Dynamic Host Configuration (DHC)
Internet-Draft
Obsoletes: 3315,3633,3736,4242,7083,7550
(if approved)
Intended status: Standards Track
Expires: December 30, 2017

T. Mrugalski, Ed.
M. Siodelski
ISC
B. Volz
A. Yourtchenko
Cisco
M. Richardson
SSW
S. Jiang
Huawei
T. Lemon
Nominum
T. Winters
UNH-IOL
June 28, 2017

Dynamic Host Configuration Protocol for IPv6 (DHCPv6) bis
draft-ietf-dhc-rfc3315bis-09

Abstract

This document describes 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 updates the text from RFC3315, the original DHCPv6 specification, and incorporates prefix delegation (RFC3633), stateless DHCPv6 (RFC3736), an option to specify an upper bound for how long a client should wait before refreshing information (RFC4242), a mechanism for throttling DHCPv6 clients when DHCPv6 service is not available (RFC7083), and clarifies the interactions between modes of operation (RFC7550). As such, this document obsoletes RFC3315, RFC3633, RFC3736, RFC4242, RFC7083, and RFC7550.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 30, 2017.

Copyright Notice

Table of Contents

1. Introduction

This document describes DHCP for IPv6 (DHCPv6), a client/server protocol that provides managed configuration of devices. 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, which are carried in options. DHCPv6 can be extended through the definition of new options to carry configuration information not specified in this document.

DHCPv6 also provides a mechanism for automated delegation of IPv6 prefixes using DHCPv6, originally specified in [RFC3633]. Through this mechanism, a delegating router can delegate prefixes to requesting routers. Use of this mechanism is specified as part of [RFC7084] and by [TR-187].

DHCPv6 can also provide only other configuration options (i.e., no addresses or prefixes). That implies that the server does not have to track any state, and thus this mode is called stateless DHCPv6. Mechanisms necessary to support stateless DHCPv6 are much smaller than to support stateful DHCPv6.

The remainder of this introduction summarizes the relationship to the previous DHCPv6 standards in Section 1.1 and clarifies the stance with regards to DHCPv4 in Section 1.2. Section 5 describes the message exchange mechanisms to illustrate DHCP operation rather than provide an exhaustive list of all possible interactions and Section 6 provides an overview of common operational models. Section 18 explains client and server operation in detail.

1.1. Relation to Previous DHCPv6 standards

The initial specification of DHCPv6 was defined in [RFC3315] and a number of follow up documents were published over the years: IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6 [RFC3633], Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6s [RFC3736], Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [RFC4242],

Modification to Default Values of SOL_MAX_RT and INF_MAX_RT [RFC7083], and Issues and Recommendations with Multiple Stateful DHCPv6 Options [RFC7550]. This document provides a unified, corrected, and cleaned up definition of DHCPv6 that also covers all errata filled against older RFCs. As such, it obsoletes a number of aforementioned RFCs. And, there are a small number of mechanisms that were obsoleted, listed in Section 25. Also see Appendix A.

1.2. Relation to DHCP in IPv4

The operational models and relevant configuration information for DHCPv4 ([RFC2131] and [RFC2132]) and DHCPv6 are sufficiently different that integration between the two services is not included in this document. [RFC3315] suggested that future work might be to extend DHCPv6 to carry IPv4 address and configuration information. However, the current consensus of the IETF 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 the DHCPv6 protocol for the dynamic provisioning of IPv4 address and configuration information.

Merging DHCPv4 and DHCPv6 configuration is out of scope of this document. [RFC4477] discusses some issues and possible strategies for running DHCPv4 and DHCPv6 services together. While this document is a bit dated, it provides a good overview of the issues at hand.

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 [RFC2119] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings, and are not to be interpreted as [RFC2119] key words.

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, so long as its external behavior is consistent with that described in this document.

3. Background

The IPv6 Specification provides the base architecture and design of IPv6. Related work in IPv6 that would best serve an implementer to study includes the IPv6 Specification [RFC2460], the IPv6 Addressing Architecture [RFC4291], IPv6 Stateless Address Autoconfiguration [RFC4862], and IPv6 Neighbor Discovery Processing [RFC4861]. These specifications enable DHCP to build upon the IPv6 work to provide robust stateful autoconfiguration.

The IPv6 Addressing Architecture specification [RFC4291] defines the address scope that can be used in an IPv6 implementation, and the various configuration architecture guidelines for network designers of the IPv6 address space. Two advantages of IPv6 are that support for multicast is required and nodes can create link-local addresses during initialization. The availability of these features means that a client can use its link-local address and a well-known multicast address to discover and communicate with DHCP servers or relay agents on its link.

IPv6 Stateless Address Autoconfiguration [RFC4862] specifies procedures by which a node may autoconfigure addresses based on router advertisements [RFC4861], and the use of a valid lifetime to support renumbering of addresses on the Internet. Compatibility with stateless address autoconfiguration is a design requirement of DHCP.

IPv6 Neighbor Discovery [RFC4861] is the node discovery protocol in IPv6 which replaces and enhances functions of ARP [RFC0826]. To understand IPv6 and stateless address autoconfiguration, it is strongly recommended that implementers understand IPv6 Neighbor Discovery.

4. Terminology

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

4.1. IPv6 Terminology

IPv6 terminology relevant to this specification from the IPv6 Protocol [RFC2460], IPv6 Addressing Architecture [RFC4291], and IPv6 Stateless Address Autoconfiguration [RFC4862] is included below.

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

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); PPP and PPPoE links; and Internet (or higher) layer "tunnels", such as tunnels over IPv4 or IPv6 itself.
link-layer identifier	A link-layer identifier for an interface. Examples include IEEE 802 addresses for Ethernet or Token Ring network interfaces, and E.164 addresses for ISDN links.
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 interface 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.

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 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, temporary). A binding containing configuration information for a client is indexed by <DUID>.

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 may contain IAADDR options).

delegating router — The router that acts as a DHCP server, and responds to requests for delegated prefixes. This document primarily uses the

term "DHCP server" or "server" when discussing the "delegating router" functionality of prefix delegation (see Section 1).

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 (or client) — A node that initiates requests on a link to obtain configuration parameters from one or more DHCP servers. Depending on the purpose of the client, it may feature the requesting router functionality, if it supports prefix delegation.

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

DHCP relay agent (or 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 (or server) — A node that responds to requests from clients, and may or may not be on the same link as the client(s). Depending on its capabilities, it may also feature the functionality of delegating router, if it supports prefix delegation.

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.

IA — Identity Association: A collection of leases assigned to a client. Each IA has an associated IAID. 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 temporary

addresses (IA_TA) holds temporary addresses (see "identity association for temporary addresses") and 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. At the time of writing this document, there are three IA types defined: IA_NA, IA_TA and IA_PD. New IA types may be defined in the future.

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, belonging to that client.

IA_NA — Identity association for Non-temporary Addresses: An IA that carries assigned addresses that are not temporary addresses (see "IA_TA").

IA_TA — Identity Association for Temporary Addresses: An IA that carries temporary addresses (see [RFC4941]).

IA_PD — Identity Association for Prefix Delegation: An IA that carries delegated prefixes.

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 7.3).

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

requesting router — The router that acts as a DHCP client and is requesting prefix(es) to be assigned. This document primarily uses the term "DHCP

client" or "client" when discussing the "requesting router" functionality of prefix delegation (see Section 1).

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 the top-level option or at any encapsulation level. Most options are singletons.

T1
: The time at which the client contacts the server from which the addresses in the IA_NA or prefixes in the IA_PD were obtained to extend the lifetimes of the addresses assigned to the IA_NA or prefixes delegated to the IA_PD.

T2
: The time at which the client contacts any available server to extend the lifetimes of the addresses assigned to the IA_NA or prefixes delegated to the IA_PD.

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 either by a client or server.

5. Client-Server Exchanges

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

A DHCP client sends most messages using a reserved, link-scoped multicast destination address 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 and the discussion of message exchanges in the remainder of this section will omit the description of message relaying by relay agents.

Once the client has determined the address of a server, it may under some circumstances send messages directly to the server using unicast.

5.1. Client-server Exchanges Involving Two Messages

When a DHCP client does not need to have a DHCP server assign it IP addresses or delegated prefixes, the client can obtain other configuration information such as a list of available DNS servers [RFC3646] or NTP servers [RFC4075] 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.

When a server has addresses and/or delegated prefixes and other configuration information committed to a client, the client and server may be able to complete the exchange using only two messages, instead of four messages as described in the next section. In this case, 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) 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.

There are additional two message exchanges between the client and server described later in this document.

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 confirmed assignment of addresses and/or delegated prefixes and other configuration information. The server responds with a Reply message that contains the confirmed addresses, delegated prefixes, and configuration.

As described in the previous section, the client can also 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. The client then performs the earlier described two message exchange. This can be used to expedite configuration changes to a client, such as the need to renumber a network (see [RFC6879]).

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 specific interface, belong to a single provisioning domain.

6.1. Stateless DHCP

Stateless DHCP [RFC3736] is used when DHCP is not used for obtaining a lease, but a node (DHCP client) desires one or more DHCP "other configuration" parameters, such as a list of DNS recursive name servers or DNS domain search lists [RFC3646]. Stateless may be used when a node initially boots or at any time the software on the 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 allocate 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 an associated preferred and valid lifetime, which constitutes 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 Section 6.6 and Section 12).

6.3. DHCP for Prefix Delegation

The prefix delegation mechanism, originally described in [RFC3633], is another stateful mode of operation and was originally intended for simple delegation of prefixes from a delegating router (DHCP server) to requesting routers (DHCP clients). It is appropriate for situations in which the delegating router does not have knowledge about the topology of the networks to which the requesting router is attached, and the delegating router does not require other information aside from the identity of the requesting router to choose a prefix for delegation. For example, these options would be used by a service provider to assign a prefix to a Customer Edge Router device acting as a router between the subscriber's internal network and the service provider's core network.

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

While [RFC3633] assumed that the DHCP client is a router (hence use of "requesting router") and that the DHCP server was a router (hence use of "delegating router"), 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 [RFC2460]. Also, in many cases (such as tethering or hosting virtual machines), hosts are already forwarding IP packets and thus operating as routers as defined in [RFC2460]. Therefore, this document mostly replaces "requesting router" with client and "delegating router" with server.

The model of operation for prefix delegation is as follows. A server is provided 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 valid and preferred lifetime, which constitutes 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.

This prefix delegation 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 in detail such use.

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

```
                      ______________________         \
                     /                      \         \
                    |    ISP core network    |         \
                     \__________ ___________/           |
                                |                       |
                        +-------+-------+               |
                        |  Aggregation  |               | ISP
                        |    device     |               | network
                        |  (delegating  |               |
                        |    router)    |               |
                        +-------+-------+               |
                                |                      /
                                |Network link to      /
                                |subscriber premises /
                                |
                         +------+------+             \
                         |     CPE     |              \
                         | (requesting |               \
                         |   router)   |                |
                         +----+---+----+                |
                              |   |                     | Subscriber
       ---+-------------+-----+   +-----+------         | Network
          |             |               |               |
     +----+-----+ +-----+----+     +----+-----+         |
     |Subscriber| |Subscriber|     |Subscriber|        /
     |    PC    | |    PC    |     |    PC    |       /
     +----------+ +----------+     +----------+      /
```

Figure 1: Prefix Delegation Network

In this example, the server (delegating router) 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 (requesting router) requests configuration information through DHCP. The DHCP messages from the client are received by the server in 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 (requesting router).

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 larger 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.

6.4. DHCP for Customer Edge Routers

The DHCP requirements and network architecture for Customer Edge Routers are described in [RFC7084]. 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. DHCP for Temporary Addresses

Temporary addresses were originally introduced to avoid privacy concerns with stateless address autoconfiguration, which based 64-bits of the address on the EUI-64 (see [RFC3041] and [RFC4941]). They were added to DHCP to provide complementary support when stateful address assignment is used.

Temporary address assignment works mostly like non-temporary address assignment (see Section 6.2), however these addresses are generally intended to be used for a short period of time and not to have their lifetimes extended, though they can be if required.

6.6. Multiple Addresses and Prefixes

The protocol 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 the client is attached to. 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 per each IA_NA option.

A client can explicitly request multiple addresses by sending multiple IA_NA (and/or IA_TA) options. A client can send multiple IA_NA (and/or IA_TA) options in its initial transmissions. Alternatively, it can send an extra Request message with additional new IA_NA (and/or IA_TA) options (or include them in a Renew message).

The same principle also applies to Prefix Delegation. In principle the protocol allows a client to request new prefixes to be delegated by sending additional IA_PD options. However, a typical operator usually prefers to delegate a single, larger prefix. In most deployments it recommended for the client to request 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 outside of scope of this document.

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

7. DHCP Constants

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

7.1. Multicast Addresses

DHCP makes use of the following multicast addresses:

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 which are within the site.

7.2. UDP Ports

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

7.3. DHCP Message Types

DHCP defines the following message types. More detail on these message types can be found in Section 8 and Section 9. Additional message types have been defined and may be defined in the future - see https://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xhtml. 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, 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 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 and some of these are defined later in Section 21. Additional options are defined in other documents or may be defined in the future.

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 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

This section presents a table of values used to describe the message transmission behavior of clients and servers.

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
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	32	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

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. 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 address or delegation of the prefix to a client and should only be used in cases were 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 or 4 byte boundaries.

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

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    msg-type   |               transaction-id                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                            options                            .
.                           (variable)                          .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Client/Server message format

msg-type — Identifies the DHCP message type; the available message types are listed in Section 7.3. A one octet long field.

transaction-id — The transaction ID for this message exchange. A three octets long 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 or 4 byte boundaries.

There are two relay agent messages, which share the following format:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    msg-type   |   hop-count   |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                         link-address                          |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
    |                               |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                         peer-address                          |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
    |                               |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    .                                                               .
    .            options (variable number and length)   ....        .
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 table defines the use of message fields in a Relay-forward message.

msg-type	RELAY-FORW (12). A one octet long field.
hop-count	Number of relay agents that have already relayed this message. A one octet long 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 unique address (including unique local address, [RFC4193]), but see discussion in Section 19.1.1. A 16 octets long field
peer-address	The address of the client or relay agent from which the message to be relayed was received. A 16 octets long field.
options	MUST include a "Relay Message option" (see Section 21.10); MAY include other options 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 how link-address is used.

9.2. Relay-reply Message

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

msg-type	RELAY-REPL (13). A one octet long field.
hop-count	Copied from the Relay-forward message. A one octet long field.
link-address	Copied from the Relay-forward message. A 16 octets long field.
peer-address	Copied from the Relay-forward message. A 16 octets long field.
options	MUST include a "Relay Message option"; see Section 21.10; MAY include other options. 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]. A domain name, or list of domain names, in DHCP MUST NOT be stored in compressed form, as described in section 4.1.4 of [RFC1035].

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 Section 21.2 and Section 21.3 for 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. And, such an attempt will be more and more unreliable as more clients adopt privacy measures, such as those defined in [RFC7844]. It is recommended to rely on the mechanism defined in [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. The stability of the DUID includes changes to virtual interfaces, such as logical PPP (over Ethernet) interfaces that may come and go in Customer Premise 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 two octets 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) - see [RFC6355]

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 two octets type field containing the value 1, a two octets hardware type code, four octets containing a time value, followed by 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 the IANA as described in [RFC0826]. Both the time and the hardware type are 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-LLT:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         DUID-Type (1)         |    hardware type (16 bits)    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        time (32 bits)                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                                                               .
    .             link-layer address (variable length)              .
    .                                                               .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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, and 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

This form of DUID is assigned by the vendor to the device. It consists of the four 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:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         DUID-Type (2)         |       enterprise-number       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   enterprise-number (contd)   |                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
 .                           identifier                          .
 .                       (variable length)                       .
 .                                                               .
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 manufacture process in case of physical devices or when the image is created or booted for the first time in case of virtual machines. 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:

```
 +---+---+---+---+---+---+---+---+
 | 0 | 2 | 0 | 0 | 0 |  9| 12|192|
 +---+---+---+---+---+---+---+---+
 |132|211| 3 | 0 | 9 | 18|
 +---+---+---+---+---+---+
```

Figure 6: DUID-EN example

This example includes the two octets type of 2, the Enterprise Number (9), followed by eight octets of identifier data (0x0CC084D303000912).

11.4. DUID Based on Link-layer Address, DUID-LL

This type of DUID consists of two octets containing the DUID type 3, a two octets 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 the IANA, as described in [RFC0826]. 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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         DUID-Type (3)         |    hardware type (16 bits)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.             link-layer address (variable length)              .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

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. 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 (UUID), 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.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         DUID-Type (4)         |        UUID (128 bits)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                                                               |
|                                -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

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 (i.e., IA_NA with IAID 0 is unique from IA_TA with IAID 0). 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 either by storing the IAID in non-volatile storage or by 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. Then the client 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 IA must be associated with exactly one interface.

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

The configuration information in an IA_TA consists of one or more IPv6 addresses. See Section 21.5 for the representation of an IA_TA 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. The lifetimes apply to the use of addresses, as described in 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 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 consists of one or more prefixes along with the T1 and T2 times for the IA_PD. See Section 21.21 for 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. The lifetimes apply to the use of delegated prefixes, as described in 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.

  * 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 not a link-local address, then the client is on the link identified by the source address in the IP datagram (note that this situation can occur only if the server has enabled the use of unicast message delivery by the client and the client has sent a message for which unicast delivery is allowed).

- The DUID supplied by the client.

- Other information in options supplied by the client, e.g., IA Address options 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 [RFC7721]. Server implementers are encouraged to review [RFC4941], [RFC7824], and [RFC7707] 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 Temporary Addresses

A client may request the assignment of temporary addresses (see [RFC4941] for the definition of temporary addresses). DHCP handling of address assignment is no different for temporary addresses.

Clients ask for temporary addresses and servers assign them. Temporary addresses are carried in the Identity Association for Temporary Addresses (IA_TA) option (see Section 21.5). Each IA_TA option contains at lease one temporary address for each of the prefixes on the link to which the client is attached.

The lifetime of the assigned temporary address is set in the IA Address option (see Section 21.6) encapsulated in the IA_TA option. It is RECOMMENDED to set short lifetimes, typically shorter than TEMP_VALID_LIFETIME and TEMP_PREFERRED_LIFETIME (see Section 5, [RFC4941]).

A DHCP server implementation MAY generate temporary addresses referring to the algorithm defined in Section 3.2.1, [RFC4941], with the additional condition that any new address is not the same as any assigned address.

The server MAY update the DNS for a temporary address, as described in section 4 of [RFC4941].

On the clients, by default, temporary addresses are preferred in source address selection, according to Rule 7, [RFC6724]. However, this policy is overridable.

One of the most important properties of a temporary address is to make it difficult to link the address to different actions over time. So, it is NOT RECOMMENDED for a client to renew temporary addresses, though DHCP provides for such a possibility (see Section 21.5).

13.3. 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; 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 care if 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 by not supplying this option.

A client may send some messages directly to a server using unicast, as described in Section 21.12.

14.1. Rate Limiting

In order to avoid prolonged message bursts that may be caused by possible logic loops, a DHCP client MUST limit the rate of DHCP messages it transmits. One example is that a client obtains an address or delegated prefix, but does not like the response; so it reverts back to Solicit procedure, discovers the same (sole) server, requests an address or delegated prefix and gets the same address or delegated prefix as before (as the server has this previously requested lease assigned to this 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, 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.

TRT Transmission Rate Limit

The Transmission Rate Limit parameter (TRT) SHOULD be configurable. A possible default could be 20 packets 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 are out of scope of this specification.

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

In certain cases, T1 and/or T2 time may be set to zero. Currently there are three such cases:

1. a client received an IA_NA option with a zero value

2. a client received an IA_PD option with a zero value

3. a client received an IA_TA option (which does not contain T1 and T2 fields and are not generally renewed)

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

When T1 and/or T2 times are set to zero, the client MUST choose a time to avoid packet storms. In particular, it MUST NOT transmit immediately. If the client received multiple IA containers, it SHOULD pick renew and/or rebind transmission times so all IA containers are handled in one exchange, if possible. The client MUST choose renew and rebind times to not violate rate limiting restrictions, 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 in this section.

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 the client successfully receives the appropriate response or responses from a server or servers, or when 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 the IA Address or IA Prefix options 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

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 include 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 are 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 a certain 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. Should a client or server receive messages which contain known options which are invalid for the message, this section explains how to process it. For example, an IA option is not allowed to appear in an Information-request message.

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

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

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

A server MUST discard any Solicit, Confirm, Rebind or Information-request messages it receives with a layer-3 unicast destination address.

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

16.1. Use of Transaction IDs

The "transaction-id" field holds a value used by clients and servers to synchronize 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.

- the message does not include a Client Identifier option.

- the contents of the Client Identifier option does 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.

- the contents of the Server Identifier option do not match the server's DUID.

- the message does not include a Client Identifier option.

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 or that do include a Server Identifier option.

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.

- the contents of the Server Identifier option does not match the server's identifier.

- the message does not include a Client Identifier option.

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 or that do include a Server Identifier option.

16.8. Decline Messages

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.

- the contents of the Server Identifier option does not match the server's identifier.

- the message does not include a Client Identifier option.

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.

- the contents of the Server Identifier option does not match the server's identifier.

- the message does not include a Client Identifier option.

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.

- 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 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; OR, 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.

- the message does not include a Client Identifier option that contains the client's DUID.

- the message does not include a Reconfigure Message option.

- 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 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

Client's behavior regarding interface selection is different depending on the purpose of the configuration.

17.1. Address, 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 which configuration is being requested for is a logical interface without direct link attachment or the client is certain that two interfaces are attached to the same link.

When a client sends a DHCP message directly to a server using unicast (after receiving the Server Unicast option from that server), the source address in the header of the IPv6 datagram MUST be an address assigned to the interface for which the client is interested in obtaining configuration and which is suitable for use by the server in responding to the client.

17.2. Address, 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 (ISP network). 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.

When a client sends a DHCP message directly to a server using unicast (after receiving the Server Unicast option from that server), the source address SHOULD be an address from the upstream interface and which is suitable for use by the server in responding to the client.

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 Router Advertisement indicates DHCPv6 is available for address configuration (see [RFC4861] Section 4.2),

4. as required to extend the lifetime of address(es) and/or delegated prefix(es), using Renew and Rebind messages,

5. or when requested to do so by a server - upon the receipt of a Reconfigure message.

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 it had obtained from the previous server (see Section 18.2.7 for more details on what stop using means), and which it is now trying to obtain from a new server.

A DHCP client that does not need to have a DHCP server assign it IP addresses or delegated prefixes, can obtain configuration information such as a list of available DNS servers [RFC3646] or NTP servers [RFC4075] 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 which respond to the client's message and 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 this 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 is a combination of the 4-message exchange (to select a server and assign the lease(s) to the client) followed by two 2-message exchanges (to extend the lifetime on the lease(s) and eventually release the lease(s)).

```
             Server                          Server
         (not selected)      Client        (selected)

               v               v               v
               |               |               |
               |     Begins initialization     |
               |               |               |
  start of     | _____________/|\_____________ |
  4-message    |/ Solicit      | Solicit      \|
  exchange     |               |               |
           Determines          |          Determines
          configuration        |         configuration
               |               |               |
               |\              |  ____________/|
               | \________     | /Advertise    |
               | Advertise\    |/              |
               |           \   |               |
               |      Collects Advertises      |
               |             \ |               |
               |     Selects configuration     |
               |               |               |
               | _____________/|\_____________ |
               |/ Request      |  Request     \|
               |               |               |
               |               |     Commits configuration
               |               |               |
  end of       |               | _____________/|
  4-message    |               |/ Reply        |
```

```
     exchange     |                |                |
                  |    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         |
                  |                |                |
                  v                v                v
```

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

The client and server use the IA_PD option to exchange information about prefix(es) in much the same way as IA_NA and IA_TA options are used for assigned addresses. Typically, a single DHCP session is used to exchange information about addresses and prefixes, i.e., IA_NA and IA_PD options are carried in the same message.

Clients SHOULD use a single transaction for all of its IA options. Servers SHOULD assign the same T1/T2 values to all IA options configured for a client, so the client will generate a single transaction when renewing its leases. For a rationale of this approach, see Section 4.2 of [RFC7550].

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 life cycle of addresses and delegated prefixes. When a client detects 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 [RFC7550]. 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 for the client to 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 an Information-request message as indicated by the Reconfigure Message option. 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.

If the client has a source address of sufficient scope that can be used by the server as a return address, and the client has received a Server Unicast option Section 21.12 from the server, the client SHOULD unicast any Request, Renew, Release and Decline messages to the server.

Use of unicast may avoid delays due to the relaying of messages by relay agents, as well as avoid overhead on servers due to the delivery of client messages to multiple servers. However, requiring the client to relay all DHCP messages through a relay agent enables the inclusion of relay agent options in all messages sent by the

client. The server should enable the use of unicast only when relay agent options will not be used.

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 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 to request the assignment of non-temporary addresses, IA_TA options to request the assignment of temporary addresses, and IA_PD options to request prefix delegation. Either IA_NA, IA_TA or IA_PD options, or a combination of all, 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 encapsulated within IA_NA and IA_TA options as hints to the server about the addresses for which the client has a preference.

The client MAY include values in IA Prefix options 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 (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 mechanism is essential for devices that are not battery powered, as they may suffer from power failure. After recovering from a power outage, many devices may start their transmission at the same time. This is much less of a concern for battery powered devices. This random delay also helps descynronize clients which start DHCP session by seeing it available in Router Advertisement messages (see [RFC4861] Section 4.2).

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 2-message exchange includes a Rapid Commit option 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 either:

- processes the Reply message as described in Section 18.2.10, and discards any Reply messages received in response to the Request message, or

- 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 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 begins 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 to be 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 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.

The client MUST include a Client Identifier option 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:

- 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 is 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 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 and the preferred-lifetime and valid-lifetime fields in the IA Address options 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 valid and preferred 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, which includes IA options for the IAs whose lease lifetimes are to be extended. The client includes IA Address options within IA_NA and IA_TA options for the addresses assigned to the IAs. The client includes IA Prefix options within IA_PD options 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 parameters assigned to an IA. However, as the client Renews/ Rebinds all IAs from the server at the same time, the client MUST select a T1 and T2 time from all IA options, which will guarantee that the client will send 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.

If T1 or T2 had been set to 0 by the server (for an IA_NA or IA_PD) or there are no T1 or T2 times (for an IA_TA) in a previous Reply, the client may send a Renew or Rebind message, respectively, at the client's discretion. 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 in the Renew message, identifying the server with which the client most recently communicated.

The client MUST include a Client Identifier option 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 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 IA Prefix option with the IPv6-prefix field set to 0 unless it is supplying a hint for the prefix length.

The client includes 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 earliest time T2 is reached. If the client is responding to a Reconfigure, the client ignores and discards the Reconfigure message. In this case, the client continues to operate as if Reconfigure message was not received, i.e., it uses T1/T2 times associated with the client's leases to determine when it should send Renew or Rebind to the server. server. The client begins a Rebind message exchange (see Section 18.2.5) when the earliest time T2 is reached.

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 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 ignores and discards the Reconfigure message.

18.2.6. Creation and Transmission of Information-request Messages

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

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 to identify itself to the server (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 includes a Server Identifier option 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 places the identifier of the server that allocated the lease(s) in a Server Identifier option.

The client MUST include a Client Identifier option 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 zero in a Router Advertisement message as described in (e) of Section 5.5.3 of [RFC4862] - also see 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 and thus the client MUST NOT include IA_PD options 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 places the identifier of the server that allocated the address(es) in a Server Identifier option.

The client MUST include a Client Identifier option 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 in 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 as a result of the conflict, to be equivalent to not having received the binding, insofar as 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 are preferred over all other Advertise messages.

- 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.

- 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.

Once a client has selected Advertise message(s), the client will typically store information about each server, such as 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 per 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 SOL_MAX_RT and INF_MAX_RT options in an Advertise message, even if the message contains a Status Code option 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 encapsulated in IA_NA or IA_TA options) and no delegated prefixes (IA Prefix options encapsulated in IA_PD options) 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 display any associated status message(s) to the user or activity log.

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), Request, Confirm, Renew, Rebind, or Information-request message, the client extracts the top-level Status Code option if present.

The client MUST process SOL_MAX_RT and INF_MAX_RT options 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).

If the client receives a Reply message with a status code of UseMulticast, the client records the receipt of the message and sends subsequent messages to the server through the interface on which the message was received using multicast. The client resends the original message using multicast.

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.

When a client received a configuration option in an earlier Reply, then sends a Renew, Rebind or Information-request and the requested option is not present in the 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 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 counter example, a client that previously received an NTP server address from the DHCP server and does not receive it any more, 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 containers, as their processing is described in detail in the next section.

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) or a Request, the client can either re-issue the message without specifying any addresses or restart the DHCP server discovery process (see Section 18).

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:

- Record T1 and T2 times, 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 [RFC4862] Section 5.4, 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, which does not have any associated reachability information, 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_NA or IA_TA. Addresses obtained from IA_NA or IA_TA 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, with the following exception: the client MUST NOT advertise any delegated prefixes or subnets from the delegated prefix(es) to the link through which it received the DHCP message from the server (see [RFC6603] for exceptions).

  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 larger 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 valid and preferred 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 if any of the IAs in the Reply message contains the NoBinding status code to the server that responded. 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 in this case the client MUST limit the rate at which it sends these messages, to avoid the Renew/Rebind storm.

- Otherwise accepts the information in the IA.

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 all the addresses and delegated prefixes for which it has bindings and try to obtain all required leases from the new server. This facilitates the client using a single state machine for all bindings.

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(s) 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 success status (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 Replies 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.

18.2.11. Receipt of Reconfigure Messages

A client receives Reconfigure messages sent to the 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.

Upon receipt of a valid Reconfigure message, the client responds with either a Renew message, a Rebind message, or an Information-request message as indicated by the Reconfigure Message option (as defined in 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 trigger the exchange (if the first Reconfigure was not received by the client) or will 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) to arrive 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:

- o The client reboots (and has stable storage and persisted DHCP state).

- o The client is reconnected to a link on which it has obtained leases.

- o The client returns from sleep mode.

- o The client changes access points (such as if using a wireless technology).

When the client detects one of the above conditions and it has obtained addresses and no delegated prefixes from a server, the client SHOULD initiate a Confirm/Reply message exchange. The client includes 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, IA_TAs, 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 a Information-Request/Reply message exchange as described in Section 18.2.6.

If not associated with one of the above mentioned conditions, a client MAY initiate a Renew/Reply exchange (as if the T1 time expired) as described in Section 18.2.4 or an Information-Request/ Reply exchange as described in Section 18.2.6 if the client detects a significant change regarding the prefixes available on the link (when new are added or existing are deprecated) as this may indicate a configuration change. However, a client MUST rate limit such 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).

18.3. Server Behavior

For this discussion, the Server is assumed to have been configured in an implementation specific manner with configuration 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 a 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, when the server is configured to respond with committed lease assignments.

These Advertise and Reply messages MUST always contain the Server Identifier option containing the server's DUID and the Client Identifier option 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 packet sizes and the use of fragmentation in section 5 of [RFC2460]. If the client included an Option Request option 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.

The server MAY initiate a configuration exchange, by sending Reconfigure messages, 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 (as defined in 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

See Section 18.4 for handling Solicit message received via unicast. Unicast transmission of Solicit is not allowed, regardless if Server Unicast option is configured or not.

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 the server is permitted to respond to the client, the client has not included a Rapid Commit option in the Solicit message or the server has not been configured to respond with committed assignment 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 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

See Section 18.4 for handling Request message received via unicast.

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 a Server Identifier option containing the server's DUID and the Client Identifier option from the Request message in the Reply message.

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

For each IA_NA and IA_TA 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 is 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 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 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_NA and IA_TA), prefixes (for IA_PD) 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 which 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 if the server policy enables reconfigure mechanism and the client supports it. Currently sending this option in Reply is technically redundant, as the use of the reconfiguration mechanism requires authentication and currently the only defined one is the Reconfigure Key Authentication Protocol (see Section 20.4) and the presence of the reconfigure key signals support for Reconfigure acceptance. However, there may be better security mechanisms defined in the future that would cause RKAP to not be used anymore. One of such mechanisms being worked on is mentioned in Section 22.

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 matches a previous transmission. The server MUST NOT cache its responses.

18.3.3. Receipt of Confirm Messages

See Section 18.4 for handling Confirm message received via unicast. Unicast transmission of Confirm is not allowed, regardless if Server Unicast option is configured or not.

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.

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 a Server Identifier option containing the server's DUID and the Client Identifier option from the Confirm message in the Reply message. The server includes a Status Code option indicating the status of the Confirm message.

18.3.4. Receipt of Renew Messages

See Section 18.4 for handling Renew message received via unicast.

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 back the IA 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 or IA Prefix option for this address or 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 to the valid lifetime for the IA option unless the server also includes other addresses or delegated prefixes which the server is able to extend for the IA. Setting T1 and T2 to values

equal to 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 with zero value in the "IPv6 prefix" field and 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 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 a Server Identifier option containing the server's DUID and the Client Identifier option from the Renew message in the Reply 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 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.

18.3.5. Receipt of Rebind Messages

See Section 18.4 for handling Rebind message received via unicast. Unicast transmission of Rebind is not allowed, regardless if Server Unicast option is configured or not.

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 back the IA to the client with new lifetimes and, if applicable, T1/T2 times. 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 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 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 the address or delegated prefix 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_NA and IA_TA) or delegated prefixes (for IA_PD). 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 which is not appropriate for the link to which the client's interface is attached, the server MAY include the IA Address option with the lifetimes of 0. For any delegated prefix which is not appropriate for the link to which the client's interface is attached, the server MAY include the IA Prefix option with the 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 about the Rapid Commit option below), the server SHOULD create a binding and return the IA with allocated leases with lifetimes and, if applicable, T1/T2 times and other information requested by the client. The server MUST NOT return any addresses or delegated prefixes in the IA which 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 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 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 a Server Identifier option containing the server's DUID and the Client Identifier option from the Rebind message in the Reply 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/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.

18.3.6. Receipt of Information-request Messages

See Section 18.4 for handling Information-request message received via unicast.

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 containing the server's DUID in the Reply message. If the client included a Client Identification option 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

See Section 18.4 for handling Release message received via unicast.

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 IA, 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 with value Success, a Server Identifier option with the server's DUID, and a Client Identifier option 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

See Section 18.4 for handling Decline message received via unicast.

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 IA (though it may choose to log an error if it finds such an address).

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 client includes a Status Code option with the value Success, a Server Identifier option with the server's DUID, and a Client Identifier option 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 and copies the Client Identifier option from the Solicit message into the Advertise message.

The server MAY add a Preference option 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 zero unless otherwise configured by the server administrator.

The server includes a Reconfigure Accept option if the server wants to indicate it supports 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 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 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 option in the IA_PD, the server MAY use the prefix contained in the IPv6-prefix field and/or the prefix length contained in the "prefix-length" field as a 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 (IA_NA or IA_TA) in subsequent Request from the client, the server MUST include the IA in the Advertise message with no addresses in the IA and a Status Code option encapsulated in the IA containing status code NoAddrsAvail.

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

Transmission of the Advertise message 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, 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. The server sets the transaction-id field to 0. The server includes a Server Identifier option containing its DUID and a Client Identifier option containing the client's DUID in the Reconfigure message.

Because of the risk of denial of service 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.

The server MUST include a Reconfigure Message option (defined in 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 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, 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, a Rebind/Reply, or Information-request/Reply message exchange with the server. The server interprets the receipt of a Renew, a Rebind, or Information-request message (whichever was specified in the

original Reconfigure message) from the client as satisfying the Reconfigure message request.

If the server does not receive a Renew, Rebind, or Information-request message from the client in REC_TIMEOUT milliseconds, the server retransmits the Reconfigure message, doubles the REC_TIMEOUT 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.

18.4. Reception of Unicast Messages

Unless otherwise stated in sections dedicated to specific messages reception (see dedicated sections in Section 18.3), the server is not supposed to accept unicast traffic when it is not explicitly configured to do so. For some messages (Solicit, Rebind, and Confirm) unicast transmission is not allowed, even if Server Unicast option is configured. For Request, Renew, Informaton-request, Release, and Decline messages, it is allowed only if Server Unicast option is configured.

When the server receives a message via unicast from a client to which the server has not sent a Server Unicast option (or is not currently configured to send a Server Unicast option to the client), the server discards that message and responds with an Advertise (when responding to Solicit) or Reply (when responding to any other messages) message containing a Status Code option with value UseMulticast, a Server Identifier option containing the server's DUID, the Client Identifier option from the client message (if any), and no other options.

19. Relay Agent Behavior

The relay agent SHOULD be configured to use a list of destination addresses, which include 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 32.

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 them 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 valid message (for a definition of a valid message, see Section 4.1 of [RFC7283]) to be relayed, 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 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 global address (including unique local address, [RFC4193]) with 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 the original message was received on. That is not recommended as it may require additional information to 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 in the Relay-forward message is set to 0.

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 in the earlier paragraphs 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 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 from the relay agent 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 global address (including unique local address, [RFC4193]), the relay agent sets the link-address field to 0; otherwise the relay agent sets the link-address field to a global address (including unique local address) assigned to the interface on which the message was received, or includes an Interface-ID option 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 described earlier in this section.

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.

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, 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 zero, 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 return a response to a client if the original message from the client was relayed to the server in a Relay-forward message or to 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 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-REPLY
hop-count:      1
link-address:   0
peer-address:   A
Relay Message option, containing:
  msg-type:     RELAY-REPLY
  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.

20. Authentication of DHCP Messages

Within this document, two security mechanisms are introduced for the authentication of DHCP messages: authentication (and encryption) of messages sent between servers and relay agents using IPsec, and protection against misconfiguration of a client caused by a Reconfigure message sent by a malicious DHCP server.

The delayed authentication protocol, defined in [RFC3315], has been obsoleted by this document (see Section 25).

20.1. Security of Messages Sent Between Servers and Relay Agents

Relay agents and servers that exchange messages can use IPsec as detailed in [I-D.ietf-dhc-relay-server-security].

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, the Reconfigure key authentication protocol, 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 replay detection method (RDM) field specifies the type of replay detection used in the replay detection field.

20.3. Replay Detection

The Replay Detection Method (RDM) field 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 counter.

Using a counter value, such as the current time of day (for example, an NTP-format timestamp [RFC5905]), can reduce the danger of replay attacks. This method MUST be supported by all protocols.

20.4. Reconfigure Key Authentication Protocol

The Reconfigure key authentication protocol 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 an 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. The format of the Authentication information is defined in the following section.

The Reconfigure Key protocol 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 the Reconfigure Key Authentication Protocol

The following fields are set in an Authentication option for the Reconfigure Key Authentication Protocol:

```
protocol   3

algorithm  1

RDM        0
```

The format of the authentication information for the Reconfigure Key Authentication Protocol is:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |                 Value (128 bits)              |
    +-+-+-+-+-+-+-+-+                                               |
    .                                                               .
    .                                                               .
    .                                               +-+-+-+-+-+-+-+-+
    |                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: RKAP Authentication Information

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

- 1 — Reconfigure Key value (used in Reply message).
- 2 — HMAC-MD5 digest of the message (used in Reconfigure message).

A one octet long field.

Value — Data as defined by the Type field. A 16 octets long field.

20.4.2. Server Considerations for Reconfigure Key Authentication Protocol

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 in the Reply message.

The Reconfigure Key is 128 bits long, and MUST be a cryptographically strong random or pseudo-random 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 zero 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 Reconfigure Key Authentication Protocol

The client will receive a Reconfigure Key from the server in an Authentication option 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 DHCP Reconfigure message, 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 describes the DHCP options defined as part of the base DHCP specification. Other options may be defined in the future in separate documents. See [RFC7227] for guidelines regarding new options definition. See Section 24 for additional information about a 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 IA_NA (see Section 21.4), IA_TA (see Section 21.5), 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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          option-code          |           option-len          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          option-data                          |
|                      (option-len octets)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: Option Format

option-code — An unsigned integer identifying the specific option type carried in this option. A two octets long field.

option-len — An unsigned integer giving the length of the option-data field in this option in octets. A two octets long 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 Section 21.4 and Section 21.6.

21.2. Client Identifier Option

The Client Identifier option is used to carry a DUID (see Section 11) identifying a client between a client and a server. The format of the Client Identifier option is:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |        OPTION_CLIENTID        |          option-len           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .                                                               .
 .                              DUID                             .
 .                        (variable length)                      .
 .                                                               .
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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) identifying a server between a client and a server. The format of the Server Identifier option is:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |        OPTION_SERVERID        |          option-len           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .                                                               .
 .                              DUID                             .
 .                        (variable length)                      .
 .                                                               .
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 option (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.

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

The format of the IA_NA option is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          OPTION_IA_NA         |          option-len           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IAID (4 octets)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              T1                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              T2                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                         IA_NA-options                         .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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_TA and IA_PD). A four octets long field.

T1 — The time at which the client should contact the server from which the addresses in the IA_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 four octets long field.

T2	The time at 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 four octets long field.
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 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 to.

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

The server selects the T1 and T2 times 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 .5 and .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 T1 and T2 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.

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

21.5. Identity Association for Temporary Addresses Option

The Identity Association for the Temporary Addresses (IA_TA) option is used to carry an IA_TA, the parameters associated with the IA_TA and the addresses associated with the IA_TA. All of the addresses in this option are used by the client as temporary addresses, as defined in [RFC4941]. The format of the IA_TA option is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          OPTION_IA_TA         |          option-len           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IAID (4 octets)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                          IA_TA-options                        .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: Identity Association for Temporary Addresses Option Format

option-code   OPTION_IA_TA (4).

option-len   4 + length of IA_TA-options field.

IAID   The unique identifier for this IA_TA; the IAID must be unique among the identifiers for

all of this client's IA_TAs. The number space for IA_TA IAIDs is separate from the number space for other IA option types (i.e., IA_NA and IA_PD). A four octets long field.

IA_TA-options — Options associated with this IA_TA. A variable length field (4 octets less than the value in the option-len field).

The IA_TA-Options field encapsulates those options that are specific to this IA_TA. For example, all of the IA Address options carrying the addresses associated with this IA_TA are in the IA_TA-options field.

Each IA_TA carries one "set" of temporary addresses. It is up to the server policy to determine how many addresses are assigned.

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

The status of any operations involving this IA_TA is indicated in a Status Code option in the IA_TA-options field.

Note that an IA has no explicit "lifetime" or "lease length" of its own. When the valid lifetimes of all of the addresses in an IA_TA have expired, the IA can be considered as having expired.

An IA_TA option does not include values for T1 and T2. A client MAY request that the valid lifetime on temporary addresses be extended by including the addresses in a IA_TA option sent in a Renew or Rebind message to a server. For example, a client would request an extension on the valid lifetime of a temporary address to allow an application to continue to use an established TCP connection. Extending only the valid, but not the preferred lifetime means the address will end up in deprecated state eventually. Existing connections could continue, but no new ones would be created using that address.

The client obtains new temporary addresses by sending an IA_TA option with a new IAID to a server. Requesting new temporary addresses from the server is the equivalent of generating new temporary addresses as described in [RFC4941]. The server will generate new temporary addresses and return them to the client. The client should request new temporary addresses before the lifetimes on the previously assigned addresses expire.

A server MUST return the same set of temporary address for the same IA_TA (as identified by the IAID) as long as those addresses are still valid. After the lifetimes of the addresses in an IA_TA have expired, the IAID may be reused to identify a new IA_TA with new temporary addresses.

21.6. IA Address Option

The IA Address option is used to specify an address associated with an IA_NA or an IA_TA. The IA Address option must be encapsulated in the Options field of an IA_NA or IA_TA option. The IAaddr-options fields encapsulates those options that are specific to this address.

The format of the IA Address option is:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          OPTION_IAADDR        |          option-len           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                         IPv6-address                          |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      preferred-lifetime                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        valid-lifetime                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                                                               .
    .                        IAaddr-options                         .
    .                                                               .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17: 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. And, a client MUST NOT assume any length of prefix that matches this address is on-link (see [RFC7421]). A 16 octets long field.

preferred-lifetime The preferred lifetime for the address in the option, expressed in units of seconds. A four octets long field.

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

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 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 and valid lifetime fields for the preferred and valid lifetimes. The values in the preferred and valid lifetimes 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, the valid lifetime of an address 0xffffffff is taken to mean "infinity" and should be used carefully.

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

The status of any operations involving this IA Address is indicated in a Status Code option in the IAaddr-options field, as specified in Section 21.13.

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:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           OPTION_ORO          |           option-len          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    requested-option-code-1    |    requested-option-code-2    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                              ...                              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18: 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. The length, in octets, is specified by option-len.

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. For certain message types, some option codes MUST be included in the Option Request option, see Table 1 for details.

The Option Request option MUST NOT include the following options: Server Identifier, Client Identifier, IA_NA, IA_PD, IA_TA, IA Address, IA Prefix, Option Request, Elapsed Time, Preference, Relay Message, Authentication, Server Unicast, Rapid Commit, User Class, Vendor Class, Interface-Id, Reconfigure Message, and Reconfigure Accept. Other top-level options MUST appear in the Option Request option or they will not be sent by the server. Only top-level options MAY appear in the Option Request option. Options 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 which specify exceptions to this restriction on including encapsulated options 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 Table 1.

21.8. Preference Option

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

The format of the Preference option is:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |       OPTION_PREFERENCE       |          option-len           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  pref-value   |
    +-+-+-+-+-+-+-+-+
```

Figure 19: Preference Option Format

option-code   OPTION_PREFERENCE (7).

option-len   1.

pref-value   The preference value for the server in this message. A one octet long field.

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 the use of the Preference option by the client and the interpretation of Preference option data value.

21.9. Elapsed Time Option

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      OPTION_ELAPSED_TIME      |           option-len          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          elapsed-time         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 20: 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 two octets long field.

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 controlling 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 an unsigned, 16 bit 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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        OPTION_RELAY_MSG       |           option-len          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                       DHCP-relay-message                      .
.                                                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 21: Relay Message Option Format

```
option-code           OPTION_RELAY_MSG (9).

option-len            Length of DHCP-relay-message.

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 delayed authentication protocol, defined in [RFC3315], has been obsoleted by this document, due to lack of usage. The format of the Authentication option is:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          OPTION_AUTH          |          option-len           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   protocol    |   algorithm   |      RDM      |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    |          replay detection (64 bits)           +-+-+-+-+-+-+-+-+
    |                                               |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    .                   authentication information                  .
    .                       (variable length)                       .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 22: 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 one octet long field.

algorithm — The algorithm used in the authentication protocol. A one octet long field.

RDM — The replay detection method used in this authentication option. A one octet long field.

Replay detection — The replay detection information for the RDM. A 64-bit (8 octets) long 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 option-len).

21.12. Server Unicast Option

The server sends this option to a client to indicate to the client that it is allowed to unicast messages to the server. The format of the Server Unicast option is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          OPTION_UNICAST       |        option-len             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                       server-address                          |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 23: Server Unicast Option Format

option-code   OPTION_UNICAST (12).

option-len   16.

server-address   The 128-bit address to which the client should send messages delivered using unicast.

The server specifies the address to which the client is to send unicast messages in the server-address field. When a client receives this option, where permissible and appropriate, the client sends messages directly to the server using the address specified in the server-address field of the option.

When the server sends a Unicast option to the client, some messages from the client will not be relayed by Relay Agents, and will not include Relay Agent options from the Relay Agents. Therefore, a server should only send a Unicast option to a client when Relay Agents are not sending Relay Agent options. A DHCP server rejects any messages sent inappropriately using unicast to ensure that messages are relayed by Relay Agents when Relay Agent options are in use.

Details about when the client may send messages to the server using unicast are in Section 18.

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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       OPTION_STATUS_CODE      |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          status-code          |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
.                                                               .
.                        status-message                         .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 24: Status Code Option Format

option-code   OPTION_STATUS_CODE (13).

option-len   2 + length of status-message.

status-code   The numeric code for the status encoded in this option. A two octets long field.

status-message   A UTF-8 encoded text string suitable for display to an end user, which MUST NOT be null-terminated. A variable length field (2 octets less than the value in option-len).

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 previously defined by [RFC3315] and [RFC3633] are:

Name	Code	Description
Success	0	Success.
UnspecFail	1	Failure, reason unspecified; this status code is sent by either a client or a server to indicate a failure not explicitly specified in this document.
NoAddrsAvail	2	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.
NoPrefixAvail	6	Server has no prefixes available to assign to the IA_PD(s).

See Section 24 for additional information about the registry maintained by IANA with the complete 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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      OPTION_RAPID_COMMIT      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 25: 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, and 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, some servers will commit leases that are not actually used by the client, which could result in bad information in the DNS server if the DHCP server updates DNS [RFC4704] or in response to leasequery requests [RFC5007].

The problem of unused leases can be minimized by designing the DHCP service so that only one server responds to the Solicit, by using relatively short lifetimes for newly assigned leases, or by having DHCP clients release unused leases using the Release message.

21.15. User Class Option

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

The format of the User Class option is:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |       OPTION_USER_CLASS       |          option-len           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                                                               .
    .                          user-class-data                      .
    .                                                               .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 26: 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. Each instance of the user class data is formatted as follows:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+-+-+-+-+-+-+
|        user-class-len         |          opaque-data          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+-+-+-+-+-+-+
```

Figure 27: User Class Data Format

The user-class-len is two 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 includes 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:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      OPTION_VENDOR_CLASS      |           option-len          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                       enterprise-number                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                                                               .
    .                       vendor-class-data                       .
    .                             . . .                             .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 28: 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 registered with IANA [IANA-PEN]. A four octets long field.

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 option-len).

The vendor-class-data 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 the vendor-class-data is formatted as follows:

```
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+-+-+-+-+-+-+
    |       vendor-class-len        |          opaque-data          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+-+-+-+-+-+-+
```

Figure 29: Vendor Class Data Format

The vendor-class-len is two 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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      OPTION_VENDOR_OPTS       |           option-len          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       enterprise-number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                       vendor-option-data                      .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 30: Vendor-specific Information Option Format

option-code OPTION_VENDOR_OPTS (17).

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

enterprise-number The vendor's registered Enterprise Number as registered with IANA [IANA-PEN]. A four octets long field.

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 option-len).

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 device's IPv6 stack to be functional.

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

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          sub-opt-code         |         sub-option-len        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                                                               .
    .                        sub-option-data                        .
    .                                                               .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 31: Vendor-specific Options Format

sub-opt-code — The code for the sub-option. A two octets long field.

sub-option-len — An unsigned integer giving the length of the sub-option-data field in this sub-option in octets. A two octets long field.

sub-option-data — The data area for the sub-option. The length, in octets, is specified by sub-option-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 Vendor-specific Information option with the same Enterprise Number. Each instance of Vendor-specific Information option MAY contain multiple sub-options.

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.

- 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 packet 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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      OPTION_INTERFACE_ID      |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                         interface-id                          .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 32: 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 for parameter assignment policies. The Interface-ID SHOULD be considered an opaque value, with policies based on exact match only; that is, the Interface-ID 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 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 this option is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      OPTION_RECONF_MSG        |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    msg-type   |
+-+-+-+-+-+-+-+-+
```

Figure 33: Reconfigure Message Option Format

option-code   OPTION_RECONF_MSG (19).

option-len   1.

msg-type   5 for Renew message, 6 for Rebind, 11 for Information-request message. A one octet long field.

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. The default behavior, in the absence of this option, means unwillingness to accept Reconfigure messages, or instruction not to accept Reconfigure messages, for the client and server messages, respectively. The following figure gives the format of the Reconfigure Accept option:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     OPTION_RECONF_ACCEPT      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 34: 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 this option is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         OPTION_IA_PD          |         option-length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IAID (4 octets)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              T1                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              T2                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                          IA_PD-options                        .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 35: Identity Association for Prefix Delegation Option Format

option-code OPTION_IA_PD (25).

option-length	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 and IA_TA). A four octets long field.
T1	The time at 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 current time expressed in units of seconds. A four octets long field.
T2	The time at which the client should contact any available server to extend the lifetimes of the prefixes assigned to the IA_PD; T2 is a time duration relative to the current time expressed in units of seconds. A four octets long field.
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 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 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 a IA_PD have expired, the IA_PD can be considered as having expired. T1 and T2 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 parameters, unless those values in those fields are 0. The values in the T1 and T2 fields are the number of seconds until T1 and T2.

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 .5 and .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 client 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.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        OPTION_IAPREFIX        |         option-length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      preferred-lifetime                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        valid-lifetime                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| prefix-length |                                               |
+-+-+-+-+-+-+-+-+          IPv6-prefix                          |
|                           (16 octets)                         |
|                                                               |
|                                                               |
|                                                               |
|               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |                                               .
+-+-+-+-+-+-+-+-+                                               .
.                       IAprefix-options                        .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 36: IA Prefix Option Format

option-code OPTION_IAPREFIX (26).

option-length 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. A four octets long field.

valid-lifetime The valid lifetime for the prefix in the option, expressed in units of seconds. A value of 0xFFFFFFFF represents infinity. A four octets long field.

prefix-length Length for this prefix in bits. A one octet long field.

IPv6-prefix An IPv6 prefix. A 16 octets long 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 and valid lifetime fields SHOULD be set to 0. The server MUST ignore any received values in these lifetime fields.

A client may set the IPv6-prefix field to zero and a given value in the prefix-length field to indicate a preference for the size of the prefix to be delegated.

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

The values in the preferred and valid lifetimes are the number of seconds remaining for 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 preferred and valid lifetime values of 0xffffffff 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 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 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:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |OPTION_INFORMATION_REFRESH_TIME|         option-len            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   information-refresh-time                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 37: 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 four octets long field.

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 ORO in any other messages.

A server sending a Reply to an Information-Request message SHOULD include this option if it is requested in the ORO of the Information-Request. The option value MUST NOT be smaller than IRT_MINIMUM. This option MUST only appear in the top-level option 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 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 longer value for SOL_MAX_RT, which 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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          option-code          |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       SOL_MAX_RT value                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 38: 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 range: 60 <= "value" <= 86400 (1 day). A four octets long field.

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, not as an encapsulated option in, e.g., an IA_NA, IA_TA, or IA_PD option.

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 Section 15 and Section 18.2.1.

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

Updated SOL_MAX_RT value applies only to the network interface on which the client received 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 longer value for INF_MAX_RT, which 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:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          option-code          |         option-len            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                       INF_MAX_RT value                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 39: 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 range: 60 <= "value" <= 86400 (1 day). A four octets long field.

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 Section 15 and Section 18.2.6.

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

22. Security Considerations

This section discusses security considerations that are not related to privacy. For dedicated privacy discussion, see Section 23. The work in progress [I-D.ietf-dhc-sedhcpv6] document provides additional analysis of the security issues and specifies a secure client and server communication mechanism.

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.

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 denial of service attack through misconfiguration of the client that causes 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 and INF_MAX_RT options, which may cause an undue delay in a client completing its DHCP protocol transaction in the case no other valid response is received. Assuming 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.

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

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.

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.

The Reconfigure Key protocol described in Section 20.4 provides protection against the use of a Reconfigure message by a malicious DHCP server to mount a denial of service 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 "in plain text" to the client.

Many of these rogue server attacks can be mitigated by making use of the mechanism described in [RFC7610].

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 threat common to both the client and the server is the resource "denial of service" (DoS) attack. These attacks typically involve the exhaustion of available assigned address or delegatable prefixes, or the exhaustion of CPU or network bandwidth, and are present anytime 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.

The messages exchanged between relay agents and servers may be used to mount a "man in the middle" or denial of service 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 [I-D.ietf-dhc-relay-server-security].

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 mis-configuration or exhaustion of other resources such as assignable address and delegatable prefixes.

Networks configured with delegated prefixes should be configured to preclude intentional or inadvertent inappropriate advertisement of these prefixes.

23. Privacy Considerations

This section focuses on the server considerations. For extended discussion about privacy considerations for the client, see [RFC7824]. In particular, Section 3 of that document discusses various identifiers that could be misused to track the client. Section 4 discusses existing mechanisms that may have an impact on client's privacy. Finally, Section 5 discusses potential attack vectors. For recommendations how to address or mitigate those issues, see [RFC7844].

This specification does not define any allocation strategies. 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 this list is not exhaustive and there are certainly other possible strategies. Readers are also encouraged to read [RFC7707], in particular Section 4.1.2 that 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 http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml

IANA is requested to update the http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xhtml page to add a reference to this document for definitions previously created by [RFC3315], [RFC3633], [RFC4242] and [RFC7083].

IANA is requested to add two columns to the DHCPv6 Option table at http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xhtml to indicate which options are allowed to appear in a client's ORO option (see Section 21.7) and which options are singleton options (only allowed to appear once as a top-level or encapsulated option - see Section 16 of [RFC7227]). Table 1 provides the data for the options assigned by IANA at the time of writing.

Option	Option Name (OPTION prefix removed)	Client ORO (1)	Singleton Option
1	CLIENTID	No	Yes
2	SERVERID	No	Yes
3	IA_NA	No	No
4	IA_TA	No	No
5	IAADDR	No	No
6	ORO	No	Yes
7	PREFERENCE	No	Yes
8	ELAPSED_TIME	No	Yes
9	RELAY_MSG	No	Yes
11	AUTH	No	Yes
12	UNICAST	Yes	Yes
13	STATUS_CODE	No	Yes
14	RAPID_COMMIT	No	Yes
15	USER_CLASS	No	Yes
16	VENDOR_CLASS	No	No (2)
17	VENDOR_OPTS	Optional	No (2)
18	INTERFACE_ID	No	Yes
19	RECONF_MSG	No	Yes
20	RECONF_ACCEPT	No	Yes
21	SIP_SERVER_D	Yes	Yes
22	SIP_SERVER_A	Yes	Yes
23	DNS_SERVERS	Yes	Yes
24	DOMAIN_LIST	Yes	Yes
25	IA_PD	No	No
26	IAPREFIX	No	No
27	NIS_SERVERS	Yes	Yes
28	NISP_SERVERS	Yes	Yes

29	NIS_DOMAIN_NAME	Yes	Yes
30	NISP_DOMAIN_NAME	Yes	Yes
31	SNTP_SERVERS	Yes	Yes
32	INFORMATION_REFRESH_TIME	Required for Information-request	Yes
33	BCMCS_SERVER_D	Yes	Yes
34	BCMCS_SERVER_A	Yes	Yes
36	GEOCONF_CIVIC	Yes	Yes
37	REMOTE_ID	No	Yes
38	SUBSCRIBER_ID	No	Yes
39	CLIENT_FQDN	Yes	Yes
40	PANA_AGENT	Yes	Yes
41	NEW_POSIX_TIMEZONE	Yes	Yes
42	NEW_TZDB_TIMEZONE	Yes	Yes
43	ERO	No	Yes
44	LQ_QUERY	No	Yes
45	CLIENT_DATA	No	Yes
46	CLT_TIME	No	Yes
47	LQ_RELAY_DATA	No	Yes
48	LQ_CLIENT_LINK	No	Yes
49	MIP6_HNIDF	Yes	Yes
50	MIP6_VDINF	Yes	Yes
51	V6_LOST	Yes	Yes
52	CAPWAP_AC_V6	Yes	Yes
53	RELAY_ID	No	Yes
54	OPTION-IPv6_Address-MoS	Yes	Yes
55	OPTION-IPv6_FQDN-MoS	Yes	Yes
56	NTP_SERVER	Yes	Yes
57	V6_ACCESS_DOMAIN	Yes	Yes
58	SIP_UA_CS_LIST	Yes	Yes
59	OPT_BOOTFILE_URL	Yes	Yes
60	OPT_BOOTFILE_PARAM	Yes	Yes
61	CLIENT_ARCH_TYPE	No	Yes
62	NII	Yes	Yes
63	GEOLOCATION	Yes	Yes
64	AFTR_NAME	Yes	Yes
65	ERP_LOCAL_DOMAIN_NAME	Yes	Yes
66	RSOO	No	Yes
67	PD_EXCLUDE	Yes	Yes
68	VSS	No	Yes
69	MIP6_IDINF	Yes	Yes
70	MIP6_UDINF	Yes	Yes
71	MIP6_HNP	Yes	Yes
72	MIP6_HAA	Yes	Yes
73	MIP6_HAF	Yes	Yes
74	RDNSS_SELECTION	Yes	No
75	KRB_PRINCIPAL_NAME	Yes	Yes
76	KRB_REALM_NAME	Yes	Yes

77	KRB_DEFAULT_REALM_NAME	Yes	Yes
78	KRB_KDC	Yes	Yes
79	CLIENT_LINKLAYER_ADDR	No	Yes
80	LINK_ADDRESS	No	Yes
81	RADIUS	No	Yes
82	SOL_MAX_RT	Required for Solicit	Yes
83	INF_MAX_RT	Required for Information-request	Yes
84	ADDRSEL	Yes	Yes
85	ADDRSEL_TABLE	Yes	Yes
86	V6_PCP_SERVER	Yes	No
87	DHCPV4_MSG	No	Yes
88	DHCP4_O_DHCP6_SERVER	Yes	Yes
89	S46_RULE	No	No (3)
90	S46_BR	No	No
91	S46_DMR	No	Yes
92	S46_V4V6BIND	No	Yes
93	S46_PORTPARAMS	No	Yes
94	S46_CONT_MAPE	Yes	No
95	S46_CONT_MAPT	Yes	Yes
96	S46_CONT_LW	Yes	Yes
97	4RD	Yes	Yes
98	4RD_MAP_RULE	Yes	Yes
99	4RD_NON_MAP_RULE	Yes	Yes
100	LQ_BASE_TIME	No	Yes
101	LQ_START_TIME	No	Yes
102	LQ_END_TIME	No	Yes
103	DHCP Captive-Portal	Yes	Yes
104	MPL_PARAMETERS	Yes	Yes
105	ANI_ATT	No	Yes
106	ANI_NETWORK_NAME	No	Yes
107	ANI_AP_NAME	No	Yes
108	ANI_AP_BSSID	No	Yes
109	ANI_OPERATOR_ID	No	Yes
110	ANI_OPERATOR_REALM	No	Yes
111	S46_PRIORITY	Yes	Yes
112	MUD_URL_V6 (TEMPORARY)	No	Yes
113	V6_PREFIX64	Yes	No
114	F_BINDING_STATUS	No	(4)
115	F_CONNECT_FLAGS	No	(4)
116	F_DNS_REMOVAL_INFO	No	(4)
117	F_DNS_HOST_NAME	No	(4)
118	F_DNS_ZONE_NAME	No	(4)
119	F_DNS_FLAGS	No	(4)
120	F_EXPIRATION_TIME	No	(4)
121	F_MAX_UNACKED_BNDUPD	No	(4)
122	F_MCLT	No	(4)

123	F_PARTNER_LIFETIME	No	(4)
124	F_PARTNER_LIFETIME_SENT	No	(4)
125	F_PARTNER_DOWN_TIME	No	(4)
126	F_PARTNER_RAW_CLT_TIME	No	(4)
127	F_PROTOCOL_VERSION	No	(4)
128	F_KEEPALIVE_TIME	No	(4)
129	F_RECONFIGURE_DATA	No	(4)
130	F_RELATIONSHIP_NAME	No	(4)
131	F_SERVER_FLAGS	No	(4)
132	F_SERVER_STATE	No	(4)
133	F_START_TIME_OF_STATE	No	(4)
134	F_STATE_EXPIRATION_TIME	No	(4)
143	IPv6_ADDRESS-ANDSF	Yes	Yes

Table 1: Updated Options Table

Notes for Table 1:

(1) For the "Client ORO" column: a "Yes" for an option means that the client includes this option code if it desires that configuration information; a "No" means that the option MUST NOT be included (and servers SHOULD silently ignore that option code if it appears in ORO).

(2) For each enterprise-number, there MUST only be a single instance.

(3) See [RFC7598] for details.

(4) See RFC8156 for details.

IANA is requested to update the All_DHCP_Relay_Agents_and_Servers (ff02::1:2) and All_DHCP_Servers (ff05::1:3) table entries in the IPv6 multicast address space registry at http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml to reference this document instead of [RFC3315].

IANA is requested to update the http://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml#authentication-protocol-id page to add an "Obsolete" annotation into the "DHCPv6 Delayed Authentication" entity in the "Authentication Suboption (value 8) - Protocol identifier values" registry, and https://www.iana.org/assignments/auth-namespaces/auth-namespaces.xhtml page to add an "Obsolete" annotation into the "Delayed Authentication" entity in the "Protocol Name Space Values"

registry. IANA is also requested to update these pages to reference this document instead of [RFC3315].

25. Obsoleted Mechanisms

This specification is mostly a corrected and cleaned up version of the original specification, [RFC3315], along with numerous additions from later RFCs. However, there are a small number of mechanisms that were not widely deployed, were underspecified or had other operational issues. Those mechanisms are now considered deprecated. Legacy implementations MAY support them, but implementations conformant to this document MUST NOT rely on them.

The following mechanisms are now obsolete:

Delayed Authentication. This mechanism was underspecified and had significant operational burden. As a result, after 10 years its adoption was extremely limited at best.

Lifetime hints sent by a client. Clients used to be allowed to send lifetime values as hints. This mechanism was not widely implemented and there were known misimplementations that sent the remaining lifetimes rather than total desired lifetimes. That in turn was sometimes misunderstood by servers as a request for ever decreasing lease lifetimes, which caused issues when values started approaching zero. Clients now SHOULD set lifetimes to 0 in IA Address and IA Prefix options, and servers MUST ignore any requested lifetime value.

T1/T2 hints sent by a client. These had similar issues to the lifetime hints. Clients now SHOULD set the T1/T2 values to 0 in IA_NA and IA_PD options, and servers MUST ignore any client supplied T1/T2 values.

26. Acknowledgments

This document is merely a refinement of earlier work by the authors of RFC3315 (Ralph Droms, Jim Bound, Bernie Volz, Ted Lemon, Charles Perkins, and Mike Carney), RFC3633 (Ole Troan and Ralph Droms), RFC3736 (Ralph Droms), RFC4242 (Stig Venaas, Tim Chown, and Bernie Volz), RFC7083 (Ralph Droms), and RFC7550 (Ole Troan, Bernie Volz, and Marcin Siodelski) and would not be possible without their original work.

A number of additional people have contributed to identifying issues with RFC3315 and RFC3633 and proposed resolutions to these issues as reflected in this document (in no particular order): Ole Troan, Robert Marks, Leaf Yeh, Michelle Cotton, Pablo Armando, John

Brzozowski, Suresh Krishnan, Hideshi Enokihara, Alexandru Petrescu, Yukiyo Akisada, Tatuya Jinmei, Fred Templin and Christian Huitema.

We also thank the following, not otherwise acknowledged and in no particular order, for their review comments: Jeremy Reed, Francis Dupont, Tatuya Jinmei, Lorenzo Colitti, Tianxiang Li, Ian Farrer, Yogendra Pal, Kim Kinnear, Shawn Routhier, Tim Chown, and Michayla Newcombe.

And, special thanks to Ralph Droms for answering many questions related to the original RFC3315 and RFC3633 work and for being the document shepherd.

27. References

27.1. Normative References

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

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <http://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, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <http://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, <http://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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc6355>.

[RFC7083] Droms, R., "Modification to Default Values of SOL_MAX_RT and INF_MAX_RT", RFC 7083, DOI 10.17487/RFC7083, November 2013, <http://www.rfc-editor.org/info/rfc7083>.

[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, <http://www.rfc-editor.org/info/rfc7227>.

[RFC7283] Cui, Y., Sun, Q., and T. Lemon, "Handling Unknown DHCPv6 Messages", RFC 7283, DOI 10.17487/RFC7283, July 2014, <http://www.rfc-editor.org/info/rfc7283>.

27.2. Informative References

[I-D.ietf-dhc-relay-server-security]
Volz, B. and Y. Pal, "Security of Messages Exchanged Between Servers and Relay Agents", draft-ietf-dhc-relay-server-security-05 (work in progress), April 2017.

[I-D.ietf-dhc-sedhcpv6]
Li, L., Jiang, S., Cui, Y., Jinmei, T., Lemon, T., and D. Zhang, "Secure DHCPv6", draft-ietf-dhc-sedhcpv6-21 (work in progress), February 2017.

[IANA-PEN]
IANA, "Private Enterprise Numbers registry http://www.iana.org/assignments/enterprise-numbers".

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

[RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <http://www.rfc-editor.org/info/rfc826>.

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

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <http://www.rfc-editor.org/info/rfc2132>.

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

[RFC3041] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, DOI 10.17487/RFC3041, January 2001, <http://www.rfc-editor.org/info/rfc3041>.

[RFC3162] Aboba, B., Zorn, G., and D. Mitton, "RADIUS and IPv6", RFC 3162, DOI 10.17487/RFC3162, August 2001, <http://www.rfc-editor.org/info/rfc3162>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <http://www.rfc-editor.org/info/rfc3315>.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <http://www.rfc-editor.org/info/rfc3633>.

[RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, DOI 10.17487/RFC3646, December 2003, <http://www.rfc-editor.org/info/rfc3646>.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, DOI 10.17487/RFC3736, April 2004, <http://www.rfc-editor.org/info/rfc3736>.

[RFC3769] Miyakawa, S. and R. Droms, "Requirements for IPv6 Prefix Delegation", RFC 3769, DOI 10.17487/RFC3769, June 2004, <http://www.rfc-editor.org/info/rfc3769>.

[RFC4075] Kalusivalingam, V., "Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6", RFC 4075, DOI 10.17487/RFC4075, May 2005, <http://www.rfc-editor.org/info/rfc4075>.

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

[RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 4242, DOI 10.17487/RFC4242, November 2005, <http://www.rfc-editor.org/info/rfc4242>.

[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, <http://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, <http://www.rfc-editor.org/info/rfc4704>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <http://www.rfc-editor.org/info/rfc4941>.

[RFC4943] Roy, S., Durand, A., and J. Paugh, "IPv6 Neighbor Discovery On-Link Assumption Considered Harmful", RFC 4943, DOI 10.17487/RFC4943, September 2007, <http://www.rfc-editor.org/info/rfc4943>.

[RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", RFC 5007, DOI 10.17487/RFC5007, September 2007, <http://www.rfc-editor.org/info/rfc5007>.

[RFC5453] Krishnan, S., "Reserved IPv6 Interface Identifiers", RFC 5453, DOI 10.17487/RFC5453, February 2009, <http://www.rfc-editor.org/info/rfc5453>.

[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, <http://www.rfc-editor.org/info/rfc5905>.

[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, <http://www.rfc-editor.org/info/rfc6603>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <http://www.rfc-editor.org/info/rfc6724>.

[RFC6879] Jiang, S., Liu, B., and B. Carpenter, "IPv6 Enterprise Network Renumbering Scenarios, Considerations, and Methods", RFC 6879, DOI 10.17487/RFC6879, February 2013, <http://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, <http://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, <http://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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc7368>.

[RFC7421] Carpenter, B., Ed., Chown, T., Gont, F., Jiang, S., Petrescu, A., and A. Yourtchenko, "Analysis of the 64-bit Boundary in IPv6 Addressing", RFC 7421, DOI 10.17487/RFC7421, January 2015, <http://www.rfc-editor.org/info/rfc7421>.

[RFC7550] Troan, O., Volz, B., and M. Siodelski, "Issues and Recommendations with Multiple Stateful DHCPv6 Options", RFC 7550, DOI 10.17487/RFC7550, May 2015, <http://www.rfc-editor.org/info/rfc7550>.

[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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc7844>.

[RFC7969] Lemon, T. and T. Mrugalski, "Customizing DHCP Configuration on the Basis of Network Topology", RFC 7969, DOI 10.17487/RFC7969, October 2016, <http://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, <http://www.rfc-editor.org/info/rfc8168>.

[TR-187] Broadband Forum, "TR-187 - IPv6 for PPP Broadband Access", February 2013, <https://www.broadband-forum.org/technical/download/TR-187_Issue-2.pdf>.

Appendix A. Summary of Changes

This appendix provides a summary of the significant changes made to this updated DHCPv6 specification.

1. The Introduction Section 1 was reorganized and updated. In particular, the client/server message exchanges were moved into a new (and expanded) section on their own (see Section 5). And, new sections were added to discuss the relation to previous DHCPv6 documents and also to DHCPv4.

2. The Requirements Section 2 and Background Section 3 had very minor edits.

3. The Terminology Section 4 had minor edits.

4. The DHCP Terminology Section 4.2 was expanded to incorporate definitions from RFC3633, add T1/T2 definitions, add a few new definitions useful in a document that combined address and prefix delegation assignments, and improve some existing definitions.

5. The Client-Server Exchanges Section 5 was added from material previously in the Introduction Section 1 of RFC3315 and was expanded.

6. The Operational Models Section 6 is new and provides information on the kinds of DHCP clients and how they operate.

7. The DHCP Constants Section 7 was primarily updated to add constants from RFC4242 and RFC7083.

8. The Client/Server Message Formats Section 8, Relay Agent/Server Message Formats Section 9, and Representation and Use of Domain Names Section 10 had only very minor changes.

9. The DHCP Unique Identifier (DUID) Section 11 now discourages, rather than disallows, a server to parse the DUID, now includes some information on the DUID-UUID (RFC6355), and has other minor edits.

10. The Identity Association Section 12 was expanded to better explain the concept and also included prefix delegation.

11. The Assignment to an IA Section 13 incorporates material from two sections (11 and 12) of RFC3315 and also includes a section on prefix delegation.

12. The Transmission of Messages by a Client Section 14 was expanded to include rate limiting by clients and how clients should handle T1 or T2 values of 0.

13. The Reliability of Client Initiated Message Exchanges Section 15 was expanded to clarify that the Elapsed Time option must be updated in retransmitted messages and that a client is not required to listen for DHCP traffic for the entire retransmission period.

14. The Message Validation Section 16 had minor edits.

15. The Client Source Address and Interface Selection Section 17 was expanded to include prefix delegation.

16. The DHCP Configuration Exchanges Section 18 consolidates what used to be in the RFC3315 DHCP Server Solicitation Section 17, DHCP Client-Initiated Configuration Exchange Section 18, and DHCP Server-Initiated Configuration Exchange Section 19. This material was reorganized and enhanced, and incorporates prefix delegation from RFC3633 and other changes from RFC4242, RFC7083, and RFC7550. A few changes of note:

    1. The Option Request option is no longer optional for some messages (Solicit and Information-request) as RFC7083 requires clients to request SOL_MAX_RT or INF_MAX_RT options.

    2. The Reconfigure message should no longer contain IA_NA/ IA_PD, ORO, or other options to indicate to the client what was reconfigured. The client should request everything it needs in the response to the Reconfigure.

    3. Lifetime and T1/T2 hints should not be sent by a client (it should send 0 values in these fields) and any non-zero values should be ignored by the server.

    4. Clarified that a server may return different addresses in the Reply than requested by a client in the Request message. Also clarified that a server must not include addresses that it will not assign.

    Also, a Refreshing Configuration Information Section 18.2.12 was added indicating use cases for when a client should try to refresh network information.

17. The Relay Agent Behavior Section 19 had minor edits.

18. The Authentication of DHCP Messages Section 20 had significant changes: IPsec materials were mostly removed and replaced with a reference to [I-D.ietf-dhc-relay-server-security], and the Delay Authentication Protocol was removed (see Section 25). Note that the Reconfigure Key Authentication Protocol is retained.

19. The DHCP Options Section 21 was expanded to incorporate the prefix delegation options from RFC3633, the Information Refresh Time option from RFC4242, and the SOL_MAX_RT and INF_MAX_RT options from RFC7083. In addition, some additional edits were made to clarify option handling, such as which options should not be in an Option Request option.

20. The Security Considerations Section 22 were updated to expand the discussion of security threats and incorporate material from the incorporated documents, primarily RFC3633.

21. The new Privacy Considerations Section 23 was added to consider privacy issues.

22. The IANA Considerations Section 24 was rewritten to reflect the changes requested for this document as other documents have already made the message, option, DUID, and status code assignments and this document does not add any new assignments.

23. The new Obsoleted Mechanisms Section 25 documents what this specification obsoletes.

24. The Appearance of Options in Message Types Appendix B and Appearance of Options in the Options Field of DHCP Appendix C were updated to reflect the incorporated options from RFC3633, RFC4242, and RFC7083.

25. Where appropriate, informational references have been added to provide further background and guidance throughout the document (as can be noted by the vast increase in references).

26. General changes to other IPv6 specifications, such as removing the use of site-local unicast addresses and adding unique local addresses, were made to the document. Note that in a few places, older obsoleted RFCs (such as RFC2462 related to M and O bit handling) are still referenced as the material cited was not added in the replacement RFC.

27. It should be noted that this document does not refer to all DHCPv6 functionality and specifications. Readers of this specification should visit http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml and

https://datatracker.ietf.org/wg/dhc/ to learn of the RFCs that define DHCPv6 messages, options, status-codes, and more.

Appendix B. Appearance of Options in Message Types

The following tables indicates with a "*" the options are allowed in each DHCP message type.

These tables are informational and should they conflict with text earlier in this document, that text should be considered authoritative.

	Client ID	Server ID	IA_NA/ IA_TA	IA_PD	ORO	Pref	Elap. Time	Relay Msg.	Auth.	Server Unicast
Solicit	*		*	*	*		*			
Advert.	*	*	*	*		*				
Request	*	*	*	*	*		*			
Confirm	*		*				*			
Renew	*	*	*	*	*		*			
Rebind	*		*	*	*		*			
Decline	*	*	*	*			*			
Release	*	*	*	*			*			
Reply	*	*	*	*					*	*
Reconf.	*	*							*	
Inform.	*	(see note)			*		*			
R-forw.								*		
R-repl.								*		

NOTE: Server ID option 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 Refresh Time
Solicit		*	*	*	*			*	
Advert.	*		*	*	*			*	
Request			*	*	*			*	
Confirm			*	*	*				
Renew			*	*	*			*	
Rebind			*	*	*			*	
Decline			*	*	*				
Release			*	*	*				
Reply	*	*	*	*	*			*	*
Reconf.							*		
Inform.			*	*	*			*	
R-forw.					*	*			
R-repl.					*	*			

	SOL_MAX_RT	INF_MAX_RT
Solicit		
Advert.	*	
Request		
Confirm		
Renew		
Rebind		
Decline		
Release		
Reply	*	*
Reconf.		
Inform.		
R-forw.		
R-repl.		

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 encapsulated in other options defined in this document. Other RFC's may define additional situations where options defined in this document are encapsulated in other options.

This table is informational and should it conflict with text earlier in this document, that text should be considered authoritative.

	Top-Level	IA_NA/IA_TA	IAADDR	IA_PD	IAPREFIX	Relay-Forw	Relay-Reply
Client ID	*						
Server ID	*						
IA_NA/IA_TA	*						
IAADDR		*					
IA_PD	*						
IAPREFIX				*			
ORO	*						
Preference	*						
Elapsed Time	*						
Relay Message						*	*
Authentic.	*						
Server Uni.	*						
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	*						

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" / "Relay-Reply" column appear in the options field of the Relay-forward and Relay-reply messages (see Section 9).

Authors' Addresses

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

Email: tomasz.mrugalski@gmail.com

Marcin Siodelski
Internet Systems Consortium, Inc.
950 Charter Street
Redwood City, CA 94063
USA

Email: msiodelski@gmail.com

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

Email: volz@cisco.com

Andrew Yourtchenko
Cisco Systems, Inc.
De Kleetlaan, 7
Diegem B-1831
Belgium

Email: ayourtch@cisco.com

Michael C. Richardson
Sandelman Software Works
470 Dawson Avenue
Ottawa, ON K1Z 5V7
CA

Email: mcr+ietf@sandelman.ca
URI: http://www.sandelman.ca/

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: jiangsheng@huawei.com

Ted Lemon
Nominum, Inc.
800 Bridge St.
Redwood City, CA 94043
USA

Email: Ted.Lemon@nominum.com

Timothy Winters
University of New Hampshire, Interoperability Lab (UNH-IOL)
Durham, NH
USA

Email: twinters@iol.unh.edu