-request message (whichever was specified in the original Reconfigure message) from the client as satisfying the Reconfigure message request.

When transmitting the Reconfigure message, the server sets the retransmission time (RT) to REC_TIMEOUT. If the server does not receive a Renew, Rebind, or Information-request message from the client before the RT elapses, the server retransmits the Reconfigure message, doubles the RT value, and waits again. The server continues this process until REC_MAX_RC unsuccessful attempts have been made, at which point the server **SHOULD** abort the reconfigure process for that client.

Default and initial values for REC_TIMEOUT and REC_MAX_RC are documented in Section 7.6.

19. Relay Agent Behavior

The relay agent **SHOULD** be configured to use a list of destination addresses that includes unicast addresses. The list of destination addresses **MAY** include the All_DHCP_Servers multicast address or other addresses selected by the network administrator. If the relay agent has not been explicitly configured, it **MUST** use the All_DHCP_Servers multicast address as the default.

If the relay agent relays messages to the All_DHCP_Servers multicast address or other multicast addresses, it sets the Hop Limit field to 8.

If the relay agent receives a message other than Relay-forward and Relay-reply and the relay agent does not recognize its message type, it **MUST** forward the message as described in Section 19.1.1.

19.1. Relaying a Client Message or a Relay-forward Message

A relay agent relays both messages from clients and Relay-forward messages from other relay agents. When a relay agent receives a Relay-forward message, a recognized message type for which it is not the intended target, or an unrecognized message type, it constructs a new Relay-forward message. The relay agent copies the source address from the header of the IP datagram in which the message was received into the peer-address field of the Relay-forward message. The relay agent copies the received DHCP message (excluding any IP or UDP headers) into a Relay Message option (see Section 21.10) in the new message. The relay agent adds to the Relay-forward message any other options it is configured to include.

[RFC6221] defines a Lightweight DHCPv6 Relay Agent (LDRA) that allows relay agent information to be inserted by an access node that performs a link-layer bridging (i.e., non-routing) function.

19.1.1. Relaying a Message from a Client

If the relay agent received the message to be relayed from a client, the relay agent places a globally scoped unicast address (i.e., GUA or ULA) from a prefix assigned to the link on which the client should be assigned leases into the link-address field. If such an address is not available, the relay agent may set the link-address field to a link-local address from the interface on which the original message was received. This is not recommended, as it may require that additional information be provided in the server configuration. See Section 3.2 of [RFC7969] for a detailed discussion.

This address will be used by the server to determine the link from which the client should be assigned leases and other configuration information.

The hop-count value in the Relay-forward message is set to 0.

The relay **SHOULD** insert a Client Link-Layer Address option as described in [RFC6939].

If the relay agent cannot use the address in the link-address field to identify the interface through which the response to the client will be relayed, the relay agent **MUST** include an Interface-Id option (see Section 21.18) in the Relay-forward message. The server will include the Interface-Id option in its Relay-reply message. The relay agent sets the link-address field as described earlier in this subsection, regardless of whether the relay agent includes an Interface-Id option in the Relay-forward message.

19.1.2. Relaying a Message from a Relay Agent

If the message received by the relay agent is a Relay-forward message and the hop-count value in the message is greater than or equal to HOP_COUNT_LIMIT, the relay agent discards the received message.

The relay agent copies the source address from the IP datagram in which the message was received into the peer-address field in the Relay-forward message and sets the hop-count field to the value of the hop-count field in the received message incremented by 1.

If the source address from the IP datagram header of the received message is a globally scoped unicast address (i.e., GUA or ULA), the relay agent sets the link-address field to 0; otherwise, the relay agent sets the link-address field to a globally scoped unicast address (i.e., GUA or ULA) assigned to the interface on which the message was received or includes an Interface-Id option (see Section 21.18) to identify the interface on which the message was received.

19.1.3. Relay Agent Behavior with Prefix Delegation

A relay agent forwards messages containing prefix delegation options in the same way as it would relay addresses (i.e., per Sections 19.1.1 and 19.1.2).

If a server communicates with a client through a relay agent about delegated prefixes, the server may need a protocol or other out-of-band communication to configure routing information for delegated prefixes on any router through which the client may forward traffic.

19.2. Relaying a Relay-reply Message

The relay agent processes any options included in the Relay-reply message in addition to the Relay Message option (see Section 21.10).

The relay agent extracts the message from the Relay Message option and relays it to the address contained in the peer-address field of the Relay-reply message. Relay agents **MUST NOT** modify the message.

If the Relay-reply message includes an Interface-Id option (see Section 21.18), the relay agent relays the message from the server to the client on the link identified by the Interface-Id option. Otherwise, if the link-address field is not set to 0, the relay agent relays the message on the link identified by the link-address field.

If the relay agent receives a Relay-reply message, it **MUST** process the message as defined above, regardless of the type of message encapsulated in the Relay Message option.

19.3. Construction of Relay-reply Messages

A server uses a Relay-reply message to (1) return a response to a client if the original message from the client was relayed to the server in a Relay-forward message or (2) send a Reconfigure message to a client if the server does not have an address it can use to send the message directly to the client.

A response to the client **MUST** be relayed through the same relay agents as the original client message. The server causes this to happen by creating a Relay-reply message that includes a Relay Message option (see Section 21.10) containing the message for the next relay agent in the return path to the client. The contained Relay-reply message contains another Relay Message

option to be sent to the next relay agent, and so on. The server must record the contents of the peer-address fields in the received message so it can construct the appropriate Relay-reply message carrying the response from the server.

For example, if client C sent a message that was relayed by relay agent A to relay agent B and then to the server, the server would send the following Relay-reply message to relay agent B:

```
msg-type:       RELAY-REPL
hop-count:      1
link-address:   0
peer-address:   A
Relay Message option containing the following:
   msg-type:     RELAY-REPL
   hop-count:    0
   link-address: address from link to which C is attached
   peer-address: C
   Relay Message option: <response from server>
```

Figure 10: Relay-reply Example

When sending a Reconfigure message to a client through a relay agent, the server creates a Relay-reply message that includes a Relay Message option containing the Reconfigure message for the next relay agent in the return path to the client. The server sets the peer-address field in the Relay-reply message header to the address of the client and sets the link-address field as required by the relay agent to relay the Reconfigure message to the client. The server obtains the addresses of the client and the relay agent through prior interaction with the client or through some external mechanism.

19.4. Interaction Between Relay Agents and Servers

Each time a message is relayed by a relay agent towards a server, a new encapsulation level is added around the message. Each relay is allowed to insert additional options on the encapsulation level it added but **MUST NOT** change anything in the message being encapsulated. If there are multiple relays between a client and a server, multiple encapsulations are used. Although it makes message processing slightly more complex, it provides the major advantage of having a clear indication as to which relay inserted which option. The response message is expected to travel through the same relays, but in reverse order. Each time a response message is relayed back towards a client, one encapsulation level is removed.

In certain cases, relays can add one or more options. These options can be added for several reasons:

- First, relays can provide additional information about the client. That source of information is usually more trusted by a server administrator, as it comes from the network infrastructure rather than the client and cannot be easily spoofed. These options can be used by the server to determine its allocation policy.

- Second, a relay may need some information to send a response back to the client. Relay agents are expected to be stateless (not retain any state after a message has been processed). A relay agent may include the Interface-Id option (see Section 21.18), which will be echoed back in the response. It can include other options and ask the server to echo one or more of the options back in the response. These options can then be used by the relay agent to send the response back to the client, or for other needs. The client will never see these options. See [RFC4994] for details.
- Third, sometimes a relay is the best device to provide values for certain options. A relay can insert an option into the message being forwarded to the server and ask the server to pass that option back to the client. The client will receive that option. It should be noted that the server is the ultimate authority here, and -- depending on its configuration -- it may or may not send the option back to the client. See [RFC6422] for details.

For various reasons, servers may need to retain the relay information after the message processing is completed. One is a bulk leasequery mechanism that may ask for all addresses and/or prefixes that were assigned via a specific relay. A second is for the reconfigure mechanism. The server may choose to not send the Reconfigure message directly to the client but rather to send it via relays. This particular behavior is considered an implementation detail and is out of scope for this document.

20. Authentication of DHCP Messages

This document defines two security mechanisms for the authentication of DHCP messages: (1) authentication (and encryption) of messages sent between servers and relay agents using IPsec and (2) protection against misconfiguration of a client caused by a Reconfigure message sent by a malicious DHCP server.

20.1. Security of Messages Sent Between Servers and Relay Agents

Relay agents and servers that exchange messages can use IPsec as detailed in [RFC8213].

20.2. Summary of DHCP Authentication

Authentication of DHCP messages is accomplished through the use of the Authentication option (see Section 21.11). The authentication information carried in the Authentication option can be used to reliably identify the source of a DHCP message and to confirm that the contents of the DHCP message have not been tampered with.

The Authentication option provides a framework for multiple authentication protocols. One such protocol, RKAP, is defined in Section 20.4. Other protocols defined in the future will be specified in separate documents.

Any DHCP message **MUST NOT** include more than one Authentication option.

The protocol field in the Authentication option identifies the specific protocol used to generate the authentication information carried in the option. The algorithm field identifies a specific algorithm within the authentication protocol; for example, the algorithm field specifies the hash algorithm used to generate the Message Authentication Code (MAC) in the Authentication option. The RDM field specifies the type of replay detection used in the replay detection field.

20.3. Replay Detection

The RDM field of the Authentication option (see Section 21.11) determines the type of replay detection used in the replay detection field.

If the RDM field contains 0x00, the replay detection field **MUST** be set to the value of a strictly monotonically increasing 64-bit unsigned integer (modulo 2^64). Using this technique can reduce the danger of replay attacks. This method **MUST** be supported by all Authentication option protocols. One choice might be to use the 64-bit NTP timestamp format [RFC5905]).

A client that receives a message with the RDM field set to 0x00 **MUST** compare its replay detection field with the previous value sent by that same server (based on the Server Identifier option; see Section 21.3) and only accept the message if the received value is greater and record this as the new value. If this is the first time a client processes an Authentication option sent by a server, the client **MUST** record the replay detection value and skip the replay detection check.

Servers that support the reconfigure mechanism **MUST** ensure that the replay detection value is retained between restarts. Failing to do so may cause clients to refuse Reconfigure messages sent by the server, effectively rendering the reconfigure mechanism useless.

20.4. Reconfiguration Key Authentication Protocol (RKAP)

RKAP provides protection against misconfiguration of a client caused by a Reconfigure message sent by a malicious DHCP server. In this protocol, a DHCP server sends a reconfigure key to the client in the initial exchange of DHCP messages. The client records the reconfigure key for use in authenticating subsequent Reconfigure messages from that server. The server then includes a Hashed Message Authentication Code (HMAC) computed from the reconfigure key in subsequent Reconfigure messages.

Both the reconfigure key sent from the server to the client and the HMAC in subsequent Reconfigure messages are carried as the authentication information in an Authentication option (see Section 21.11). The format of the authentication information is defined in the following section.

RKAP is used (initiated by the server) only if the client and server have negotiated to use Reconfigure messages.

20.4.1. Use of the Authentication Option in RKAP

The following fields are set in an Authentication option (see Section 21.11) for RKAP:

- protocol: 3

- algorithm: 1
- RDM: 0

The format of the authentication information for RKAP is:

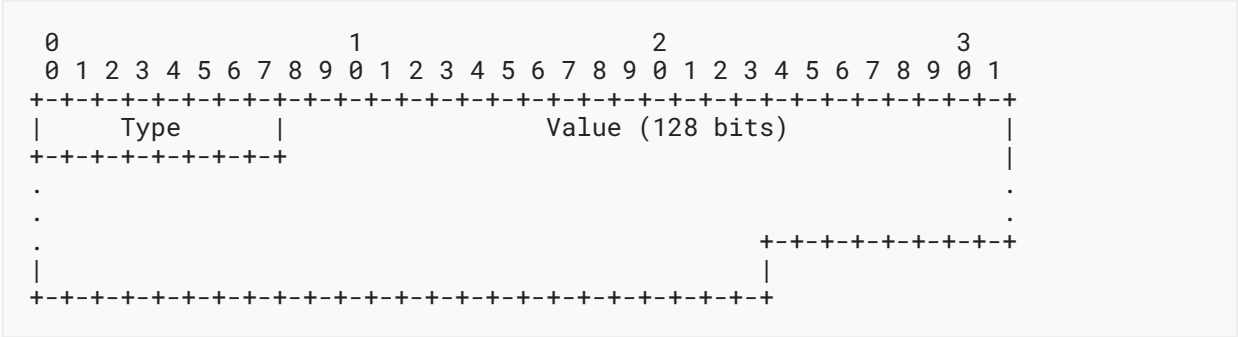


Figure 11: RKAP Authentication Information

Type: Type of data in the Value field carried in this option:

1 Reconfigure key value (used in the Reply message).

2 HMAC-MD5 digest of the message (used in the Reconfigure message).

A 1-octet field.

Value: Data as defined by the Type field. A 16-octet field.

20.4.2. Server Considerations for RKAP

The server selects a reconfigure key for a client during the Request/Reply, Solicit/Reply, or Information-request/Reply message exchange. The server records the reconfigure key and transmits that key to the client in an Authentication option (see Section 21.11) in the Reply message.

The reconfigure key is 128 bits long and **MUST** be a cryptographically strong random or pseudorandom number that cannot easily be predicted.

To provide authentication for a Reconfigure message, the server selects a replay detection value according to the RDM selected by the server and computes an HMAC-MD5 of the Reconfigure message using the reconfigure key for the client. The server computes the HMAC-MD5 over the entire DHCP Reconfigure message, including the Authentication option; the HMAC-MD5 field in the Authentication option is set to 0 for the HMAC-MD5 computation. The server includes the HMAC-MD5 in the authentication information field in an Authentication option included in the Reconfigure message sent to the client.

20.4.3. Client Considerations for RKAP

The client will receive a reconfigure key from the server in an Authentication option (see Section 21.11) in the initial Reply message from the server. The client records the reconfigure key for use in authenticating subsequent Reconfigure messages.

To authenticate a Reconfigure message, the client computes an HMAC-MD5 over the Reconfigure message, with zeroes substituted for the HMAC-MD5 field, using the reconfigure key received from the server. If this computed HMAC-MD5 matches the value in the Authentication option, the client accepts the Reconfigure message.

21. DHCP Options

Options are used to carry additional information and parameters in DHCP messages. Every option shares a common base format, as described in Section 21.1. All values in options are represented in network byte order.

This document specifies the DHCP options defined as part of this base DHCP specification. Other options have been or may be defined in the future in separate documents. See [RFC7227] for guidelines regarding the definition of new options. See Section 24 for additional information about the DHCPv6 "Option Codes" registry maintained by IANA.

Unless otherwise noted, each option may appear only in the options area of a DHCP message and may appear only once. If an option does appear multiple times, each instance is considered separate and the data areas of the options **MUST NOT** be concatenated or otherwise combined.

Options that are allowed to appear only once are called "singleton options". The only non-singleton options defined in this document are the IA_NA (see Section 21.4), Vendor Class (see Section 21.16), Vendor-specific Information (see Section 21.17), and IA_PD (see Section 21.21) options. Also, IA Address (see Section 21.6) and IA Prefix (see Section 21.22) may appear in their respective IA options more than once.

21.1. Format of DHCP Options

The format of DHCP options is:

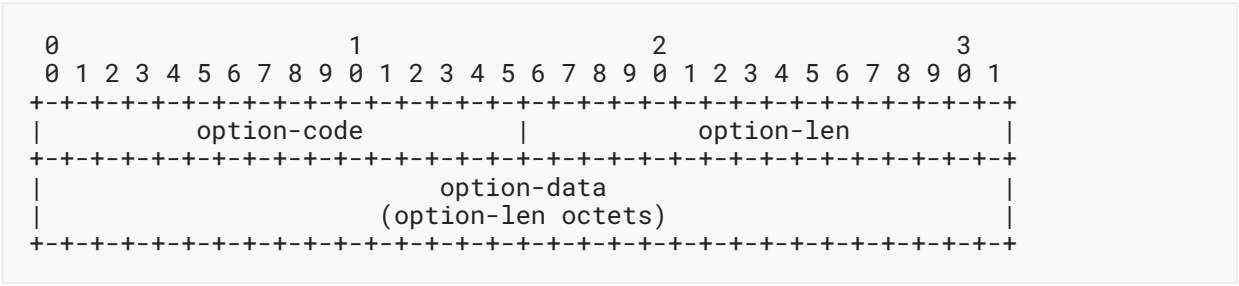


Figure 12: Option Format

option-code: An unsigned integer identifying the specific option type carried in this option. A 2-octet field.

option-len: An unsigned integer giving the length of the option-data field in this option in octets. A 2-octet field.

option-data: The data for the option; the format of this data depends on the definition of the option. A variable-length field (the length, in octets, is specified by option-len).

DHCP options are scoped by using encapsulation. Some options apply generally to the client, some are specific to an IA, and some are specific to the addresses within an IA. These latter two cases are discussed in Sections 21.4 and 21.6.

21.2. Client Identifier Option

The Client Identifier option is used to carry a DUID (see Section 11) that identifies the client. The format of the Client Identifier option is:

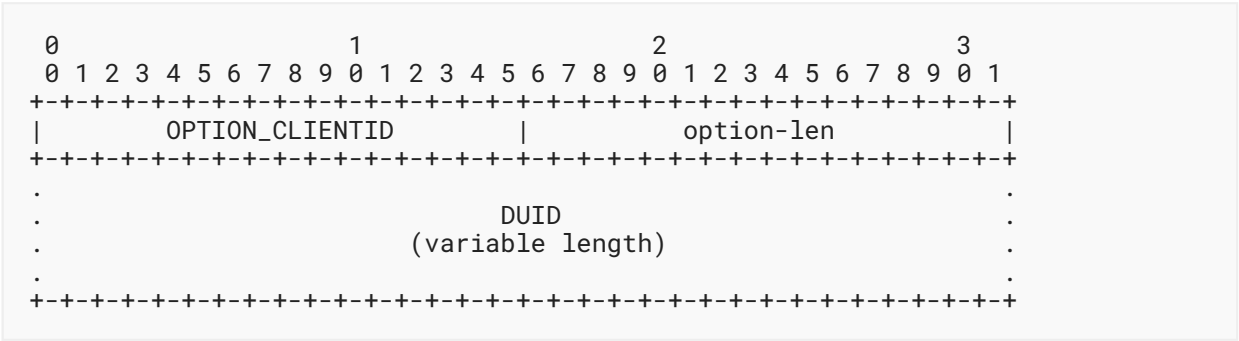


Figure 13: Client Identifier Option Format

option-code: OPTION_CLIENTID (1).

option-len: Length of DUID in octets.

DUID: The DUID for the client.

21.3. Server Identifier Option

The Server Identifier option is used to carry a DUID (see Section 11) that identifies the server. The format of the Server Identifier option is:

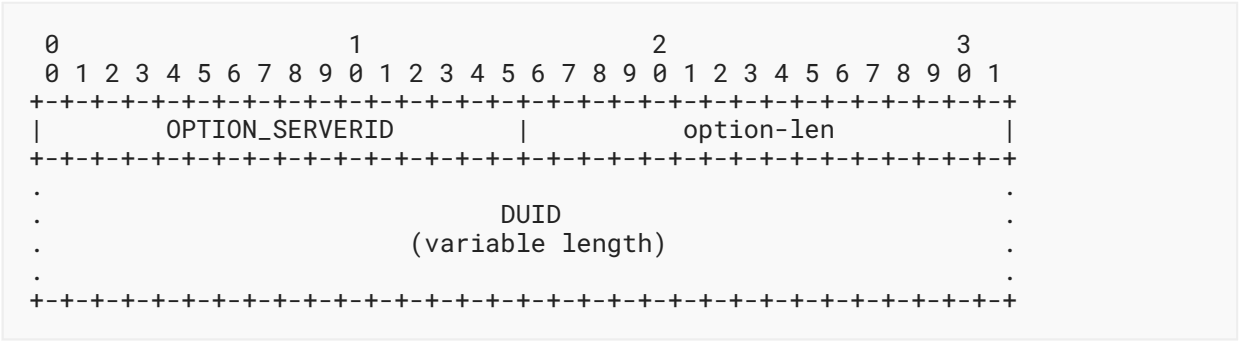


Figure 14: Server Identifier Option Format

option-code: OPTION_SERVERID (2).

option-len: Length of DUID in octets.

DUID: The DUID for the server.

21.4. Identity Association for Non-Temporary Addresses Option

The Identity Association for Non-temporary Addresses (IA_NA) option is used to carry an IA_NA, the parameters associated with the IA_NA, and the non-temporary addresses associated with the IA_NA.

A client that needs a short-term / special-purpose address can use a new IA_NA binding to request an address and release it when finished with it.

Note: Addresses appearing in an IA_NA option are not temporary addresses (see Section 21.5).

The format of the IA_NA option is:

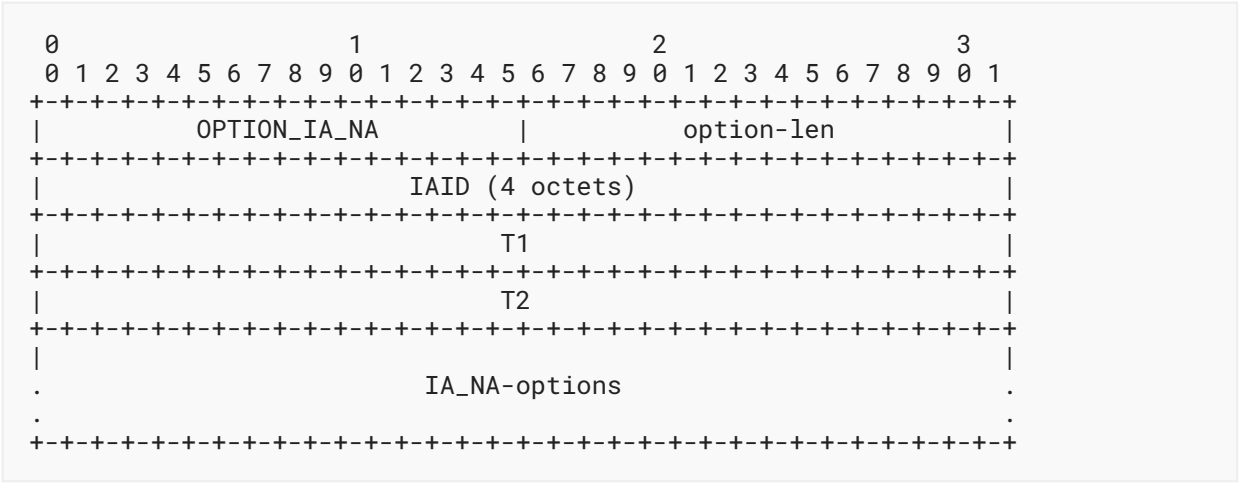


Figure 15: Identity Association for Non-Temporary Addresses Option Format

option-code: OPTION_IA_NA (3).

option-len: 12 + length of IA_NA-options field.

IAID: The unique identifier for this IA_NA; the IAID must be unique among the identifiers for all of this client's IA_NAs. The number space for IA_NA IAIDs is separate from the number space for other IA option types (i.e., 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_NA were obtained to extend the lifetimes of the addresses assigned to the IA_NA; 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 lifetimes of the addresses assigned to the IA_NA; T2 is a time duration relative to the current time expressed in units of seconds. A 4-octet field containing an unsigned integer.

IA_NA-options: Options associated with this IA_NA. A variable-length field (12 octets less than the value in the option-len field).

The IA_NA-options field encapsulates those options that are specific to this IA_NA. For example, all of the IA Address options (see Section 21.6) carrying the addresses associated with this IA_NA are in the IA_NA-options field.

Each IA_NA carries one "set" of non-temporary addresses; it is up to the server policy to determine how many addresses are assigned, but typically at most one address is assigned from each prefix assigned to the link to which the client is attached.

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

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

Note that an IA_NA has no explicit "lifetime" or "lease length" of its own. When the valid lifetimes of all of the addresses in an IA_NA have expired, the IA_NA 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_NA.

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 values in those fields are 0. The values in the T1 and T2 fields are the number of seconds until T1 and T2 and are calculated since reception of the message.

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

The server selects the T1 and T2 values to allow the client to extend the lifetimes of any addresses in the IA_NA before the lifetimes expire, even if the server is unavailable for some short period of time. Recommended values for T1 and T2 are 0.5 and 0.8 times the shortest preferred lifetime of the addresses in the IA that the server is willing to extend, respectively. If the "shortest" preferred lifetime is 0xffffffff ("infinity"), the recommended T1 and T2 values are also 0xffffffff. If the time at which the addresses in an IA_NA are to be renewed is to be left to the discretion of the client, the server sets the T1 and T2 values to 0. The client **MUST** follow the rules defined in Section 14.2.

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

21.5. Identity Association for Temporary Addresses Option

The Identity Association for Temporary Addresses (IA_TA) option is obsoleted. Please refer to [RFC8415] for historical information on this option.

The client **SHOULD NOT** send this option. The server **SHOULD NOT** send this option. When the server receives an IA_TA option, the option **SHOULD** be ignored and the message processing should continue as usual.

As this option was never popular among server or client implementations before being deprecated, any implementations that still attempt to send it are unlikely to have the option processed.

21.6. IA Address Option

The IA Address option is used to specify an address. In this document, it is only specified to be encapsulated within an IA_NA. DHCPv6 Leasequery [RFC5007] makes use of the IA Address option without encapsulating it in IA_NA. The IAaddr-options field encapsulates those options that are specific to this address.

The format of the IA Address option is:

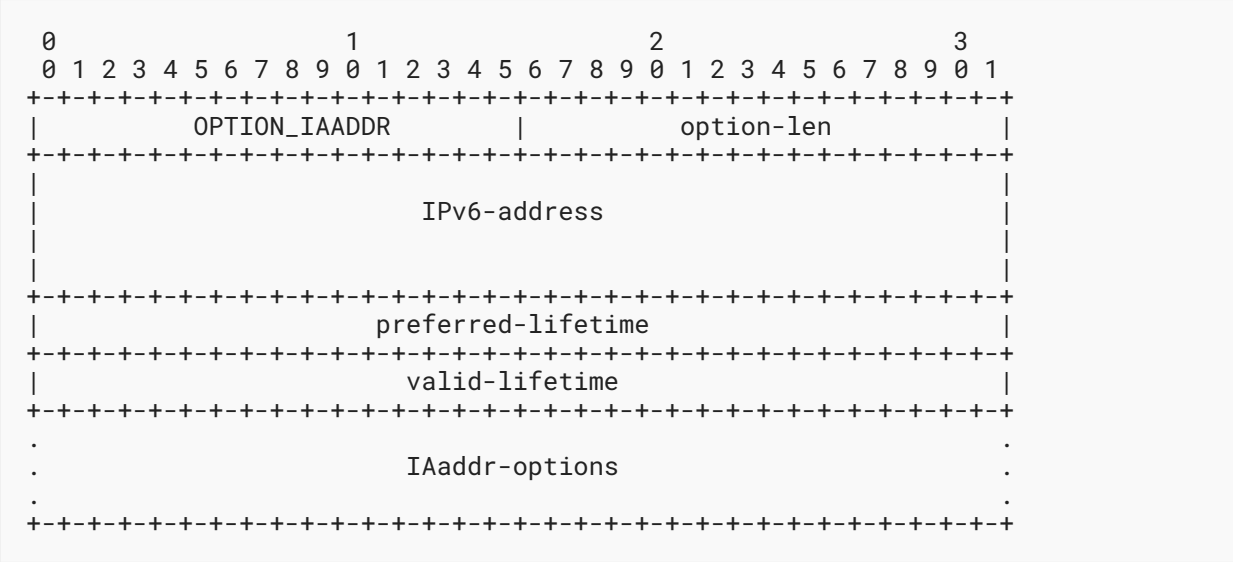


Figure 16: IA Address Option Format

option-code: OPTION_IAADDR (5).

option-len: 24 + length of IAaddr-options field.

IPv6-address: An IPv6 address. A client **MUST NOT** form an implicit prefix with a length other than 128 for this address. A 16-octet field.

preferred-lifetime: The preferred lifetime for the address in the option, expressed in units of seconds. A 4-octet field containing an unsigned integer.

valid-lifetime: The valid lifetime for the address in the option, expressed in units of seconds. A 4-octet field containing an unsigned integer.

IAaddr-options: Options associated with this address. A variable-length field (24 octets less than the value in the option-len field).

In a message sent by a client to a server, the preferred-lifetime and valid-lifetime fields **SHOULD** be set to 0. The server **MUST** ignore any received values.

The client **SHOULD NOT** send the IA Address option with an unspecified address (::).

In a message sent by a server to a client, the client **MUST** use the values in the preferred-lifetime and valid-lifetime fields for the preferred and valid lifetimes. The values in these fields are the number of seconds remaining in each lifetime.

The client **MUST** discard any addresses for which the preferred lifetime is greater than the valid lifetime.

As per Section 7.7, if the valid lifetime of an address is 0xffffffff, it is taken to mean "infinity" and should be used carefully.

More than one IA Address option can appear in an IA_NA option.

21.7. Option Request Option

The Option Request option is used to identify a list of options in a message between a client and a server. The format of the Option Request option is:

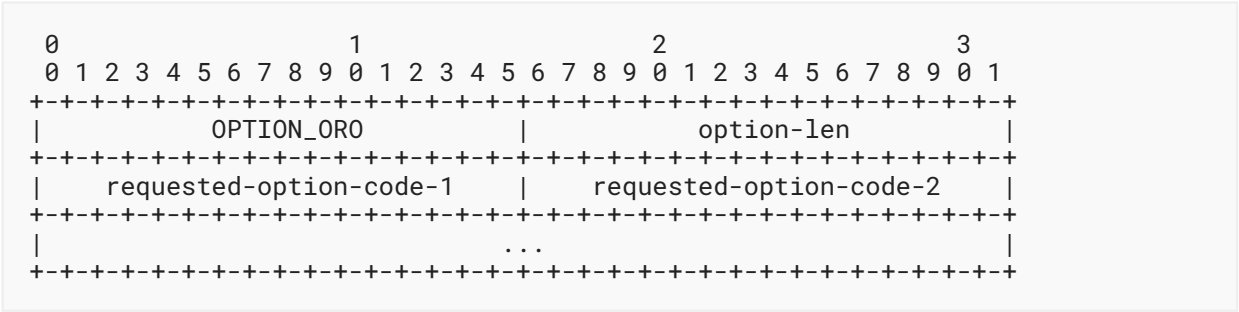


Figure 17: Option Request Option Format

option-code: OPTION_ORO (6).

option-len: 2 * number of requested options.

requested-option-code-n: The option code for an option requested by the client. Each option code is a 2-octet field containing an unsigned integer.

A client **MUST** include an Option Request option in a Solicit, Request, Renew, Rebind, or Information-request message to inform the server about options the client wants the server to send to the client.

The Option Request option **MUST NOT** include the following option codes:

- Client Identifier (see Section 21.2)
- Server Identifier (see Section 21.3)
- IA_NA (see Section 21.4)
- IA_TA (option obsoleted, see Section 21.5)

- IA_PD (see Section 21.21)
- IA Address (see Section 21.6)
- IA Prefix (see Section 21.22)
- Option Request (this section)
- Elapsed Time (see Section 21.9)
- Preference (see Section 21.8)
- Relay Message (see Section 21.10)
- Authentication (see Section 21.11)
- Server Unicast (option obsoleted, see Section 21.12)
- Status Code (see Section 21.13)
- Rapid Commit (see Section 21.14)
- User Class (see Section 21.15)
- Vendor Class (see Section 21.16)
- Interface-Id (see Section 21.18)
- Reconfigure Message (see Section 21.19)
- Reconfigure Accept (see Section 21.20)

Other top-level option codes **MUST** appear in the Option Request option or they will not be sent by the server. Only top-level option codes **MAY** appear in the Option Request option. Option codes encapsulated in a container option **SHOULD NOT** appear in an Option Request option; see [RFC7598] for an example of container options. However, options **MAY** be defined that specify exceptions to this restriction on including encapsulated option codes in an Option Request option. For example, the Option Request option **MAY** be used to signal support for a feature even when that option is encapsulated, as in the case of the Prefix Exclude option [RFC6603]. See [IANA-OPTION-DETAILS].

See [IANA-OPTION-DETAILS] for the authoritative list of which option codes are required, permitted, or forbidden.

21.8. Preference Option

The Preference option is sent by a server to a client to control the selection of a server by the client.

The format of the Preference option is:

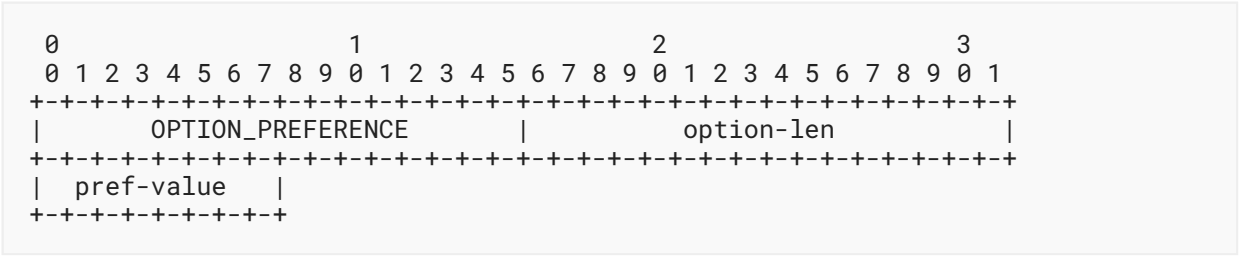


Figure 18: Preference Option Format

option-code: OPTION_PREFERENCE (7).

option-len: 1.

pref-value: The preference value for the server in this message. Allowed values are from 0 (least) to 255 (most preferred). Absence of option means preference 0. A 1-octet unsigned integer.

A server **MAY** include a Preference option in an Advertise message to control the selection of a server by the client. See Section 18.2.9 for information regarding the use of the Preference option by the client and the interpretation of the Preference option data value.

21.9. Elapsed Time Option

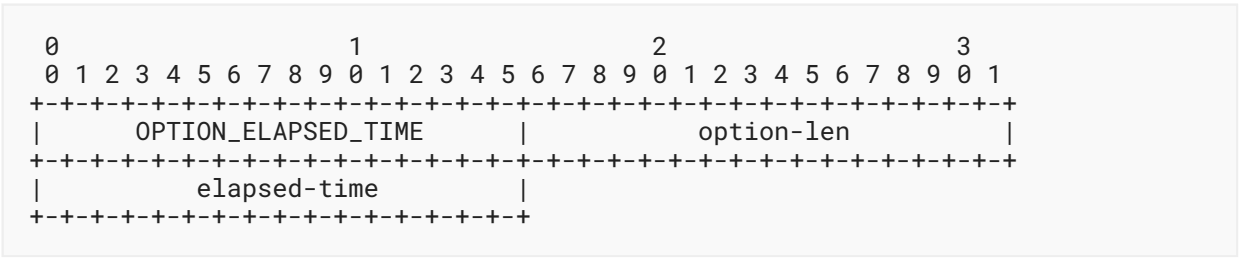


Figure 19: Elapsed Time Option Format

option-code: OPTION_ELAPSED_TIME (8).

option-len: 2.

elapsed-time: The amount of time since the client began its current DHCP transaction. This time is expressed in hundredths of a second (10^-2 seconds). A 2-octet field containing an unsigned integer.

A client **MUST** include an Elapsed Time option in messages to indicate how long the client has been trying to complete a DHCP message exchange. The elapsed time is measured from the time at which the client sent the first message in the message exchange, and the elapsed-time field is set to 0 in the first message in the message exchange. Servers and relay agents use the data value

in this option as input to policy that controls how a server responds to a client message. For example, the Elapsed Time option allows a secondary DHCP server to respond to a request when a primary server has not answered in a reasonable time. The elapsed-time value is a 16-bit (2-octet) unsigned integer. The client uses the value 0xffff to represent any elapsed-time values greater than the largest time value that can be represented in the Elapsed Time option.

21.10. Relay Message Option

The Relay Message option carries a DHCP message in a Relay-forward or Relay-reply message.

The format of the Relay Message option is:

Figure 20: Relay Message Option Format

option-code: OPTION_RELAY_MSG (9).

option-len: Length of DHCP-relay-message field.

DHCP-relay-message: In a Relay-forward message, the received message, relayed verbatim to the next relay agent or server; in a Relay-reply message, the message to be copied and relayed to the relay agent or client whose address is in the peer-address field of the Relay-reply message. The length, in octets, is specified by option-len.

21.11. Authentication Option

The Authentication option carries authentication information to authenticate the identity and contents of DHCP messages. The use of the Authentication option is described in Section 20. The format of the Authentication option is:

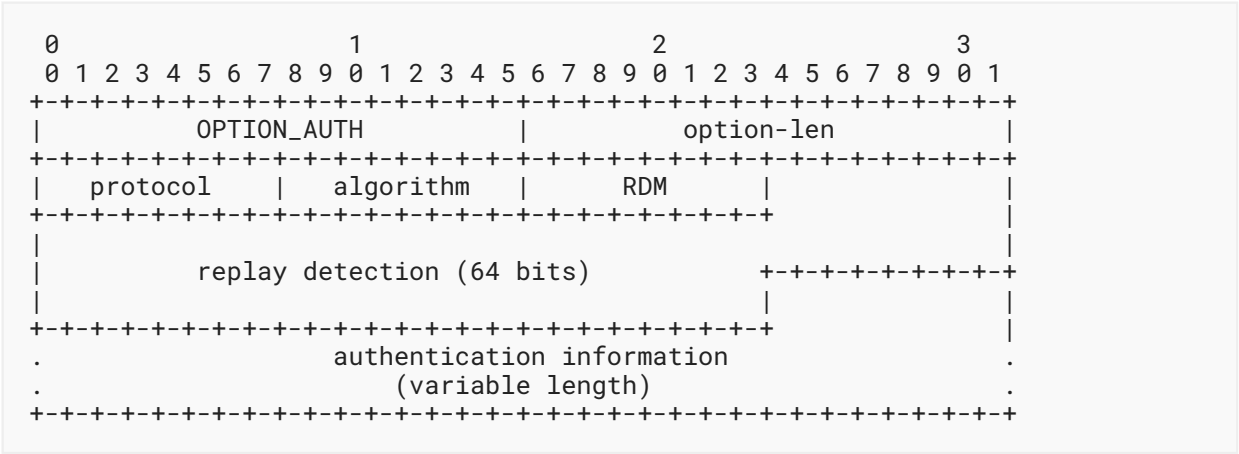


Figure 21: Authentication Option Format

option-code: OPTION_AUTH (11).

option-len: 11 + length of authentication information field.

protocol: The authentication protocol used in this Authentication option. A 1-octet unsigned integer.

algorithm: The algorithm used in the authentication protocol. A 1-octet unsigned integer.

RDM: The replay detection method used in this Authentication option. A 1-octet unsigned integer.

replay detection: The replay detection information for the RDM. A 64-bit (8-octet) field.

authentication information: The authentication information, as specified by the protocol and algorithm used in this Authentication option. A variable-length field (11 octets less than the value in the option-len field).

IANA maintains a registry for the protocol, algorithm, and RDM values at <https://www.iana.org/assignments/auth-namespaces>.

21.12. Server Unicast Option

The Server Unicast option is obsolete. Please refer to [RFC8415] for historical information on this option.

The server **MUST NOT** send this option. When any entity receives the Server Unicast option, the option SHOULD be ignored and the message processing should continue as usual.

21.13. Status Code Option

This option returns a status indication related to the DHCP message or option in which it appears. The format of the Status Code option is:

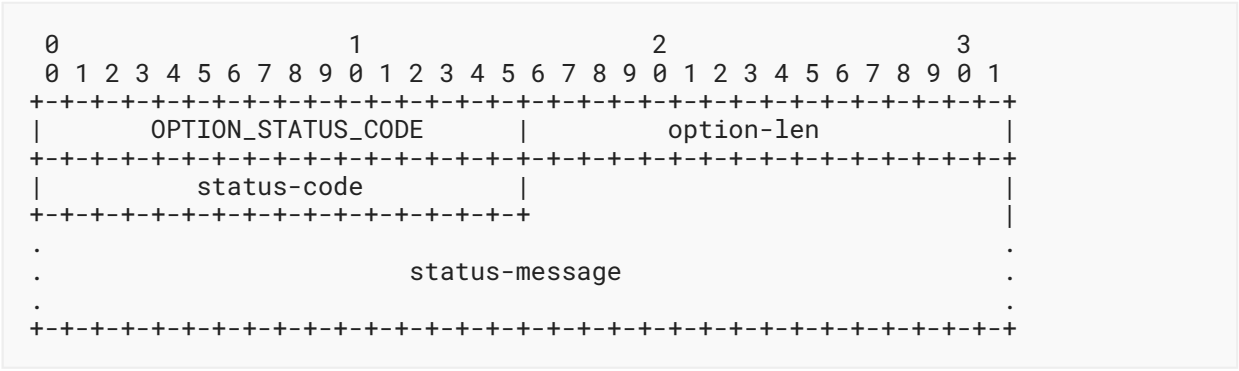


Figure 22: Status Code Option Format

option-code: OPTION_STATUS_CODE (13).

option-len: 2 + length of status-message field.

status-code: The numeric code for the status encoded in this option. A 2-octet field containing an unsigned integer.

status-message: A UTF-8 encoded [RFC3629] text string suitable for display to an end user. **MUST NOT** be NUL-terminated. A variable-length field (2 octets less than the value in the option-len field).

A Status Code option may appear in the "options" field of a DHCP message and/or in the "options" field of another option. If the Status Code option does not appear in a message in which the option could appear, the status of the message is assumed to be Success.

The status-code values are:

Name	Code	Description
Success	0	Success.
UnspecFail	1	Failure, reason unspecified; this status code is sent by a server to indicate a failure not explicitly specified in this document.
NoAddrsAvail	2	The server has no addresses available to assign to the IA(s).
NoBinding	3	Client record (binding) unavailable.
NotOnLink	4	The prefix for the address is not appropriate for the link to which the client is attached.
UseMulticast	5	Sent by a server to a client to force the client to send messages to the server using the All_DHCP_Relay_Agents_and_Servers multicast address. Obsoleted; no longer used.

Name	Code	Description
NoPrefixAvail	6	The server has no prefixes available to assign to the IA_PD(s).

Table 3: Status Code Definitions

See the "Status Codes" registry at <https://www.iana.org/assignments/dhcpv6-parameters> for the current list of status codes.

21.14. Rapid Commit Option

The Rapid Commit option is used to signal the use of the two-message exchange for address assignment. The format of the Rapid Commit option is:

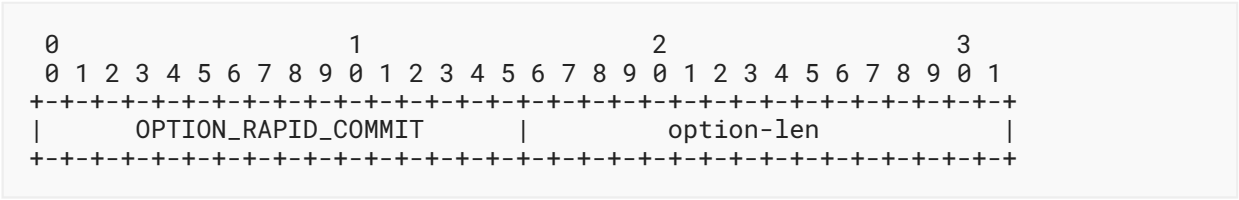


Figure 23: Rapid Commit Option Format

option-code: OPTION_RAPID_COMMIT (14).

option-len: 0.

A client **MAY** include this option in a Solicit message if the client is prepared to perform the Solicit/Reply message exchange described in Section 18.2.1.

A server **MUST** include this option in a Reply message sent in response to a Solicit message when completing the Solicit/Reply message exchange.

DISCUSSION:

- Each server that responds with a Reply to a Solicit that includes a Rapid Commit option will commit the leases in the Reply message to the client but will not receive any confirmation that the client has received the Reply message. Therefore, if more than one server responds to a Solicit that includes a Rapid Commit option, all but one server will commit leases that are not actually used by the client; this could result in incorrect address information in DNS if the DHCP servers update DNS [RFC4704], and responses to leasequery requests [RFC5007] may include information on leases not in use by the client.
- The problem of unused leases can be minimized by designing the DHCP service so that only one server responds to the Solicit or by using relatively short lifetimes for newly assigned leases.

21.15. User Class Option

The User Class option is used by a client to identify the type or category of users or applications it represents.

The format of the User Class option is:

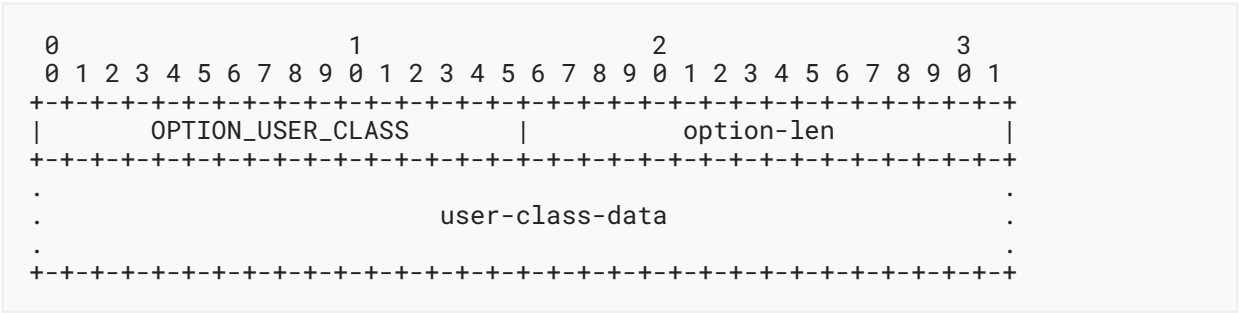


Figure 24: User Class Option Format

option-code: OPTION_USER_CLASS (15).

option-len: Length of user-class-data field.

user-class-data: The user classes carried by the client. The length, in octets, is specified by option-len.

The information contained in the data area of this option is contained in one or more opaque fields that represent the user class or classes of which the client is a member. A server selects configuration information for the client based on the classes identified in this option. For example, the User Class option can be used to configure all clients of people in the accounting department with a different printer than clients of people in the marketing department. The user class information carried in this option **MUST** be configurable on the client.

The data area of the User Class option **MUST** contain one or more instances of user-class-data information. Each instance of user-class-data is formatted as follows:

Figure 25: Format of user-class-data Field

The user-class-len field is 2 octets long and specifies the length of the opaque user-class-data in network byte order.

A server interprets the classes identified in this option according to its configuration to select the appropriate configuration information for the client. A server may use only those user classes that it is configured to interpret in selecting configuration information for a client and ignore any other user classes. In response to a message containing a User Class option, a server may include a User Class option containing those classes that were successfully interpreted by the server so that the client can be informed of the classes interpreted by the server.

21.16. Vendor Class Option

This option is used by a client to identify the vendor that manufactured the hardware on which the client is running. The information contained in the data area of this option is contained in one or more opaque fields that identify details of the hardware configuration. The format of the Vendor Class option is:

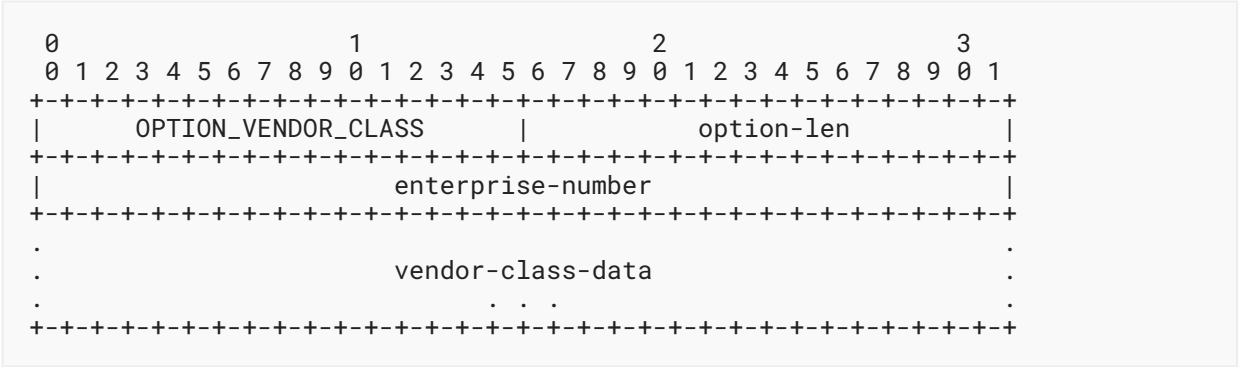


Figure 26: Vendor Class Option Format

option-code: OPTION_VENDOR_CLASS (16).

option-len: 4 + length of vendor-class-data field.

enterprise-number: The vendor's registered Enterprise Number as maintained by IANA [IANA-PEN]. A 4-octet field containing an unsigned integer.

vendor-class-data: The hardware configuration of the node on which the client is running. A variable-length field (4 octets less than the value in the option-len field).

The vendor-class-data field is composed of a series of separate items, each of which describes some characteristic of the client's hardware configuration. Examples of vendor-class-data instances might include the version of the operating system the client is running or the amount of memory installed on the client.

Each instance of vendor-class-data is formatted as follows:

Figure 27: Format of vendor-class-data Field

The vendor-class-len field is 2 octets long and specifies the length of the opaque vendor-class-data in network byte order.

Servers and clients **MUST NOT** include more than one instance of OPTION_VENDOR_CLASS with the same Enterprise Number. Each instance of OPTION_VENDOR_CLASS can carry multiple vendor-class-data instances.

21.17. Vendor-Specific Information Option

This option is used by clients and servers to exchange vendor-specific information.

The format of the Vendor-specific Information option is:

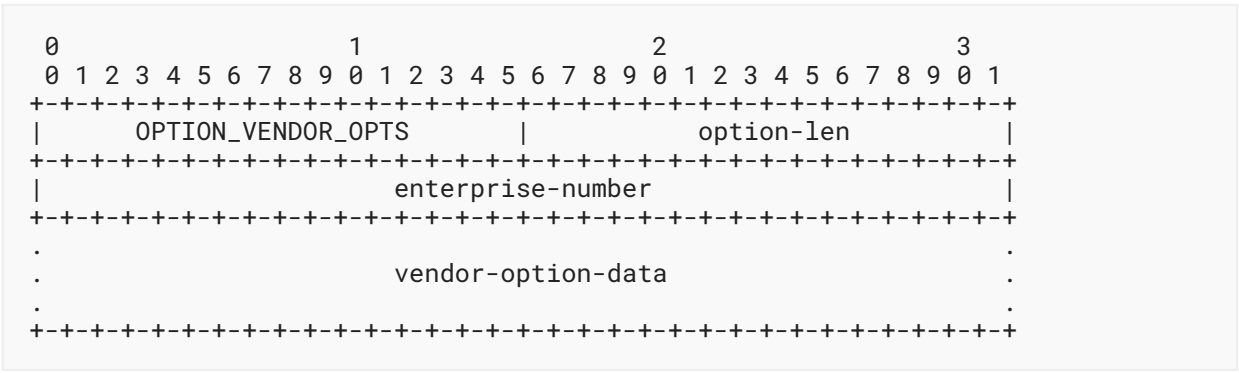


Figure 28: Vendor-Specific Information Option Format

option-code: OPTION_VENDOR_OPTS (17).

option-len: 4 + length of vendor-option-data field.

enterprise-number: The vendor's registered Enterprise Number as maintained by IANA [IANA-PEN]. A 4-octet field containing an unsigned integer.

vendor-option-data: Vendor options, interpreted by vendor-specific code on the clients and servers. A variable-length field (4 octets less than the value in the option-len field).

The definition of the information carried in this option is vendor specific. The vendor is indicated in the enterprise-number field. Use of vendor-specific information allows enhanced operation, utilizing additional features in a vendor's DHCP implementation. A DHCP client that does not receive requested vendor-specific information will still configure the node's IPv6 stack to be functional.

The vendor-option-data field **MUST** be encoded as a sequence of code/length/value fields of format identical to the DHCP options (see Section 21.1). The suboption codes are defined by the vendor identified in the enterprise-number field and are not managed by IANA. Each of the suboptions is formatted as follows:

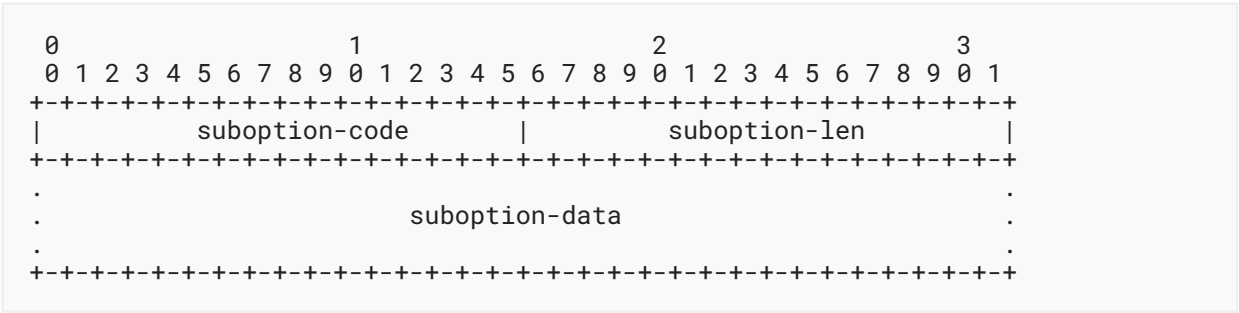


Figure 29: Vendor-Specific Options Format

suboption-code: The code for the suboption. A 2-octet field.

suboption-len: An unsigned integer giving the length of the suboption-data field in this suboption in octets. A 2-octet field.

suboption-data: The data area for the suboption. The length, in octets, is specified by suboption-len.

Multiple instances of the Vendor-specific Information option may appear in a DHCP message. Each instance of the option is interpreted according to the option codes defined by the vendor identified by the Enterprise Number in that option. Servers and clients **MUST NOT** send more than one instance of the Vendor-specific Information option with the same Enterprise Number. Each instance of the Vendor-specific Information option **MAY** contain multiple suboptions.

A client that is interested in receiving a Vendor-specific Information option:

- **MUST** specify the Vendor-specific Information option in an Option Request option.
- **MAY** specify an associated Vendor Class option (see Section 21.16).
- **MAY** specify the Vendor-specific Information option with appropriate data.

Servers only return the Vendor-specific Information options if specified in Option Request options from clients and:

- **MAY** use the Enterprise Numbers in the associated Vendor Class options to restrict the set of Enterprise Numbers in the Vendor-specific Information options returned.
- **MAY** return all configured Vendor-specific Information options.
- **MAY** use other information in the message or in its configuration to determine which set of Enterprise Numbers in the Vendor-specific Information options to return.

21.18. Interface-Id Option

The relay agent **MAY** send the Interface-Id option to identify the interface on which the client message was received. If a relay agent receives a Relay-reply message with an Interface-Id option, the relay agent relays the message to the client through the interface identified by the option.

The format of the Interface-Id option is:

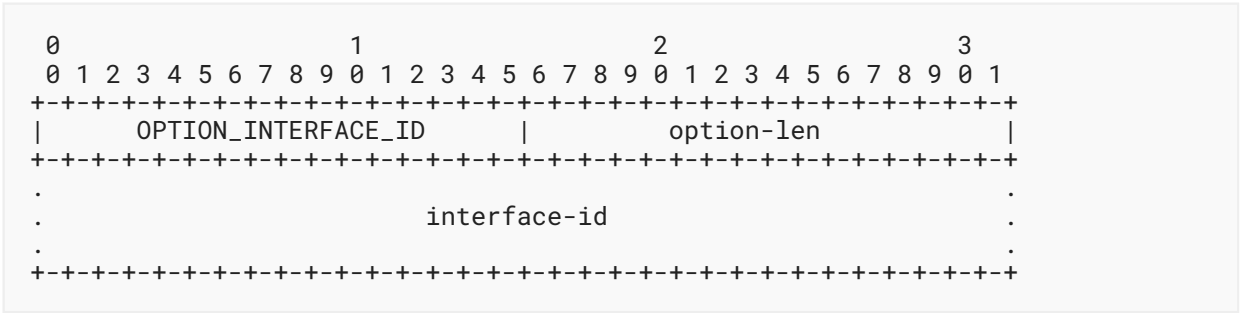


Figure 30: Interface-Id Option Format

option-code: OPTION_INTERFACE_ID (18).

option-len: Length of interface-id field.

interface-id: An opaque value of arbitrary length generated by the relay agent to identify one of the relay agent's interfaces. The length, in octets, is specified by option-len.

The server **MUST** copy the Interface-Id option from the Relay-forward message into the Relay-reply message the server sends to the relay agent in response to the Relay-forward message. This option **MUST NOT** appear in any message except a Relay-forward or Relay-reply message.

Servers **MAY** use the interface-id field for parameter assignment policies. The interface-id value **SHOULD** be considered an opaque value, with policies based on exact match only; that is, the interface-id field **SHOULD NOT** be internally parsed by the server. The interface-id value for an interface **SHOULD** be stable and remain unchanged -- for example, after the relay agent is restarted; if the interface-id value changes, a server will not be able to use it reliably in parameter assignment policies.

21.19. Reconfigure Message Option

A server includes a Reconfigure Message option in a Reconfigure message to indicate to the client whether the client responds with a Renew message, a Rebind message, or an Information-request message. The format of the Reconfigure Message option is:

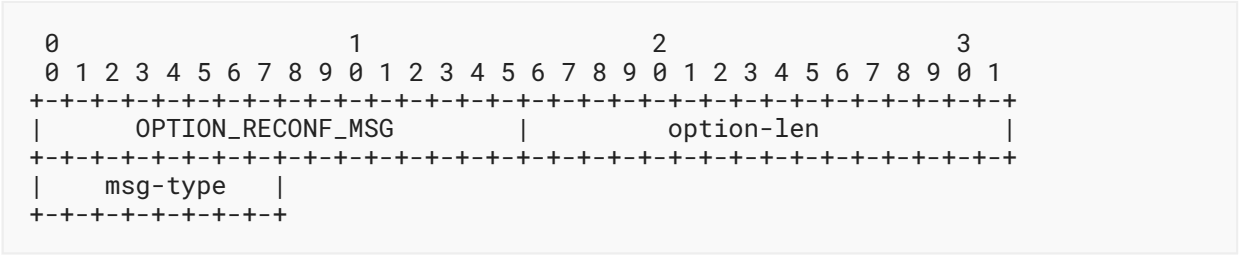


Figure 31: Reconfigure Message Option Format

option-code: OPTION_RECONF_MSG (19).

option-len: 1.

msg-type: 5 for Renew message, 6 for Rebind message, 11 for Information-request message. A 1-octet unsigned integer.

The Reconfigure Message option can only appear in a Reconfigure message.

21.20. Reconfigure Accept Option

A client uses the Reconfigure Accept option to announce to the server whether the client is willing to accept Reconfigure messages, and a server uses this option to tell the client whether or not to accept Reconfigure messages. In the absence of this option, the default behavior is that the client is unwilling to accept Reconfigure messages. The format of the Reconfigure Accept option is:

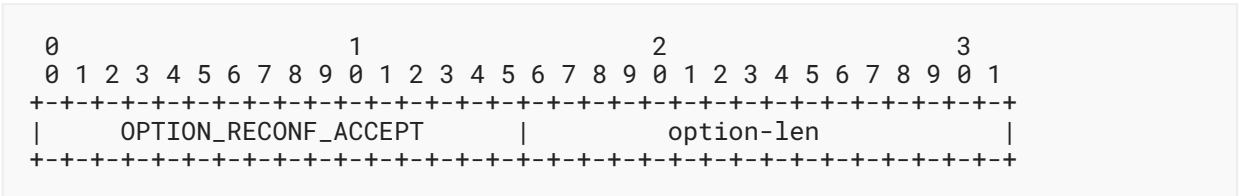


Figure 32: Reconfigure Accept Option Format

option-code: OPTION_RECONF_ACCEPT (20).

option-len: 0.

21.21. Identity Association for Prefix Delegation Option

The IA_PD option is used to carry a prefix delegation identity association, the parameters associated with the IA_PD, and the prefixes associated with it. The format of the IA_PD option is:

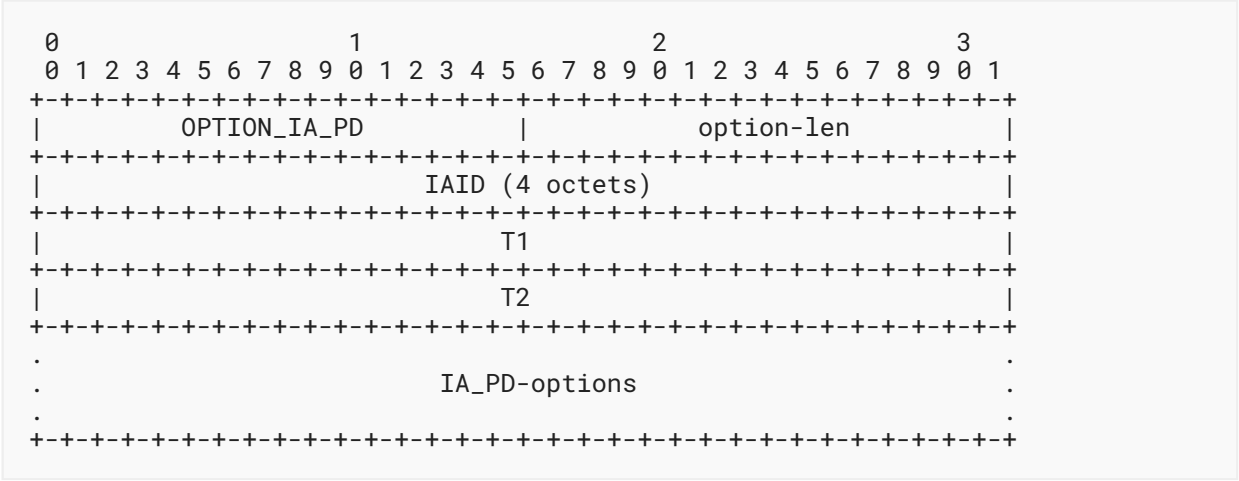


Figure 33: Identity Association for Prefix Delegation Option Format

option-code: OPTION_IA_PD (25).

option-len: 12 + length of IA_PD-options field.

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

T1: The time interval after which the client should contact the server from which the prefixes in the IA_PD were obtained to extend the lifetimes of the prefixes delegated to the IA_PD; T1 is a time duration relative to the message reception 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 lifetimes of the prefixes assigned to the IA_PD; T2 is a time duration relative to the message reception time expressed in units of seconds. A 4-octet field containing an unsigned integer.

IA_PD-options: Options associated with this IA_PD. A variable-length field (12 octets less than the value in the option-len field).

The IA_PD-options field encapsulates those options that are specific to this IA_PD. For example, all of the IA Prefix options (see Section 21.22) carrying the prefixes associated with this IA_PD are in the IA_PD-options field.

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

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

Note that an IA_PD has no explicit "lifetime" or "lease length" of its own. When the valid lifetimes of all of the prefixes in an IA_PD have expired, the IA_PD can be considered as having expired. T1 and T2 fields are included to give the server explicit control over when a client should contact the server about a specific IA_PD.

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 timers, unless values in those fields are 0. The values in the T1 and T2 fields are the number of seconds until T1 and T2.

The server selects the T1 and T2 times to allow the client to extend the lifetimes of any prefixes in the IA_PD before the lifetimes expire, even if the server is unavailable for some short period of time. Recommended values for T1 and T2 are 0.5 and 0.8 times the shortest preferred lifetime of the prefixes in the IA_PD that the server is willing to extend, respectively. If the time at which the prefixes in an IA_PD 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.

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

21.22. IA Prefix Option

The IA Prefix option is used to specify a prefix associated with an IA_PD. The IA Prefix option must be encapsulated in the IA_PD-options field of an IA_PD option (see Section 21.21).

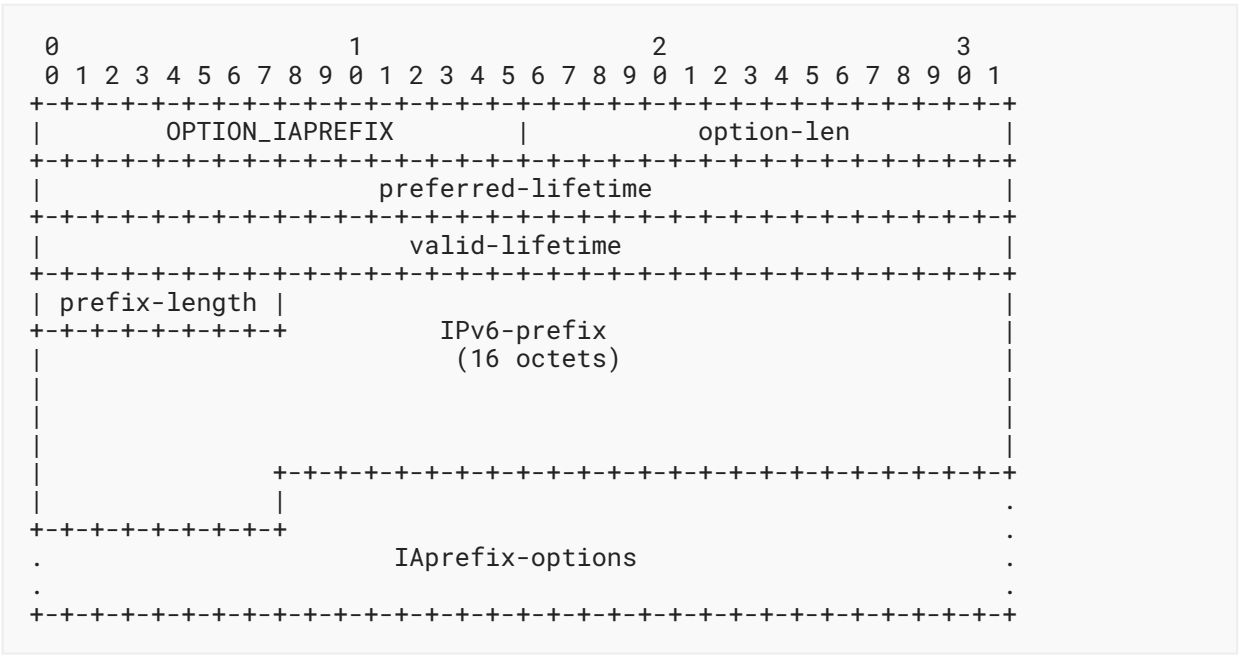


Figure 34: IA Prefix Option Format

option-code: OPTION_IAPREFIX (26).

option-len: 25 + length of IAprefix-options field.

preferred-lifetime: The preferred lifetime for the prefix in the option, expressed in units of seconds. A value of 0xffffffff represents "infinity" (see Section 7.7). A 4-octet field containing an unsigned integer.

valid-lifetime: The valid lifetime for the prefix in the option, expressed in units of seconds. A value of 0xffffffff represents "infinity". A 4-octet field containing an unsigned integer.

prefix-length: Length for this prefix in bits. A 1-octet unsigned integer.

IPv6-prefix: An IPv6 prefix. A 16-octet field.

IAprefix-options: Options associated with this prefix. A variable-length field (25 octets less than the value in the option-len field).

In a message sent by a client to a server, the preferred-lifetime and valid-lifetime fields **SHOULD** be set to 0. The server **MUST** ignore any received values in these lifetime fields.

The client **SHOULD NOT** send an IA Prefix option with 0 in the "prefix-length" field (and an unspecified value (::) in the "IPv6-prefix" field). A client **MAY** send a non-zero value in the "prefix-length" field and the unspecified value (::) in the "IPv6-prefix" field to indicate a preference for the size of the prefix to be delegated. See [RFC8168] for further details on prefix-length hints.

The client **MUST** discard any prefixes for which the preferred lifetime is greater than the valid lifetime.

The values in the preferred-lifetime and valid-lifetime fields are the number of seconds remaining in each lifetime. See Section 18.2.10.1 for more details on how these values are used for delegated prefixes.

As per Section 7.7, the value of 0xffffffff for the preferred lifetime or the valid lifetime is taken to mean "infinity" and should be used carefully.

An IA Prefix option may appear only in an IA_PD option. More than one IA Prefix option can appear in a single IA_PD option.

The status of any operations involving this IA Prefix option is indicated in a Status Code option (see Section 21.13) in the IAprefix-options field.

21.23. Information Refresh Time Option

This option is requested by clients and returned by servers to specify an upper bound for how long a client should wait before refreshing information retrieved from a DHCP server. It is only used in Reply messages in response to Information-request messages. In other messages, there will usually be other information that indicates when the client should contact the server, e.g., T1/T2 times and lifetimes. This option is useful when the configuration parameters change or during a renumbering event, as clients running in the stateless mode will be able to update their configuration.

The format of the Information Refresh Time option is:

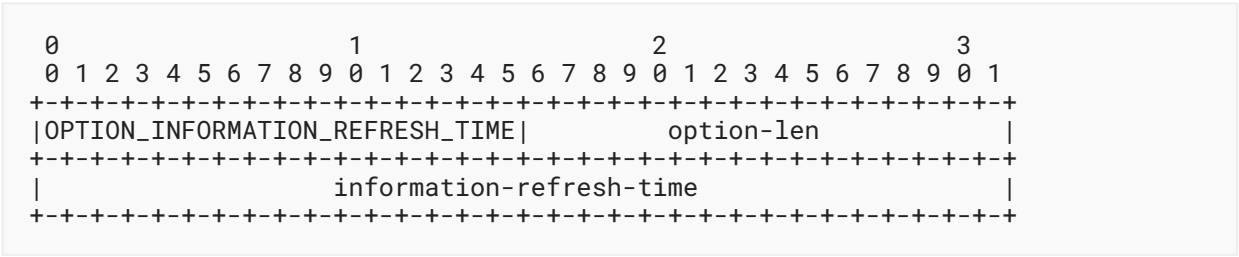


Figure 35: Information Refresh Time Option Format

option-code: OPTION_INFORMATION_REFRESH_TIME (32).

option-len: 4.

information-refresh-time: Time duration relative to the current time, expressed in units of seconds. A 4-octet field containing an unsigned integer.

A DHCP client **MUST** request this option in the Option Request option (see Section 21.7) when sending Information-request messages. A client **MUST NOT** request this option in the Option Request option in any other messages.

A server sending a Reply to an Information-request message **SHOULD** include this option if it is requested in the Option Request option of the Information-request. The option value **MUST NOT** be smaller than IRT_MINIMUM. This option **MUST** only appear in the top-level options area of Reply messages.

If the Reply to an Information-request message does not contain this option, the client **MUST** behave as if the option with the value IRT_DEFAULT was provided.

A client **MUST** use the refresh time IRT_MINIMUM if it receives the option with a value less than IRT_MINIMUM.

As per Section 7.7, the value 0xffffffff is taken to mean "infinity" and implies that the client should not refresh its configuration data without some other trigger (such as detecting movement to a new link).

If a client contacts the server to obtain new data or refresh some existing data before the refresh time expires, then it **SHOULD** also refresh all data covered by this option.

When the client detects that the refresh time has expired, it **SHOULD** try to update its configuration data by sending an Information-request as specified in Section 18.2.6, except that the client **MUST** delay sending the first Information-request by a random amount of time between 0 and INF_MAX_DELAY.

A client **MAY** have a maximum value for the refresh time, where that value is used whenever the client receives this option with a value higher than the maximum. This also means that the maximum value is used when the received value is "infinity". A maximum value might make the client less vulnerable to attacks based on forged DHCP messages. Without a maximum value, a client may be made to use wrong information for a possibly infinite period of time. There may, however, be reasons for having a very long refresh time, so it may be useful for this maximum value to be configurable.

21.24. SOL_MAX_RT Option

A DHCP server sends the SOL_MAX_RT option to a client to override the default value of SOL_MAX_RT. The value of SOL_MAX_RT in the option replaces the default value defined in Section 7.6. One use for the SOL_MAX_RT option is to set a higher value for SOL_MAX_RT; this reduces the Solicit traffic from a client that has not received a response to its Solicit messages.

The format of the SOL_MAX_RT option is:

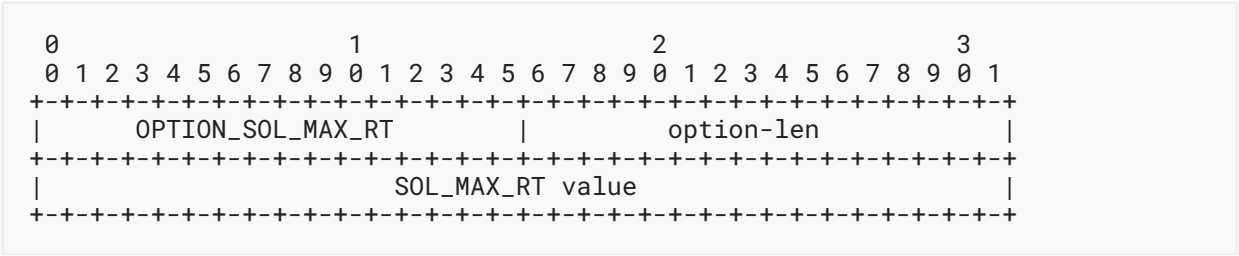


Figure 36: SOL_MAX_RT Option Format

option-code: OPTION_SOL_MAX_RT (82).

option-len: 4.

SOL_MAX_RT value: Overriding value for SOL_MAX_RT in seconds; **MUST** be in this range: 60 <= "value" <= 86400 (1 day). A 4-octet field containing an unsigned integer.

A DHCP client **MUST** include the SOL_MAX_RT option code in any Option Request option (see Section 21.7) it sends in a Solicit message.

The DHCP server **MAY** include the SOL_MAX_RT option in any response it sends to a client that has included the SOL_MAX_RT option code in an Option Request option. The SOL_MAX_RT option is sent as a top-level option in the message to the client.

A DHCP client **MUST** ignore any SOL_MAX_RT option values that are less than 60 or more than 86400.

If a DHCP client receives a message containing a SOL_MAX_RT option that has a valid value for SOL_MAX_RT, the client **MUST** set its internal SOL_MAX_RT parameter to the value contained in the SOL_MAX_RT option. This value of SOL_MAX_RT is then used by the retransmission mechanism defined in Sections 15 and 18.2.1.

The purpose of this mechanism is to give network administrators a way to avoid excessive DHCP traffic if all DHCP servers become unavailable. Therefore, this value is expected to be retained for as long as practically possible.

An updated SOL_MAX_RT value applies only to the network interface on which the client received the SOL_MAX_RT option.

21.25. INF_MAX_RT Option

A DHCP server sends the INF_MAX_RT option to a client to override the default value of INF_MAX_RT. The value of INF_MAX_RT in the option replaces the default value defined in Section 7.6. One use for the INF_MAX_RT option is to set a higher value for INF_MAX_RT; this reduces the Information-request traffic from a client that has not received a response to its Information-request messages.

The format of the INF_MAX_RT option is:

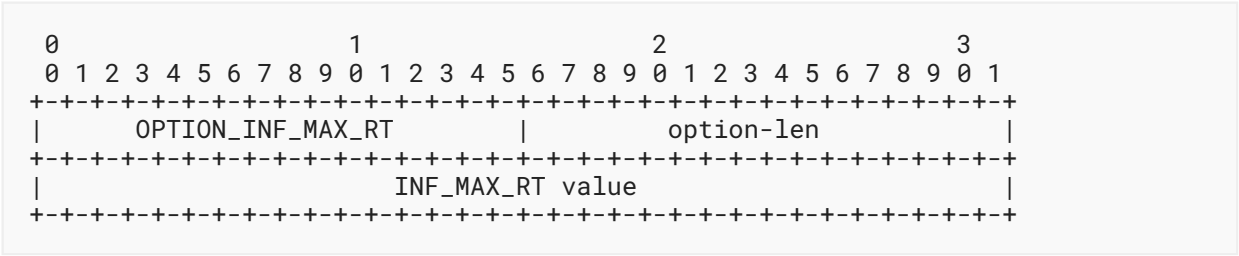


Figure 37: INF_MAX_RT Option Format

option-code: OPTION_INF_MAX_RT (83).

option-len: 4.

INF_MAX_RT value: Overriding value for INF_MAX_RT in seconds; **MUST** be in this range: 60 <= "value" <= 86400 (1 day). A 4-octet field containing an unsigned integer.

A DHCP client **MUST** include the INF_MAX_RT option code in any Option Request option (see Section 21.7) it sends in an Information-request message.

The DHCP server **MAY** include the INF_MAX_RT option in any response it sends to a client that has included the INF_MAX_RT option code in an Option Request option. The INF_MAX_RT option is a top-level option in the message to the client.

A DHCP client **MUST** ignore any INF_MAX_RT option values that are less than 60 or more than 86400.

If a DHCP client receives a message containing an INF_MAX_RT option that has a valid value for INF_MAX_RT, the client **MUST** set its internal INF_MAX_RT parameter to the value contained in the INF_MAX_RT option. This value of INF_MAX_RT is then used by the retransmission mechanism defined in Sections 15 and 18.2.6.

An updated INF_MAX_RT value applies only to the network interface on which the client received the INF_MAX_RT option.

22. Security Considerations

This section discusses security considerations that are not related to privacy. See Section 23 for a discussion dedicated to privacy.

The threat to DHCP is inherently an insider threat (assuming a properly configured network where DHCP ports are blocked on the perimeter gateways of the enterprise). Regardless of the gateway configuration, however, the potential attacks by insiders and outsiders are the same.

DHCP lacks end-to-end encryption between clients and servers; thus, hijacking, tampering, and eavesdropping attacks are all possible as a result. Some network environments (discussed below) can be secured through various means to minimize these attacks.

The threat common to both the client and the server is the "resource-exhaustion" DoS attack. Typically, these attacks involve the exhaustion of available assigned addresses or delegatable prefixes or the exhaustion of CPU or network bandwidth, and they are present any time there is a shared resource. Some forms of these exhaustion attacks can be partially mitigated by appropriate server policy, e.g., limiting the maximum number of leases any one client can get, limiting the number of leases one client can decline, and limiting the number of messages a single client can transmit in a period of time.

22.1. Client Security Considerations

One attack specific to a DHCP client is the establishment of a malicious server with the intent of providing incorrect configuration information to the client. The motivation for doing so may be to mount a "man-in-the-middle" attack that causes the client to communicate with a malicious server instead of a valid server for some service (such as DNS or NTP). The malicious server may also mount a DoS attack through misconfiguration of the client; this attack would cause all network communication from the client to fail.

A malicious DHCP server might cause a client to set its SOL_MAX_RT and INF_MAX_RT parameters to an unreasonably high value with the SOL_MAX_RT (see Section 21.24) and INF_MAX_RT (see Section 21.25) options; this may cause an undue delay in a client completing its DHCP protocol transaction in the case where no other valid response is received. Assuming that the client also receives a response from a valid DHCP server, large values for SOL_MAX_RT and INF_MAX_RT will not have any effect.

Another threat to DHCP clients originates from mistakenly or accidentally configured DHCP servers that answer DHCP client requests with unintentionally incorrect configuration parameters.

If a client implementation supports the reconfigure mechanism, see Section 22.3.

22.2. Server Security Considerations

The threat specific to a DHCP server is an invalid client masquerading as a valid client. The motivation for this may be for theft of service or to circumvent auditing for any number of nefarious purposes.

The messages exchanged between relay agents and servers may be used to mount a man-in-the-middle or DoS attack. Communication between a server and a relay agent, and communication between relay agents, can be authenticated and encrypted through the use of IPsec, as described in [RFC8213].

However, the use of manually configured pre-shared keys for IPsec between relay agents and servers does not defend against replayed DHCP messages. Replayed messages can represent a DoS attack through exhaustion of processing resources but not through misconfiguration or exhaustion of other resources such as assignable addresses and delegatable prefixes.

If a server implementation supports the reconfigure mechanism, see Section 22.3.

22.3. Reconfigure Security Considerations

RKAP, described in Section 20.4, provides protection against the use of a Reconfigure message by a malicious DHCP server to mount a DoS or man-in-the-middle attack on a client. This protocol can be compromised by an attacker that can intercept the initial message in which the DHCP server sends the key as plain text to the client.

Because of the opportunity for attack through the Reconfigure message, a DHCP client **MUST** discard any Reconfigure message that does not include authentication or that does not pass the validation process for the authentication protocol.

A DHCP client may also be subject to attack through the receipt of a Reconfigure message from a malicious server that causes the client to obtain incorrect configuration information from that server. Note that although a client sends its response (Renew, Rebind, or Information-request message) through a relay agent and, therefore, that response will only be received by servers to which DHCP messages are relayed, a malicious server could send a Reconfigure message to a client, followed (after an appropriate delay) by a Reply message that would be accepted by the client. Thus, a malicious server that is not on the network path between the client and the server may still be able to mount a Reconfigure attack on a client. The use of transaction IDs that are cryptographically sound and cannot easily be predicted will also reduce the probability that such an attack will be successful.

22.4. Mitigation Considerations

Various network environments also offer levels of security if deployed as described below.

- In enterprise and factory networks, use of authentication per [IEEE8802.1x] can prevent unknown or untrusted clients from connecting to the network. However, this does not necessarily assure that the connected client will be a good DHCP or network actor.
- For wired networks where clients typically are connected to a switch port, snooping DHCP multicast (or unicast) traffic becomes difficult, as the switches limit the traffic delivered to a port. The client's DHCP messages (multicast to All_DHCP_Relay_Agents_and_Servers) are only forwarded to the DHCP server's (or relay's) switch port -- not all ports. Also, the server's (or relay's) unicast replies are only delivered to the target client's port -- not all ports.
- In public networks (such as a Wi-Fi network in a coffee shop or airport), it is possible for others within radio range to snoop DHCP and other traffic. But in these environments, there is very little if anything that can be learned from the DHCP traffic itself (either from client to server or from server to client) if the privacy considerations provided in Section 23 are followed. Even for devices that do not follow the privacy considerations, there is little that can be learned that would not be available from subsequent communications anyway (such as the device's Media Access Control (MAC) address). Also, because all clients will typically receive similar configuration details, a bad actor that initiates a DHCP request itself can learn much of such information. As mentioned above, one threat is that the RKAP key for a client can be learned (if the initial Solicit/Advertise/Request/Reply exchange is monitored) and trigger a premature reconfiguration, but this is relatively easily prevented by

disallowing direct client-to-client communication on these networks or using [RFC7610] and [RFC7513].

Many of the attacks by rogue servers can be mitigated by making use of the mechanisms described in [RFC7610] and [RFC7513].

23. Privacy Considerations

For an extended discussion about privacy considerations for the client, see [RFC7824]:

- In particular, Section 3 of [RFC7824] discusses various identifiers that could be misused to track the client.
- Section 4 of [RFC7824] discusses existing mechanisms that may have an impact on a client's privacy.
- Finally, Section 5 of [RFC7824] discusses potential attack vectors.

For recommendations regarding how to address or mitigate those issues, see [RFC7844].

This specification does not define any allocation strategies for servers. Implementers are expected to develop their own algorithm for the server to choose a resource out of the available pool. Several possible allocation strategies are mentioned in Section 4.3 of [RFC7824]. Please keep in mind that the list in [RFC7824] is not exhaustive; there are certainly other possible strategies. Readers are also encouraged to read [RFC7707] -- in particular, Section 4.1.2 of [RFC7707], which discusses the problems with certain allocation strategies.

24. IANA Considerations

This document does not define any new DHCP name spaces or definitions.

The publication of this document does not change the assignment rules for new values for message types, option codes, DUID types, or status codes.

The list of assigned values used in DHCPv6 is available at <https://www.iana.org/assignments/dhcpv6-parameters>.

IANA has updated all references to [RFC8415] to this document at <https://www.iana.org/assignments/dhcpv6-parameters>.

IANA has added a new column "Status" to all registries on the DHCPv6 parameters page at <https://www.iana.org/assignments/dhcpv6-parameters> and has left each entry blank except as indicated below:

- In the "Option Codes" registry, the "Status" column value has been set to "Obsolete" for the IA_TA (option code 4) and UNICAST (option code 12) rows.
- In the "Status Codes" registry, the "Status" column value has been set to "Obsolete" for the UseMulticast (status code 5) row.

IANA has updated other references to [RFC8415] with references to this document at:

- <https://www.iana.org/assignments/auth-namespaces> (four entries)
- <https://www.iana.org/assignments/bootp-dhcp-parameters> (two entries)
- <https://www.iana.org/assignments/ipv6-multicast-addresses> (two entries)
- <https://www.iana.org/assignments/service-names-port-numbers> (two entries; UDP ports 546 and 547)

25. References

25.1. Normative References

[BCP145] Best Current Practice 145, <https://www.rfc-editor.org/info/bcp145>.
At the time of writing, this BCP comprises the following:

Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[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>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[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>.

[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>.

[RFC6221] Miles, D., Ed., Ooghe, S., Dec, W., Krishnan, S., and A. Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221, DOI 10.17487/RFC6221, May 2011, <https://www.rfc-editor.org/info/rfc6221>.

[RFC6355] Narten, T. and J. Johnson, "Definition of the UUID-Based DHCPv6 Unique Identifier (DUID-UUID)", RFC 6355, DOI 10.17487/RFC6355, August 2011, <https://www.rfc-editor.org/info/rfc6355>.

[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>.

[RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", BCP 204, RFC 7934, DOI 10.17487/RFC7934, July 2016, <https://www.rfc-editor.org/info/rfc7934>.

[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>.

[RFC8213] Volz, B. and Y. Pal, "Security of Messages Exchanged between Servers and Relay Agents", RFC 8213, DOI 10.17487/RFC8213, August 2017, <https://www.rfc-editor.org/info/rfc8213>.

25.2. Informative References

[Err6159] RFC Errata, Erratum ID 6159, RFC 8415, <https://www.rfc-editor.org/errata/eid1912>.

[Err6183] RFC Errata, Erratum ID 6183, RFC 8415, <https://www.rfc-editor.org/errata/eid1912>.

[Err6269] RFC Errata, Erratum ID 6269, RFC 8415, <https://www.rfc-editor.org/errata/eid1912>.

[IANA-HARDWARE-TYPES] IANA, "Hardware Types", <https://www.iana.org/assignments/arp-parameters>.

[IANA-OPTION-DETAILS] IANA, "Option Codes", <https://www.iana.org/assignments/dhcpv6-parameters>.

[IANA-PEN] IANA, "Private Enterprise Numbers", <https://www.iana.org/assignments/enterprise-numbers>.

[IANA-RESERVED-IID] IANA, "Reserved IPv6 Interface Identifiers", <https://www.iana.org/assignments/ipv6-interface-ids>.

[IEEE8802.1x] IEEE/ISO/IEC, "IEEE/ISO/IEC International Standard-Telecommunications and exchange between information technology systems--Requirements for local and metropolitan area networks--Part 1X:Port-based network access control", ISO/IEC/IEEE 8802-1X:2021, DOI 10.1109/IEEESTD.2021.9650828, 2021, <https://ieeexplore.ieee.org/document/9650828>.

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

[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>.

[RFC3162] Aboba, B., Zorn, G., and D. Mitton, "RADIUS and IPv6", RFC 3162, DOI 10.17487/RFC3162, August 2001, <https://www.rfc-editor.org/info/rfc3162>.

[RFC3290] Bernet, Y., Blake, S., Grossman, D., and A. Smith, "An Informal Management Model for Diffserv Routers", RFC 3290, DOI 10.17487/RFC3290, May 2002, <https://www.rfc-editor.org/info/rfc3290>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/info/rfc3629>.

[RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, DOI 10.17487/RFC3646, December 2003, <https://www.rfc-editor.org/info/rfc3646>.

[RFC3769] Miyakawa, S. and R. Droms, "Requirements for IPv6 Prefix Delegation", RFC 3769, DOI 10.17487/RFC3769, June 2004, <https://www.rfc-editor.org/info/rfc3769>.

[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>.

[RFC4477] Chown, T., Venaas, S., and C. Strauf, "Dynamic Host Configuration Protocol (DHCP): IPv4 and IPv6 Dual-Stack Issues", RFC 4477, DOI 10.17487/RFC4477, May 2006, <https://www.rfc-editor.org/info/rfc4477>.

[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>.

[RFC4943] Roy, S., Durand, A., and J. Paugh, "IPv6 Neighbor Discovery On-Link Assumption Considered Harmful", RFC 4943, DOI 10.17487/RFC4943, September 2007, <https://www.rfc-editor.org/info/rfc4943>.

[RFC4957] Krishnan, S., Ed., Montavont, N., Njedjou, E., Veerepalli, S., and A. Yegin, Ed., "Link-Layer Event Notifications for Detecting Network Attachments", RFC 4957, DOI 10.17487/RFC4957, August 2007, <https://www.rfc-editor.org/info/rfc4957>.

[RFC4994] Zeng, S., Volz, B., Kinnear, K., and J. Brzozowski, "DHCPv6 Relay Agent Echo Request Option", RFC 4994, DOI 10.17487/RFC4994, September 2007, <https://www.rfc-editor.org/info/rfc4994>.

[RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", RFC 5007, DOI 10.17487/RFC5007, September 2007, <https://www.rfc-editor.org/info/rfc5007>.

[RFC5453] Krishnan, S., "Reserved IPv6 Interface Identifiers", RFC 5453, DOI 10.17487/RFC5453, February 2009, <https://www.rfc-editor.org/info/rfc5453>.

[RFC5612] Eronen, P. and D. Harrington, "Enterprise Number for Documentation Use", RFC 5612, DOI 10.17487/RFC5612, August 2009, <https://www.rfc-editor.org/info/rfc5612>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC5908] Gayraud, R. and B. Lourdelet, "Network Time Protocol (NTP) Server Option for DHCPv6", RFC 5908, DOI 10.17487/RFC5908, June 2010, <https://www.rfc-editor.org/info/rfc5908>.

[RFC6059] Krishnan, S. and G. Daley, "Simple Procedures for Detecting Network Attachment in IPv6", RFC 6059, DOI 10.17487/RFC6059, November 2010, <https://www.rfc-editor.org/info/rfc6059>.

[RFC6422] Lemon, T. and Q. Wu, "Relay-Supplied DHCP Options", RFC 6422, DOI 10.17487/RFC6422, December 2011, <https://www.rfc-editor.org/info/rfc6422>.

[RFC6603] Korhonen, J., Ed., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", RFC 6603, DOI 10.17487/RFC6603, May 2012, <https://www.rfc-editor.org/info/rfc6603>.

[RFC6879] Jiang, S., Liu, B., and B. Carpenter, "IPv6 Enterprise Network Renumbering Scenarios, Considerations, and Methods", RFC 6879, DOI 10.17487/RFC6879, February 2013, <https://www.rfc-editor.org/info/rfc6879>.

[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>.

[RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <https://www.rfc-editor.org/info/rfc7084>.

[RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", RFC 7136, DOI 10.17487/RFC7136, February 2014, <https://www.rfc-editor.org/info/rfc7136>.

[RFC7341] Sun, Q., Cui, Y., Siodelski, M., Krishnan, S., and I. Farrer, "DHCPv4-over-DHCPv6 (DHCP 4o6) Transport", RFC 7341, DOI 10.17487/RFC7341, August 2014, <https://www.rfc-editor.org/info/rfc7341>.

[RFC7368] Chown, T., Ed., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", RFC 7368, DOI 10.17487/RFC7368, October 2014, <https://www.rfc-editor.org/info/rfc7368>.

[RFC7513] Bi, J., Wu, J., Yao, G., and F. Baker, "Source Address Validation Improvement (SAVI) Solution for DHCP", RFC 7513, DOI 10.17487/RFC7513, May 2015, <https://www.rfc-editor.org/info/rfc7513>.

[RFC7598] Mrugalski, T., Troan, O., Farrer, I., Perreault, S., Dec, W., Bao, C., Yeh, L., and X. Deng, "DHCPv6 Options for Configuration of Softwire Address and Port-Mapped Clients", RFC 7598, DOI 10.17487/RFC7598, July 2015, <https://www.rfc-editor.org/info/rfc7598>.

[RFC7610] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield: Protecting against Rogue DHCPv6 Servers", BCP 199, RFC 7610, DOI 10.17487/RFC7610, August 2015, <https://www.rfc-editor.org/info/rfc7610>.

[RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016, <https://www.rfc-editor.org/info/rfc7707>.

[RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <https://www.rfc-editor.org/info/rfc7721>.

[RFC7824] Krishnan, S., Mrugalski, T., and S. Jiang, "Privacy Considerations for DHCPv6", RFC 7824, DOI 10.17487/RFC7824, May 2016, <https://www.rfc-editor.org/info/rfc7824>.

[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>.

[RFC7943] Gont, F. and W. Liu, "A Method for Generating Semantically Opaque Interface Identifiers (IIDs) with the Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 7943, DOI 10.17487/RFC7943, September 2016, <https://www.rfc-editor.org/info/rfc7943>.

[RFC7969] Lemon, T. and T. Mrugalski, "Customizing DHCP Configuration on the Basis of Network Topology", RFC 7969, DOI 10.17487/RFC7969, October 2016, <https://www.rfc-editor.org/info/rfc7969>.

[RFC8168] Li, T., Liu, C., and Y. Cui, "DHCPv6 Prefix-Length Hint Issues", RFC 8168, DOI 10.17487/RFC8168, May 2017, <https://www.rfc-editor.org/info/rfc8168>.

[RFC8357] Shen, N. and E. Chen, "Generalized UDP Source Port for DHCP Relay", RFC 8357, DOI 10.17487/RFC8357, March 2018, <https://www.rfc-editor.org/info/rfc8357>.

[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>.

[RFC8947] Volz, B., Mrugalski, T., and CJ. Bernardos, "Link-Layer Address Assignment Mechanism for DHCPv6", RFC 8947, DOI 10.17487/RFC8947, December 2020, <https://www.rfc-editor.org/info/rfc8947>.

[RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", RFC 8981, DOI 10.17487/RFC8981, February 2021, <https://www.rfc-editor.org/info/rfc8981>.

[RFC8987] Farrer, I., Kottapalli, N., Hunek, M., and R. Patterson, "DHCPv6 Prefix Delegating Relay Requirements", RFC 8987, DOI 10.17487/RFC8987, February 2021, <https://www.rfc-editor.org/info/rfc8987>.

[RFC9096] Gont, F., Žorž, J., Patterson, R., and B. Volz, "Improving the Reaction of Customer Edge Routers to IPv6 Renumbering Events", BCP 234, RFC 9096, DOI 10.17487/RFC9096, August 2021, <https://www.rfc-editor.org/info/rfc9096>.

[RFC9243] Farrer, I., Ed., "A YANG Data Model for DHCPv6 Configuration", RFC 9243, DOI 10.17487/RFC9243, June 2022, <https://www.rfc-editor.org/info/rfc9243>.

[RFC9686] Kumari, W., Krishnan, S., Asati, R., Colitti, L., Linkova, J., and S. Jiang, "Registering Self-Generated IPv6 Addresses Using DHCPv6", RFC 9686, DOI 10.17487/RFC9686, December 2024, <https://www.rfc-editor.org/info/rfc9686>.

[TR-187] Broadband Forum, "IPv6 for PPP Broadband Access", TR-187, Issue: 2, February 2013, <https://www.broadband-forum.org/pdfs/tr-187-2-0-0.pdf>.

Appendix A. Summary of Changes from RFC 8415

This appendix provides a summary of the differences between this document and [RFC8415]:

1. The following mechanisms were obsoleted. These were not widely deployed while adding complexity to client and server implementations. Legacy implementations **MAY** support them, but implementations conformant to this document **MUST NOT** rely on them. Obsoleting these features does not cause any interoperability issues when mixing updated and non-updated clients, relay agents, and servers as these mechanisms were "optional".
   - IA_TA option. The Identity Association for Temporary Addresses option has been obsoleted. A client that needs a short-term / special purpose address can use a new IA_NA binding to request an address and release it when finished with it.
   - UNICAST option. The Server Unicast option has been obsoleted. Use of this was rarely practical as typically relay agents between the client and server need to glean information from the communication and cannot be bypassed.

- UseMulticast status code. The UseMulticast status code has been obsoleted. Clients will always multicast messages (as Server Unicast option has been obsoleted) and servers will no longer check for unicast traffic.

2. The following errata reports for [RFC8415] were incorporated: [Err6159], [Err6269], and [Err6183]. Note that EID 6269 was no longer applicable after the Server Unicast Option was obsoleted. Note that EID 6159 was also no longer applicable as temporary addresses have been obsoleted. Indeed, the text that EID 6159 corrects has been deleted.
3. A reference to [RFC7943] was added to Section 13.1 as it documents a method that might be used to generate addresses and was inadvertently missed when compiling [RFC8415].
4. Clarified the UDP ports used by clients, servers, and relay agents (Section 7.2).
5. Several additional RFCs have been referenced and editorial and reviews comments incorporated.

Appendix B. Appearance of Options in Message Types

The following tables indicate with a "*" the options that are allowed in each DHCP message type.

These tables are informational. If they conflict with text earlier in this document, that text should be considered authoritative.

	Client ID	Server ID	IA_NA	IA_PD	ORO	Pref	Elap. Time	Relay Msg.	Auth.
Solicit	*		*	*	*		*		
Advert.	*	*	*	*		*			
Request	*	*	*	*	*		*		
Confirm	*		*				*		
Renew	*	*	*	*	*		*		
Rebind	*		*	*	*		*		
Decline	*	*	*	*			*		
Release	*	*	*	*			*		
Reply	*	*	*	*					*
Reconf	*	*							*
Inform.	*	(see note)			*		*		

	Client ID	Server ID	IA_NA	IA_PD	ORO	Pref	Elap. Time	Relay Msg.	Auth.
R-forw.								*	
R-repl.								*	

Table 4: Appearance of Options in Message Types (Part 1 of 3)

NOTE: The Server Identifier option (see Section 21.3) is only included in Information-request messages that are sent in response to a Reconfigure (see Section 18.2.6).

	Status Code	Rap. Comm.	User Class	Vendor Class	Vendor Spec.	Inter. ID	Recon. Msg.	Recon. Accept	Info Refr. Time
Solicit		*	*	*	*			*	
Advert.	*		*	*	*			*	
Request			*	*	*			*	
Confirm			*	*	*				
Renew			*	*	*			*	
Rebind			*	*	*			*	
Decline			*	*	*				
Release			*	*	*				
Reply	*	*	*	*	*			*	*
Reconf.							*		
Inform.			*	*	*			*	
R-forw.					*	*			
R-repl.					*	*			

Table 5: Appearance of Options in Message Types (Part 2 of 3)

	SOL_MAX_RT	INF_MAX_RT
Solicit		
Advert.	*	

	SOL_MAX_RT	INF_MAX_RT
Request		
Confirm		
Renew		
Rebind		
Decline		
Release		
Reply	*	*
Reconf.		
Inform.		
R-forw.		
R-repl.		

Table 6: Appearance of Options in Message Types (Part 3 of 3)

Appendix C. Appearance of Options in the "options" Field of DHCP Options

The following table indicates with a "*" where options defined in this document can appear as top-level options or can be encapsulated in other options defined in this document. Other RFCs may define additional situations where options defined in this document are encapsulated in other options.

This table is informational. If it conflicts with text earlier in this document, that text should be considered authoritative.

	Top-Level	IA_NA	IAADDR	IA_PD	IAPREFIX	RELAY-FORW	RELAY-REPL
Client ID	*						
Server ID	*						
IA_NA	*						
IAADDR		*					

	Top-Level	IA_NA	IAADDR	IA_PD	IAPREFIX	RELAY-FORW	RELAY-REPL
IA_PD	*						
IAPREFIX				*			
ORO	*						
Preference	*						
Elapsed Time	*						
Relay Message						*	*
Authentic.	*						
Status Code	*	*		*			
Rapid Comm.	*						
User Class	*						
Vendor Class	*						
Vendor Info.	*					*	*
Interf. ID						*	*
Reconf. MSG.	*						
Reconf. Accept	*						
Info Refresh Time	*						
SOL_MAX_RT	*						
INF_MAX_RT	*						

Table 7: Appearance of Options

Notes: Options asterisked in the "Top-Level" column appear in the "options" field of client messages (see Section 8). Options asterisked in the "RELAY-FORW" and "RELAY-REPL" columns appear in the "options" field of the Relay-forward and Relay-reply messages (see Section 9).

Acknowledgments

This document is merely a refinement of earlier work as described in [RFC8415].

A number of additional people have contributed to identifying issues with [RFC8415] including Fernando Gont, Felix Hamme, Rene Engel, Esko Dijk, Jen Linkova, Tomoyuki Sahara, and Ted Lemon.

We thank the thorough and thoughtful reviewers during the IETF process, especially Mohamed Boucadair, Tim Chown, Roman Danyliw, Tatuya Jinmei, Jim Read, Ketan Talaulikar, Éric Vyncke, and Dave Worley. We also thank the DHC WG members for their reviews of this updated document.

And special thanks to Suresh Krishnan for shepherding this document through the IETF process.

Authors' Addresses

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

Bernie Volz
Individual Contributor
116 Hawkins Pond Road
Center Harbor, NH 03226
United States of America
Email: bevolz@gmail.com

Michael C. Richardson
Sandelman Software Works
470 Dawson Avenue
Ottawa ON K1Z 5V7
Canada
Email: mcr+ietf@sandelman.ca
URI: http://www.sandelman.ca/

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

Timothy Winters
QA Cafe
100 Main Street, Suite #212
Dover, NH 03820
United States of America
Email: tim@qacafe.com